

Avances en Informática y Automática

Séptimo Workshop



Avances en Informática y Automática

Séptimo Workshop

Avances en Informática y Automática

Séptimo Workshop

Editores:

Juan Cruz Benito

Alicia García Holgado

Sergio García Sánchez

Daniel Hernández Alfageme

María Navarro Cáceres

Roberto Vega Ruiz

Publicado en España por:

Departamento de Informática y Automática
Facultad de Ciencias
Universidad de Salamanca
Plaza de los Caídos s/n
37008, Salamanca, España
Tel.: + 34 923294653
Fax: + 34 923294514
Web: <http://mastersi.usal.es>
Email: mastersi@usal.es

ISBN - 10 84-695-8670-X

ISBN - 13 978-84-695-8670-9

Editores:

Juan Cruz Benito
Alicia García Holgado
Sergio García Sánchez
Daniel Hernández Alfageme
María Navarro Cáceres
Roberto Vega Ruiz

Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 España.



Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. Como colofón a las diferentes tareas de investigación que se llevan a cabo durante el Máster, el Departamento de Informática y Automática organiza un *workshop* en el que se da la oportunidad a los estudiantes de presentar los principales resultados de sus Trabajos de Fin de Máster y obtener una realimentación del interés de los mismos.

La séptima edición del *workshop* “Avances en Informática y Automática”, correspondiente al curso 2012 - 2013, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación, desde las redes neuronales, los sistemas inteligentes o la minería de datos hasta la informática educativa, la visualización de información o la recuperación de información. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral.

Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* “Avances en Infomática y Automática” está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Program Committee

Belén Curto Diego
José Rafael García-Bermejo Giner
Vivian F. López Batista
Luis Antonio Miguel Quintales
Belén Pérez Lancho
Ángel Luis Sánchez Lázaro
Pastora Isabel Vega Cruz

Índice general

| | |
|--|-----|
| Desarrollo de una Calculadora Vocal para Android | 1 |
| <i>Juan José Álvarez Navarrete y Ángel Luis Sánchez Lázaro</i> | |
| Análisis Léxico sobre los <i>Tweets</i> de <i>Twitter</i> | 21 |
| <i>Astrid Paola Bográn, José Luis Alonso Berrocal y Luis Carlos García de Figuerola Paniagua</i> | |
| Soluciones Visuales Interactivas aplicadas a Grandes Volúmenes de Datos de Entornos 3D de Aprendizaje y Prácticas | 37 |
| <i>Juan Cruz Benito, Roberto Therón Sánchez y Emiliana Pizarro Lucas</i> | |
| Análisis de integración de soluciones basadas en <i>software</i> como servicio para la implantación de ecosistemas tecnológicos corporativos | 55 |
| <i>Alicia García Holgado y Francisco José García Peñalvo</i> | |
| Modelado Matemático para la Generación de un Mallado Espacial con Aplicación a Nano-Estructuras Semiconductoras | 73 |
| <i>Sergio García, José Rafael García-Bermejo e Ignacio Íñiguez-de-la-Torre</i> | |
| Sistema Multiagente desplegado en un entorno Cloud Computing para gestión de zonas de aparcamiento | 93 |
| <i>Daniel Hernández Alfageme y Juan Manuel Corchado</i> | |
| Aplicación Multiagente en Entornos Móviles para la Toma de Datos en Zonas de Aparcamiento | 107 |
| <i>María Navarro y Juan Manuel Corchado</i> | |
| Estudio de Técnicas de Análisis Visual de Guiones Cinematográficos | 119 |
| <i>Daniel Toimil Martín y Roberto Therón Sánchez</i> | |
| Mobile Image Recognition with Convolutional Neural Networks | 133 |
| <i>William Raveane and María Angélica González Arrieta</i> | |
| Sistema Multiagente Basado en Organizaciones para el Control de Vídeo Multidispositivo | 149 |
| <i>Juan Agustín Rodríguez Valle y Juan Manuel Corchado</i> | |
| Intelligent Models and Simulation to Optimise Industrial Processes | 163 |
| <i>Roberto Vega Ruiz, Héctor Quintián Pardo and Emilio Corchado Rodríguez</i> | |

| | |
|--|-----|
| Análisis Cibernético y Visual de Twitter | 177 |
| <i>Zhangxian Zheng, José Luis Alonso Berrocal y Luis Carlos García de Figuerola Paniagua</i> | |
| Autores | 189 |

Desarrollo de una Calculadora Vocal para Android

Juan José Álvarez Navarrete¹ y Ángel Luis Sánchez Lázaro¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{juanjo7an,alsl}@usal.es

Resumen Es por todos conocida la rápida evolución de las tecnologías móviles en estos últimos años. Hace tan solo una década nadie podría augurar la desorbitada cantidad de dispositivos (smartphone) que existen en la actualidad. Su rápida expansión y uso ha provocado un elevado aumento en la demanda de aplicaciones orientadas a estos dispositivos, ya sean videojuegos, aplicaciones básicas y comunes (como podría serlo una calculadora o un cronómetro) o algunas más complejas, pero no por ello menos útiles, como es el caso de una calculadora vocal.

Siguiendo esta corriente de expansión y visto que el empleo de estos dispositivos móviles es tremendamente dificultoso o imposible para personas cuyo sentido de la visión es reducido o nulo, se ha decidido desarrollar una aplicación que nos permita realizar operaciones básicas, sirviéndonos para ello únicamente de la voz.

Keywords: Reconocimiento de Voz, RAH, Android, Dispositivos Móviles, Smartphone

1. Introducción

El proceso de reconocimiento automático del habla (RAH) dota a las máquinas de la capacidad de recibir mensajes orales. Tomando como entrada la señal acústica recogida por un micrófono, el proceso de reconocimiento automático del habla tiene como objetivo final descodificar el mensaje contenido en la onda acústica para realizar las acciones pertinentes. Para lograr este fin, un sistema de RAH necesitaría conjugar una gran cantidad de conocimientos acerca del sistema auditivo humano, sobre la estructura del lenguaje, la representación del significado de los mensajes y sobre todo el autoaprendizaje de la experiencia diaria. Actualmente estamos lejos de lograr un sistema completo que pueda comprender cualquier mensaje oral en cualquier contexto tal y como lo podría hacer un ser humano. Sin embargo, la tecnología actual sí que permite realizar sistemas de RAH que pueden trabajar, con un error aceptable, en entornos semánticos restringidos. Básicamente, el reconocimiento del habla es un proceso de clasificación de patrones, cuyo objetivo es clasificar la señal de entrada (onda acústica) en una secuencia de patrones previamente aprendidos y almacenados en unos diccionarios de modelos acústicos y de lenguaje. Este proceso de clasificación

supone, en primer lugar que la señal de voz puede ser analizada en segmentos de corta duración y representar cada uno de los segmentos mediante su contenido frecuencial, de forma análoga al funcionamiento del oído, en segundo lugar que mediante un proceso de clasificación podemos asignar a cada segmento o conjuntos consecutivos de segmentos una unidad con significado lingüístico y finalmente, en tercer lugar, que mediante un procesador lingüístico podemos dar significado a las secuencias de unidades. Este último paso del sistema supone incorporar al sistema de RAH conocimiento acerca de la estructura sintáctica, semántica y pragmática del lenguaje. Sin embargo, los sistemas actuales de RAH solo incorporan estas fuentes de conocimiento sobre tareas muy restringidas y controladas, estando la mayoría de ellos en experimentación en condiciones de laboratorio.



Figura 1. Reconocimiento Automático del Habla

De forma progresiva, la complejidad de los sistemas de reconocimiento de voz se ha ido incrementando junto con su precisión, por desgracia a expensas de la velocidad de éste. La razón de este desarrollo está asociada a un criterio único en la evaluación para sistemas de reconocimiento de voz: la exactitud de las palabras en los bancos de pruebas. Con este paradigma, la velocidad de reconocimiento sólo es relevante cuando limita el número de experimentos que pueden llevarse a cabo para elevar la precisión de las palabras. Para conseguir una punta de rendimiento, los reconocedores normales utilizados en tales evaluaciones tardan varios cientos de segundos en procesar un simple segundo de voz introducida en los ordenadores actuales. Sin embargo, para la mayoría de las aplicaciones del mundo real, la velocidad de reconocimiento es tan importante como la precisión en el reconocimiento. Un sistema de dictado que invierta cuatro días en procesar

media hora de habla no será un buen producto. Un usuario no aceptará que una interfaz de base de datos tarde quince minutos en responder a una consulta de cuatro segundos. Decir que la velocidad carece de importancia porque cada nueva generación de programas es más veloz es válido sólo en parte. Por un lado, los sistemas tienden a hacerse más complejos en una progresión superior a la del aumento de velocidad de los ordenadores. Por otro, se supone que un sistema de reconocimiento de voz aplicado como interfaz para a introducción de datos no debería devorar el 100 %.

2. Revisión del estado del arte

En 1952 Davis, Bidulph y Balashek, de los laboratorios Bell fabricaron el primer reconocedor capaz de discriminar con cierta precisión los diez dígitos ingleses pronunciados de forma aislada por un único lector. El dispositivo era totalmente electrónico. Los primeros trabajos que hacen uso de tecnología informática, comienzan a aparecer en 1959/1960; Deves y Mathews introducen el concepto de normalización temporal no lineal, que permite la comparación de parámetros de palabras guals pronunciadas a distinta velocidad.

A partir de estas fechas comienza la explosión de trabajos, principalmente de reconocimiento de palabras aisladas, con la extrapolación optimista, por parte de investigadores y organismos financieros, de llegar, en poco tiempo, a sistemas capaces de reconocer de forma precisa frases cualesquiera, pronunciadas por un lector cualquiera, de forma continua [1].

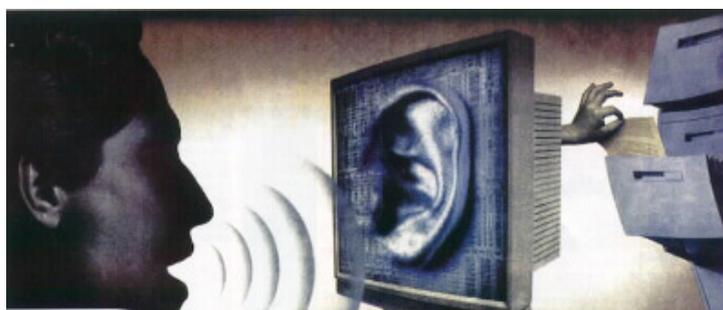


Figura 2. Reconocimiento de la voz

Con este objetivo mas o menos en mente, se lanzan grandes proyectos de investigación en los que se pretende llegar a las menores restricciones gramaticales posibles de las frases a reconocer, así como del léxico utilizado. Son varios los países en los que se comienza a trabajar en proyectos de ésta índoles (Japón, Francia, etc.), pero en EE.UU. donde se lanza, en 1971, el mayor proyecto conocido en la historia del reconocimiento del habla. Se trata del «ARPA-SUR»

(Advanced Research Projects Agency - Speech Understanding Research), con un presupuesto de quince millones de dólares y una duración de cinco años.

Aunque los ambiciosos objetivos pretendidos en éste y otros proyectos no llegaron realmente a alcanzarse. Las aportaciones derivadas de ellos, contribuyeron de forma notable a un mejor conocimiento de los mecanismos del habla y de las limitaciones de los sistemas automáticos de reconocimiento.

2.1. Problemas de Reconocimiento

1. Multinteractividad: Cada uno de estos niveles utiliza los conocimientos sobre el lenguaje, para extraer su parte correspondiente de la Información total necesaria para la comprensión del lenguaje.
 - 1) Nivel acústico: Las características físicas de la señal de voz.
 - 2) Nivel fonético: Los objetos sonoros elementales.
 - 3) Nivel léxico: Donde empieza la abstracción y se determinan las estructuras simbólicas primarias (palabras o morfemas).
 - 4) Nivel sintáctico: En el que se aplican reglas para la sucesión de palabras y comprobar su adecuación a la gramática del lenguaje.
 - 5) Nivel semántico-pragmático: Donde se llega a la comprensión del significado del lenguaje, comprobando la coherencia del mensaje con el de la realidad, así como el del contexto.
2. Continuidad: A pesar de que se tenga la impresión contraria, en el habla natural ni los fonemas, ni las sílabas, ni siquiera las palabras, constituyen elementos discretos que se puedan separar fácilmente de forma automática. La separación de estos elementos la realiza el ser humano gracias a sus conocimientos previos de la lengua.
3. Variabilidad: El habla presenta una gran variabilidad: es imposible que un locutor pronuncie dos veces exactamente igual una misma sílaba, palabra o frase. En cuanto a la variabilidad del interlocutor, ésta es debida principalmente a las diferencias físicas de los aparatos fonadores de los locutores. También son importantes los hábitos de habla diferentes, según la procedencia geográfica, etc. Una importante variabilidad es el entorno y el canal de transmisión: ruidos, interferencias...
4. Redundancia: La mayor parte de la información contenida en la voz es redundante. Se puede demostrar que unos 50 bits/segundo son suficientes para transmitir el mensaje lingüístico contenido en ella, mientras que para una transmisión sonora completa se requieren del orden de 100.000 bits/segundo (8 KHz de frecuencia de muestreo y 12 bits por muestra, por ejemplo). Este suplemento de información contiene los datos que identifican al locutor, su estado de ánimo, su entorno,... y los que hacen posible la comprensión a pesar de la variabilidad y en entornos plagados de ruido.

2.2. Técnicas usadas

Por medio de las técnicas actuales de reconocimiento de formas de señales acústicas como la FFT ("Fast Fourier Transform") y a través de los métodos de

I.A. (Inteligencia Artificial) hasta la fecha conocidas, podemos resolver en parte la mayoría de las variables del problema que en el Reconocimiento del habla existen. En el habla, el universo físico de los objetos a reconocer está constituido

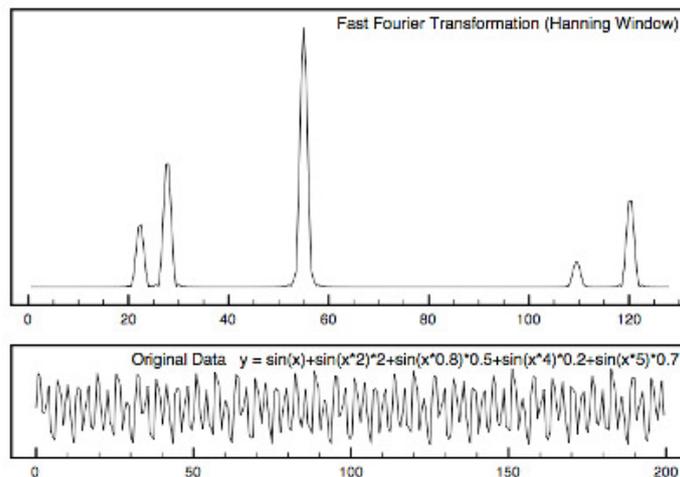


Figura 3. Fast Fourier Transform

por las ondas de presión producidas por el aparato fonador humano. Los objetos externos de este universo son los constituyen las diferentes formas acústicas del habla. La parte inicial de todo subsistema de preproceso de la señal vocal estará siempre construida por:

- Un micrófono, que convertirá la onda sonora de presión en una señal eléctrica.
- Un amplificador, que extenderá hasta nivel manejable la débil señal que proporciona el micrófono.
- Un filtro activo para bajas, que eliminará la altas frecuencias indispensables según el teorema de muestreo de Nyquist.

Fast Fourier Transform A partir de la señal eléctrica que produce el amplificador sería teóricamente posible construir un sistema de reconocimiento por medios totalmente analógicos. Sin embargo, en el estado actual de la tecnología, resulta mas conveniente utilizar técnicas digitales: sobre todo para las partes del sistema involucradas en la decisión [2]. Básicamente un convertidor A/D debe realizar dos tareas:

- Muestrear la señal analógica: Es decir, medir la amplitud de dicha señal, cierto intervalo de tiempo.
- Cuantificar la señal muestreada; es decir, codificar numéricamente el resultado de cada una de las medidas.

De esta manera, una función continua en el tiempo quedará representada por una serie discreta de valores numéricos. Al proceso de combinado de transducción, muestreo y cuantificación se le llama adquisición.

1. Muestreo: Una señal muestre a intervalos de tiempo T , $S_M(t)$, puede definirse como el producto de la señal continua $s(t)$ y una función “peine” de Δ Direc (función impulso):

$$S_M(t) = S(t) \sum_{k=-\infty}^i nfty \Delta(t - kT) \quad (1)$$

$$S_M(t) = S(t) \sum_{k=-\infty} S(kT) \Delta(t - kT) \quad (2)$$

De donde es inmediato demostrar que la expresión del espectro $S_M(j\omega)$ de la señal muestreada en función de la señal sin muestrear $S(j\omega)$, adopta la forma:

$$S_M(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^i nfty S[j(\Omega + k\Omega_0)] \quad (3)$$

$$\Omega_0 = \frac{2}{T} \quad (4)$$

Expresión que representa la superposición del espectro de $s(t)$ con las sucesivas versiones del mismo desplazadas en el eje de las frecuencias con periodicidad $1/T$. Resulta evidente que si el ancho de banda de la señal a muestrear es excesivo con relación a la frecuencia de muestreo, se producirá un solapamiento irreversible de los espectros sucesivos, haciendo imposible la reconstrucción de la señal original. Este solapamiento (aliasing) ocurre siempre que la máxima frecuencia (F_b) del espectro no nulo de la señal a muestrear sea superior a la mitad de la frecuencia de muestreo (F_m) (frecuencia Nyquist).

$$F_M > 2F_b \quad (5)$$

Antes de muestrear una señal será pues necesario limitar la frecuencia máxima de ésta a la mitad de la de muestreo, lo que se puede conseguir mediante un filtro analógico de paso bajo previo al convertidos A/D, cuya frecuencia de corte sea la de Nyquist como máximo. La anchura de banda de la señal resultante deberá preservar la información relevante necesaria para una adecuada descripción de los objetivos acústicos a tratar.

2. Cuantificación: En cada impulso de muestreo, el convertidor A/D compara la señal muestreando con cosa dado un conjunto de entradas, podemos usar unidades sumadoras con nivel de disparo (Threshold) como simples compuertas AND, OR, y NOT poniendo apropiadamente el nivel de cuantificación predefinidos. El número de niveles (N) determina la precisión del análisis y, por tanto, el número de bits (b) necesarios para la presentación digital de cada muestra:

$$b = \log_2 N \quad (6)$$

Teniendo en cuenta la relación de Nyquist se puede determinar el flujo de información, en bits por segundo, resultante del proceso combinado de muestreo y cuantificación:

$$\Theta > 2Fb \log_2 N \quad (7)$$

Para señales vocales adquiridas directamente en el dominio del tiempo dicho flujo suele oscilar entre 50 Kbits/s y 300Kbits/s.

Densidad de cruce por cero La densidad de cruces por cero ha sido objeto de numerosos estudios teóricos y práctico. Su utilidad en reconocimiento del habla radica en que proporciona una estimación aproximada del contenido frecuencial de una señal, basada en la idea de que una sinusoide pura cruza el eje de abscisas 2 veces por periodo. Es un parámetro de muy baja complejidad de cálculo, y se le ha utilizado para detectar segmentos fricativos (señal de pequeña energía y elevada densidad de cruces por cero). Después de obtenidos los parámetros, estos se alimentan a la red neuronal de aprendizaje (Técnica de Inteligencia Artificial) que distinguirá entre uno y otro fonema. Redes Neuronales (Back-propagation). ¿Qué es lo que una red multicapas puede computar? La respuesta es: cualquiera. El mayor problema es el aprendizaje. La representación del conocimiento en las redes Neuronales es un poquito opaco: Las redes deben aprender su propia representación debido que programarlas a mano es imposible. Las redes Neuronales [3] pueden aprender cualquier cosa que ellas puedan computar. En la figura, Xi, Hi y Oi representan las unidades de niveles de activación de las unidades de entrada ocultas y de salida. Los pesos en conexión entre las unidades de entrada y las ocultas están denotadas aquí por las relaciones que existen entre los niveles de las O's y los niveles de las H's, al mismo tiempo los pesos entre las unidades ocultas con las capas de salida están denotadas por las relaciones que existen entre los niveles de las H's y los niveles de las X's.

Estas redes tienen tres capas, aunque esto es posible y algunas veces es útil tener más. Cada unidad en una capa está conectada en la dirección hacia delante con cada unidad en la capa próxima. La activación fluye desde la unidad de entrada hacia la capa oculta, entonces pasa a la capa de salida. Como es usual, el conocimiento de la red se codifica en los pesos de conexión entre las unidades. En contraste con el método paralelo de relajación usado por las redes de Hopfield. Las redes de propagación hacia atrás lleva a cabo una simple serie de cálculos. Debido a que la activación fluye solo en una dirección, no hay necesidad de un proceso iterativo de relajación. Los niveles de las unidades de la capa de salida determinan la salida de la red. La esperanza al atacar problemas como el reconocimiento de la escritura a mano es que las redes Neuronales no solamente aprenderán a clasificar las entradas con que fueron entrenadas sino que generalizarán y serán capaces de clasificar entradas que aún no han sido vistas.

2.3. Estándar ETSI 201 108

El estándar ETSI 201 108 define las características de un sistema de reconocimiento del habla distribuido completo, como son: el procesado de la señal,

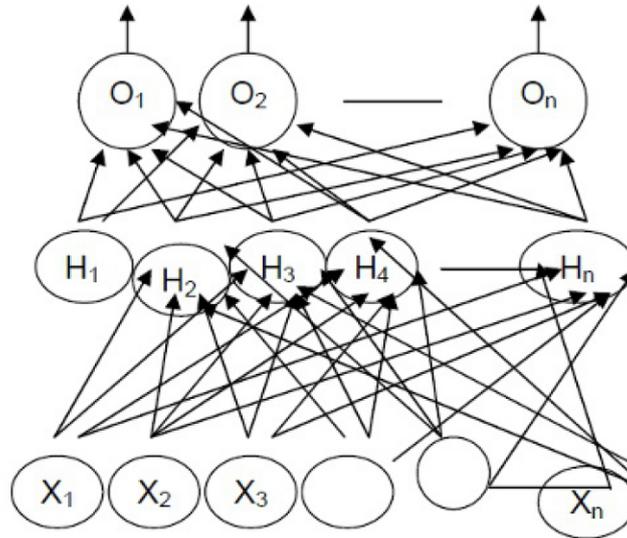


Figura 4. Ejemplo Red Neuronal en Capas

el algoritmo de extracción de características, los algoritmos de compresión, la transmisión y la calidad [4].

2.4. Comunicación en sistemas distribuidos

Un sistema distribuido se define como una colección de computadoras autónomas separadas físicamente y conectadas entre sí por una red de comunicación. En ella cada máquina posee sus propios componentes de hardware y el software adecuado para que el sistema sea visto por los usuarios como un único sistema de computación.

Arquitectura cliente-servidor La arquitectura cliente-servidor es un modelo de sistema distribuido en el que el trabajo se reparte entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta, como podemos ver en la representación de la Figura. La mayoría de las aplicaciones en Internet utilizan la arquitectura cliente-servidor, como por ejemplo los servidores web, los servidores de archivos, los servidores del correo, etc. [5].

Cliente: Es quien inicia las peticiones y espera las respuestas del servidor. Por lo general, puede conectarse a varios servidores a la vez y normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

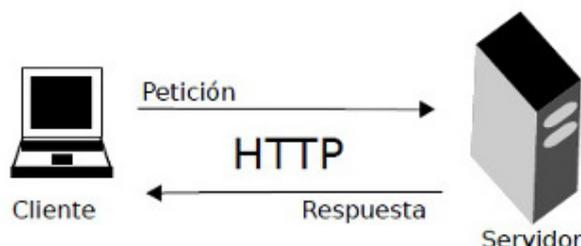


Figura 5. Sistema cliente-servidor

Servidor: Al iniciarse espera a que le lleguen las peticiones de los clientes. Tras la recepción de una solicitud, la procesa y luego envía la respuesta al cliente. Por lo general, acepta conexiones desde un gran número de clientes y no es frecuente que interactúa directamente con los usuarios finales.

A los puntos finales de una comunicación entre dos sistemas que intercambian información a través de una red de comunicaciones se les llama sockets. Un socket queda definido por la dirección IP del dispositivo en el que se encuentra y un número de puerto de 16 bits. Una conexión está determinada por un par de sockets, el del cliente y el del servidor. En la Figura se muestra un modelo típico de comunicación por sockets en una arquitectura cliente-servidor. Hay dos tipos de sockets: socket stream y socket datagram.

Socket stream: Utilizan el protocolo TCP (Transmission Control Protocol), por lo que también son conocidos como sockets orientados a conexión. El utilizar el protocolo TCP implica que antes de enviar la información se debe establecer la conexión entre los dos sockets, y ofrece la ventaja de que incorpora la corrección de errores de forma transparente al programador [6].

Socket datagram: Utilizan el protocolo UDP (User Datagram Protocol), por lo que también son conocidos como sockets sin conexión. Al utilizar dicho protocolo se puede enviar la información sin necesidad de haber establecido la comunicación entre los sockets. La ventaja de este tipo de sockets es que produce una menor sobrecarga sobre la información transmitida, a cambio de no tener incorporada la corrección de errores. Esto lo hace más rápido y por tanto más interesante en una red en la que se pierdan pocos paquetes y para aplicaciones de tiempo real.

2.5. Análisis

En un futuro próximo se espera poder tener reconocedores de lenguaje humano que será sólo para un locutor definido. Gracias a las técnicas de reconocimiento de patrones de voz como FFT usadas junto con las redes Neuronales lograran buenos resultados.

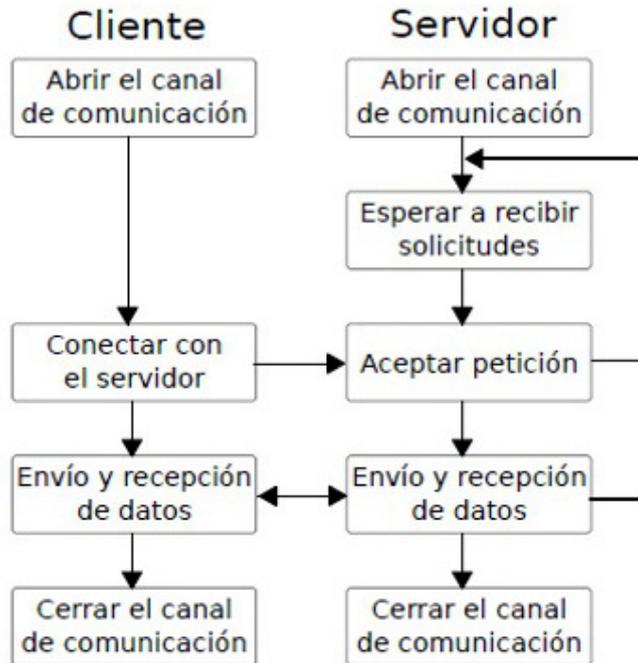


Figura 6. Esquema funcionamiento cliente servidor

3. Dispositivos Móviles

Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada y diseñados específicamente para una función, aunque pueden llevar a cabo otras más generales [7].

Definición:

Dispositivo móvil de datos mejorado: Dispositivos que tienen pantallas de medianas a grandes (por encima de los 240 x 120 pixels), avegación de tipo stylus, y que ofrecen las mismas características que el “dispositivo móvil de datos básico”, más aplicaciones nativas y corporativas usuales, en versión móvil. Este tipo de dispositivos utilizan un sistema operativo como iOS, Windows Phone, Symbian, Android, etc.

3.1. Sistemas operativos móviles

Un sistema operativo móvil es un programa que gestiona los procesos básicos de un dispositivo móvil al igual que las computadoras utilizan Windows

o Linux entre otros. Sin embargo, los sistemas operativos móviles son bastante más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

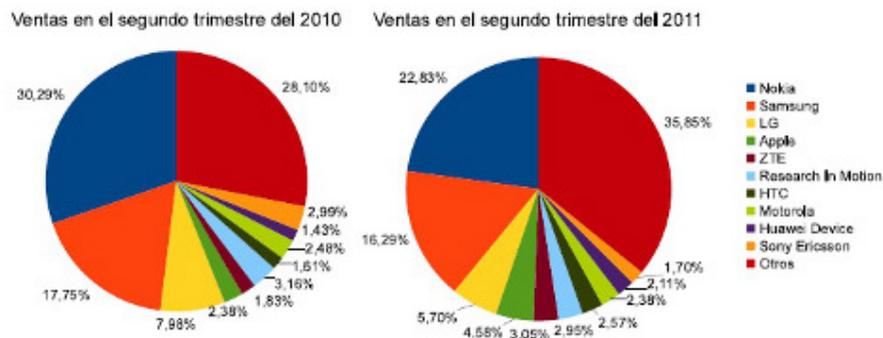


Figura 7. Gráfico de las ventas por marca de móvil 2010-2011

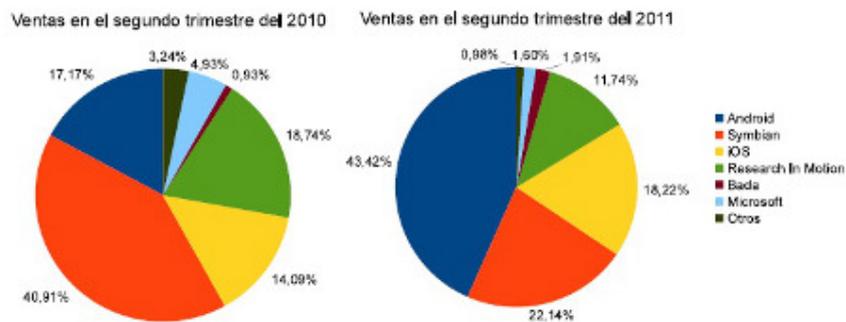


Figura 8. Gráfico de las ventas por sistema operativo 2010-2011

En el estudio realizado por Gartner Inc. se muestra una comparativa entre los dispositivos móviles vendidos a nivel mundial en el segundo trimestre del 2011 respecto al mismo periodo del 2010 [8]. Presenta los resultados por empresas fabricantes y por sistemas operativos. De los resultados de este estudio, podemos concluir que los usuarios prefieren dispositivos móviles con los sistemas operativos Android e iOS, mientras que los dispositivos con Symbian y BlackBerry OS

(Research In Motion) cada día son menos apreciados. Observando la gráfica de fabricantes, se confirma la afirmación anterior: Nokia está perdiendo cuota de mercado a pasos agigantados por haber apostado por un sistema operativo móvil que no acaba de gustar a los usuarios, como es Windows Phone de Microsoft, que al igual que Symbian (su anterior sistema operativo) y el BlackBerry OS de Research In Motion, están cediendo terreno a Android e iOS.

En cuanto al reconocimiento de voz parece que su uso en estos dispositivos se irá cada vez más generalizado, cada día existe más competencia entre los distintos sistemas operativos para ser los primeros en esta tecnología. Ésta es una de las que más interés despierta entre los desarrolladores y fabricantes que operan en el ámbito de las comunicaciones en dispositivos móviles (tablet y smartphone).

A continuación veremos las diferencias existentes entre cada uno de los sistemas operativos profundizando en el reconocimiento de voz en los tres últimos sistemas citados.

BlackBerry OS BlackBerry OS es el sistema operativo utilizado en una línea de teléfonos inteligentes denominados BlackBerry, que integran el servicio de correo electrónico móvil. BlackBerry fue desarrollado por la compañía canadiense Research In Motion (RIM). Aunque incluye las aplicaciones típicas de un teléfono inteligente (libreta de direcciones, calendario, listas de tareas, etc.), es fundamentalmente conocido por su capacidad para enviar y recibir correo electrónico de Internet accediendo a las redes móviles de las compañías de telecomunicaciones que brindan este servicio.

Symbian Symbian es un sistema operativo móvil que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y Handhelds de Psion. El objetivo de esta alianza fue crear un sistema operativo para dispositivos móviles que pudiera competir contra el de Palm y el Windows Phone de Microsoft, y actualmente contra Android de Google Inc., iOS de Apple Inc. y BlackBerry OS.

En 2009 se estableció que la fundación Symbian debía liberar la plataforma Symbian como un proyecto de software libre y gratuito. Sin embargo, en noviembre del 2010 dicha fundación abandonó sus actividades operacionales como resultado de la situación económica global y las condiciones del mercado, pasando la plataforma Symbian a estar bajo la supervisión de Nokia.

El código fuente de Symbian se encuentra disponible en la página web symbian.nokia.com, donde Nokia aclara que no se trata de una plataforma de software libre bajo la premisa: Not Open Source, just Open for Business.

Nokia, que utilizó Symbian en la mayoría de sus dispositivos móviles, anunció en febrero del 2011 que su plan estratégico para competir contra iOS y Android sería utilizar Windows Phone como sistema operativo en sus nuevos dispositivos móviles.

Windows Phone Windows Phone 7 es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. Está pensado para el mercado de consumo generalista en lugar del mercado empresarial, por lo que carece de muchas de las funcionalidades que proporciona la versión anterior. Microsoft ha decidido no hacer compatible Windows Phone 7 con Windows Mobile 6, por lo que las aplicaciones existentes no funcionan en Windows Phone 7, haciendo necesario desarrollar nuevas aplicaciones. Con Windows Phone 7, Microsoft ofrece una nueva interfaz de usuario, integra varios servicios en el sistema operativo y planea un estricto control del hardware que implementará el sistema operativo, evitando la fragmentación con la evolución del sistema.

Microsoft realizó una importante actualización a finales del 2011, que incluye Internet Explorer 9 y algunas mejoras que, según Microsoft, hacen que Windows Phone 7 sea realmente competitivo contra los sistemas operativos iOS de Apple y Android de Google. Por el momento, Windows Phone 7 continúa perdiendo cuota de mercado, como se puede apreciar en la Figura, frente a competidores como Android o iOS.

Por otro lado centrándonos más en el funcionamiento interno tenemos el SDK de Cloud Services para Windows Phone, se trata de un kit de desarrollo que nos permite aprovechar el potencial de Azure desde nuestros teléfonos de forma sencilla.

El SDK incluye un ejemplo y documentación para usarlo sobre una de las características más interesantes de Cloud Services: STT (Speech To Text o Reconocimiento de voz) que nos permite usar nuestro dispositivo WP7 como si de una grabadora se tratase para acto seguido enviar el audio a Azure y obtener como resultado una transcripción del audio enviado. Una vez enviado el audio, recibimos varias opciones de transcripción del mismo, para que podamos escoger la que mejor se ajuste a la realidad.

Por ahora solo cuenta con un tipo de diccionario de voz: independiente del contexto y en inglés. En el sample incluido en el SDK disponemos de un archivo wav pregrabado que podemos usar para probar que el servicio funciona correctamente.

iOS iOS (anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone, siendo después utilizado en todos los dispositivos iPhone, iPod Touch e iPad. Es un derivado de Mac OS X, que a su vez está basado en Darwin BSD. El iOS tiene 4 capas de abstracción: la capa del Núcleo del sistema operativo, la capa de Servicios principales, la capa de Medios de comunicación y la capa de Cocoa Touch. Todo el sistema se encuentra en la partición /root del dispositivo y ocupa poco menos de 500 megabytes. El reconocimiento de voz de iOS es el más completo de todos en cuanto a comandos. Podemos llamar, ya sea llamada normal o Facetime, y controlar el reproductor de música, todo con comandos de voz. Para usarlo, simplemente hay que mantener pulsado el botón central, esperar a que aparezca la pantalla y decir el comando. La verdad es que con tantos comandos no es sencillo que reconozca el comando adecuadamente. Además, iOS tiene un problema importante. Tanto Android



Figura 9. Iphone reconociendo la voz

como Windows Phone reconocen perfectamente cuándo dejas de hablar. iOS no, con lo que añade más ruido a los comandos y empeora más todavía los resultados.

La última versión del sistema operativo iOS de Apple, iOS 5, presente en los dispositivos portátiles de la marca, incluye Siri como una de sus novedades. Este es un asistente personal por voz que, por ahora, solo es compatible con el iPhone 4S, el modelo más actual de smartphone de Apple.

Siri permite a los usuarios realizar tareas mediante comandos de voz que comprenden el habla natural, es decir, no es necesario indicar frases cortas y directas a la aplicación, sino que también se puede hablar de forma coloquial, de igual modo que nos comunicamos con las personas de nuestro entorno mediante frases como "¿Qué tiempo va a hacer mañana por aquí?". "¿Cuál es mi agenda para hoy?". En el primer caso, Siri interactúa con el GPS para saber la posición del usuario y luego busca en la Red el pronóstico para, en consecuencia, ofrecerlo. En el segundo, muestra la agenda que tiene el usuario en iCalendar o en Google Calendar.

Este asistente virtual permite realizar otras funciones que no son solo de búsqueda de información en el teléfono del usuario o recopilación de datos en Internet. También está capacitado para realizar pequeñas tareas como escribir mensajes de texto y correo electrónico, organizar reuniones, tomar notas o realizar llamadas telefónicas. Por ahora, Siri solo está disponible en inglés de Estados Unidos, Reino Unido y Australia, así como en francés y alemán (actualmente ya dispone de español en tres tipos: España, México y Estados Unidos).

Android Android es un sistema operativo basado en el núcleo Linux diseñado originalmente para teléfonos inteligentes. Posteriormente se expandió su desa-

rrollo para soportar otros dispositivos tales como tabletas, reproductores MP3, netbook, PC, televisores, lectores de libros electrónicos e incluso se han mostrado microondas y lavadoras funcionando con Android en la Feria Internacional de Electrónica de Consumo de Las Vegas en enero del 2010 [9].

Fué desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 84 compañías de hardware, software y telecomunicaciones, que han hecho de Android su principal producto [10]. Este conglomerado de compañías se dedica al desarrollo de estándares abiertos para dispositivos móviles, con el objetivo de acelerar la innovación en estos dispositivos y ofrecer a los usuarios una experiencia móvil más rica, barata y mejor.

En el 2008 Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. Android tiene la mayor cuota de mercado desde enero del 2011, con más de un tercio del cómputo global de ventas. En los dos últimos años ha pasado a ser el sistema operativo móvil más utilizado [5].

Android también incluye reconocimiento de voz, aunque en algunos casos necesitaréis descargar alguna aplicación adicional, como “Búsqueda por voz”. Como sólo ofrece transcripción, para usarlo simplemente hay que pulsar en el botón de micrófono que aparece cuando podamos dictar al teléfono.

La primera diferencia con iOS y Windows Phone es que no hay comandos para abrir aplicaciones o llamar a nadie. Sólo podemos hablar y el teléfono transcribe lo que decimos. Esto sería útil a la hora de escribir mensajes o textos un poco largos, ya que hablar suele ser más cómodo que escribir mucho en los teclados de móvil. Hay dos opciones para activarlo: usar la búsqueda por voz manteniendo pulsado el botón de Búsqueda del teléfono o transcribir la voz a texto pulsando el botón del micrófono en el teclado virtual.

Con este sistema de reconocimiento por voz podemos, a través de comandos sencillos, controlar ciertas funciones, como enviar mensajes, llamar a contactos o negocios, usar Google Maps y navegar por Internet.

Con la salida del Samsung Galaxy S3 Samsung lanza su propia versión del popular asistente por reconocimiento de voz de Apple. Desde que Apple lanzó al mercado el iPhone 4S en octubre de 2011, numerosos desarrolladores de aplicaciones Android quedaron prendados de Siri, una herramienta que ha dado mucho que hablar durante los últimos meses.

Meses después Samsung lanza S Voice, su propio asistente por reconocimiento de voz, disponible en el Samsung Galaxy S3. El hecho de que sea Samsung el primer fabricante de smartphones Android que desarrolle su propia versión del Siri de Apple echará mas leña al fuego en la constante guerra entre ambas empresas. Apple y Samsung llevan tiempo enzarzadas en una batalla legal debido a continuas denuncias por infracción de patentes.

A pesar de estar inspirado en Siri, Samsung ha desarrollado S Voice de una manera mucho más completa. Mientras que Siri aún sigue en la versión beta,

el sistema de S Voice ya está completamente resuelto. Además, S Voice será compatible con más idiomas que Siri. De momento S Voice entenderá inglés (acento americano y británico), español (incluido el latino), japonés, alemán, francés e italiano.

S Voice reconoce diferentes comandos. El Samsung Galaxy S3 permite al usuario redactar mensajes de texto y correos electrónicos. Puedes saludar a tu Samsung Galaxy S3 para que se active, diciendo “Hola Galaxy” o “Hola Amigo”, el dispositivo reconocerá el saludo y la pantalla se encenderá. Además, es capaz de reconocer frases como “¿Qué hay en mi agenda?” para ver qué citas tienes en el calendario, modificarlas o crear de nuevas.

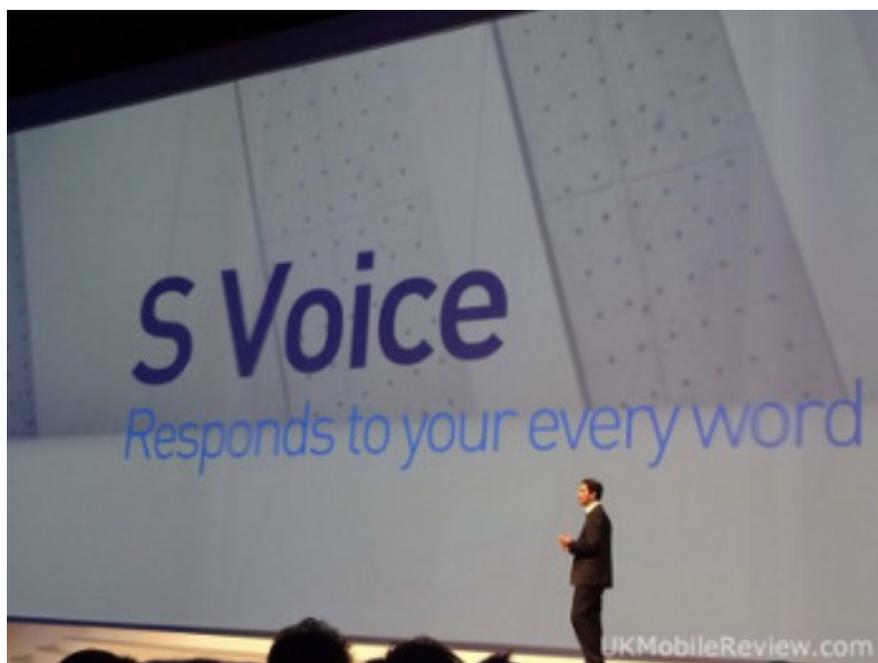


Figura 10. Sistema de reconocimiento S-Voice de Samsung

También permite activar la cámara, poner una alarma, reproducir música, consultar la información meteorológica o realizar búsquedas a través de Google. Y los usuarios que además utilicen el dispositivo como navegador GPS del coche, tendrán la posibilidad de dar órdenes al navegador para configurar las rutas sin tocar la pantalla.

Un punto a favor de Samsung en la batalla, que ha sabido aprovechar los agujeros de su principal competidor en el campo del reconocimiento de voz.

En cuanto al funcionamiento de la API de google es muy parecido al de Windows Phone. El sistema envía el archivo de sonido a los servidores y estos devuelven lo que más se ajusta al sonido enviado en una lista con las posibles respuestas, de la que más se ajusta a la que menos.

4. Arquitectura

El sistema es distribuido con una arquitectura cliente-servidor. El programa está compuesto por una aplicación cliente que corre en un dispositivo móvil que se comunica mediante sockets con un servidor de google, de modo que el dispositivo móvil sólo se encarga de la interacción con el usuario. Dicho dispositivo digitaliza la voz del usuario y le muestra la respuesta del sistema, mientras que en el servidor se realizan las tareas que requieren mayor nivel de procesado y memoria: el reconocimiento del habla y la traducción.

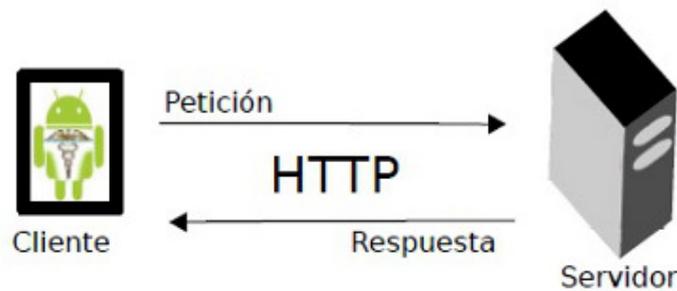


Figura 11. Funcionamiento de nuestro sistema

5. Análisis de los resultados

En este apartado nos centraremos en el vector que el servidor nos devuelve con el resultado del reconocimiento. Nuestra aplicación se encarga principalmente de tratar estos datos. Tenemos que tener en cuenta la dificultad que entraña realizar el cálculo, ya que este reconocimiento no fue pensado para reconocer números, sino que está pensado para palabras o frases así que tuvimos que adaptar los resultados a nuestras necesidades.

Para comenzar tuvimos que realizar un análisis para saber si íbamos a realizar un filtro previo de los resultados para recoger el más acertado. En un primer análisis comprobamos que los resultados que nos devolvía el servidor era exactamente el necesario para nuestro cálculo, aunque en el 80 % de los casos el primer resultado era el que menos conversiones necesitaba para obtener la operación correcta.

5.1. Análisis de los resultados

Ejemplo a analizar Tenemos un ejemplo sobre el reconocimiento y porqué tuvimos que realizar una conversión de los datos: En primer lugar probamos con la suma $5 + 3$, los resultados que obtenemos son los siguientes:

- 0: 5 más 3
- 1: cinco más 3
- 2: 5 o más 3
- 3: cinco más tres
- 4: cinco mas tres

Podríamos continuar así con los demás resultados... Como podemos observar tenemos que el comportamiento del reconocimiento es mostrar los números como texto en casi todos los casos. Así que nuestro programa siempre hace una conversión previa del texto recibido y modifica los números en texto (palabras) por dígitos. Lo mismo haría con los signos que se encuentre como texto como “más” o “mas” a +.

5.2. Conclusiones

Después de realizar multitud de ensayos, se ha comprobado que no es necesario un filtro previo de las respuestas en el vector que nos devuelve el servidor, ya que casi cualquiera es válida para realizar el cálculo y mucho más si se trata de la primera opción, por ello la decisión ha sido que sea ésta la que se utilice para la realización de la operación.

6. Conclusiones

Todos sabemos que el reconocimiento del habla en los dispositivos móviles está en pleno apogeo. Como ya hemos visto los 3 principales sistemas están en competencia continua y no es un aspecto que se descuide sobre todo en los dispositivos de gama alta, tales como iPhone 4s o Samsung Galaxy S3.

Podemos estar seguros de que el reconocimiento del habla natural será una realidad en un futuro cercano y esta capacidad quizás haga desaparecer en ciertos aparatos los tediosos teclados con teclas minúsculas que tan poco nos gustan. Debemos ser cautos puesto que hoy en día todavía estamos lejos de conseguir una correcta comprensión del lenguaje natural.

Con la aparición de esta clase de dispositivos (smartphones) se ha abierto un nuevo mundo en cuanto al reconocimiento del habla y sus posibles aplicaciones reales, su capacidad tan elevada de procesamiento, unidos a su reducido tamaño los hacen perfectos para poder llevarlos encima en todo momento, lo cual nos hace plantearnos infinitas posibilidades.

En este momento solo existen calculadoras vocales compatibles con inglés, pero no existe ninguna en el mercado que lo haga en español. Aunque sin menospreciar otras aplicaciones debo decir que después de haber comprobado el

funcionamiento de otras solamente conseguí que me imprimiese el resultado un par de veces de unos 10 intentos (eso probando operaciones simples como $1+2$) en cambio con la aplicación desarrollada se consigue un acierto de 9 sobre 10 en el caso de operaciones simples y unos 7 sobre 10 cuando complicamos la operación con decimales y números más complejos.

En definitiva, pienso que la aplicación aunque como ya hemos comentado, podríamos mejorarlo con más funcionalidades, el reconocimiento de las operaciones es satisfactorio en un porcentaje bastante elevado.

Bibliografía

- [1] M. E. Esparza Arellano and J. Bento Avalos Briseño, “Reconocimiento de voz.”
- [2] P. Marín Plaza, “Síntesis de voz y reconocimiento del habla. implementación en el robot hoap-3,” 2011.
- [3] C. A. De Luna Ortega, J. C. Martínez Romo, and M. Mora Gonzalez, “Reconocimiento de voz con redes neuronales, dtw y modelos ocultos de markov.”
- [4] ETSI, “Transmission and quality aspects(stq);distributed speech recognition; fronted feature extraction algorithm; compression algorithms,” 2000. ETSI Standard: 201 1’8 v1 1.2 Speech Processing.
- [5] Google, “Android developers.”
- [6] E. Granell Romero, “Desarrollo de un sistema de reconocimiento del habla en android.”
- [7] Wikipedia, “Dispositivo movil - wikipedia, la enciclopedia libre (2011).”
- [8] Gratner, “Gratner says sales of mobile devices in second quarter of 2011 gre 16.5 percent year-on-year; smartphone sales grew 74 percent.”
- [9] Google, “Android,” 2011.
- [10] Google, “Android ndk (2011).”

Análisis Léxico sobre los *Tweets* de *Twitter*

Astrid Paola Bográn¹, José Luis Alonso Berrocal¹ y Luis Carlos García de Figuerola Paniagua¹

¹Departamento de Informática y Automática - Facultad de Traducción y Documentación, Universidad de Salamanca. Calle de Francisco Vitoria 6-16. 37008 Salamanca, España
{abogran,berrocal,figue}@usal.es

Resumen Este documento ofrece un acercamiento sobre el Análisis léxico enfocado en los *tweets* de *Twitter*, presentando el desarrollo de una aplicación web que se podrá conectar con *Twitter* involucrando el manejo de un clasificador de texto sobre la web, y de esta manera poder descubrir las características esenciales de los *tweets* seleccionados ya sea de forma individual o masiva, todo esto ejecutándose en tiempo real o bien agregando el contenido a la base de datos que ofrece la aplicación para posteriormente tratar los *tweets* al gusto del usuario. Durante el proceso de investigación se han utilizado técnicas de *stemming* y tokenización que ayudan a procesar el *tweet* de forma más limpia y sin ruido; igualmente, para la clasificación se han creado varios diccionarios en formato *XML* basados en las áreas de ciencia y tecnología, también diccionarios que ayuden a identificar palabras vacías; para realizar la clasificación se propuso el algoritmo Naïve Bayes.

Keywords: Análisis Léxico, Tweets, Twitter, Clasificación, Algoritmo Naïve Bayes

1. Introducción

En la actualidad las redes sociales han llegado a ser de gran importancia para la sociedad, es el caso de *Twitter* que actualmente cuenta con más de 500 millones de personas conectadas en tiempo real alrededor del mundo, gracias a esta red que más que un *microblogging* se ha convertido en una fuente de noticias al último minuto, los empresarios, famosos, investigadores, instituciones gubernamentales y privadas además de un sinfín de usuarios que también se conectan a esta red para compartir sus experiencias e inquietudes o noticias de última hora; con toda esta gran cantidad de información que se encuentra en la red y con su rápido crecimiento hacen que muchos quieran indagar un poco más sobre algunos perfiles que pudieran ser importantes para la investigación moderna y también ser parte de nuevos descubrimientos, soluciones o simplemente por encontrar su potencial dentro de la red. Pensamos entonces frente a tanta cantidad de datos, crear una herramienta que ayude a analizar lexicalmente el contenido presente en tiempo real, agregando diversas características para que el usuario pueda clasificar el contenido. Cuando hablamos de Análisis léxico se nos viene a

la cabeza una gran cantidad de palabras; de eso se trata, analizar todo el contenido que encontramos en una o varias cuentas específicas pertenecientes a la famosa red social *Twitter*. Para ello se ha creado una aplicación web la cual nos ayudará a descubrir las características de un *tweet*, permitiendo que las personas con diversos intereses o conocimientos la utilicen involucrando los patrones de sus propias áreas de trabajo.

La investigación se basa en la incorporación de técnicas de minería de texto web, utilizando el algoritmo Naïve Bayes para poder generar la clasificación y el Análisis sobre cada uno de los *tweets*. Estos podrán ser escogidos de manera individual o masiva. Entre otras técnicas empleadas dentro de la aplicación encontramos la utilización de *stemmings* y tokenización, los cuales son fundamentales a la hora de realizar la clasificación. Esto se logró con la utilización e implementación de varias herramientas de desarrollo como *Visual Studio 2010 .NET*, lenguajes de programación *C#*, *HTML5*, *JavaScript*, y para conseguir que la aplicación de desarrollo cliente se conecte con *Twitter* fue necesario la utilización de una Interfaz de programación de aplicaciones (API) la cual es la encargada del conjunto de llamadas sobre las bibliotecas para este caso se utilizó la librería *LinqtoTwitter*, en la clasificación de los *tweets* fue preciso incorporar el servidor *UClasify*. Igualmente para obtener el acceso a los recursos protegidos de *Twitter* se utilizó el protocolo abierto para desarrolladores *OAuth*.

Sin todo este estudio de fondo no hubiera sido posible la obtención de los resultados y descubrimientos que se puedan realizar sobre alguna cuenta específica. El desarrollo de un sitio web como este supone, hoy en día, un avance tecnológico de alta calidad y a gran escala con expectativas futuristas referentes al manejo del contenido web pero específicamente sobre las redes sociales tales como *Twitter*.

2. Conocimiento Previo

Las Redes Sociales son básicamente unidades virtuales, donde existen usuarios que interactúan unos con otros a través de internet. Se agregan perfiles que tienen algo en común y la plataforma que se utiliza permite conectar gente que se conoce realmente o que desea conocerse por algún atributo que encuentren en común, según sea el caso de la plataforma en algunas se pueden compartir fotos, vídeos, centralizar recursos, foros de discusión, mensajes, etc [1]. Existen actualmente muchos estudios sobre las redes sociales, que se basan en gran medida en estudios individuales sobre el comportamiento de las personas, también están los enfocados en mecanismos que utiliza la red para unir todo este ambiente, donde analizan y describen los niveles de la red social.

Twitter se refiere a una red de información en tiempo real como servicio de *microblogging* que conecta con las últimas historias, ideas, opiniones y noticias sobre lo que las personas encuentran interesante. Fue creado por Jack Dorsey,

quien dispuso que la longitud máxima de cada mensaje fuera de 140 caracteres.

Twitter permite a la gente a seguir y comunicarse con los demás. Con el tiempo, esto ha demostrado ser un medio de comunicación de gran alcance debido a su funcionamiento y simplicidad. Actualmente *Twitter* cuenta con 500 millones de usuarios esto para el 2012, según un estudio realizado por *semioCast* [2]. Se ha convertido en uno de los principales canales de comunicación e información a nivel mundial, se calcula que alrededor de 400 millones de *tweets* son enviados cada día. Hoy en día existen muchos estudios sobre las redes sociales, especialmente centrados en el comportamiento de las personas, en la mayoría de los casos basados en los sentimientos, en donde también se utilizan técnicas de minería de texto utilizando una serie de algoritmos que ayudan a determinar ciertos patrones en el comportamiento de los humanos, aquí encontramos el caso del algoritmo Naïve Bayes que ayuda en la clasificación.

3. Análisis léxico

Se refiere al proceso de convertir una serie de caracteres en una secuencia de *tokens* agrupados que se leen de izquierda a derecha. El programa que realiza esta función se llama analizador léxico [3]. Para una clasificación automática de texto es importante recalcar que el Análisis léxico ayudará a extraer características que detallarán a cada documento, así como, sus clases o categorías. También en la definición de reglas que por lo general consisten en expresiones regulares, y definen el conjunto de posibles secuencias de caracteres que se utilizan para formar los *tokens* o los lexemas individuales [3].

La secuencia de caracteres que forman un *token* se llama lexema. El *token* es el símbolo general a la que un lexema pertenece. Existen dos métodos principales principales para el Análisis de una palabra:

- **Tokens:** Son secuencias de caracteres con un significado colectivo, [3] el proceso de formación de los *tokens* que permite el flujo de entrada de caracteres se llama tokenización, y el analizador léxico les clasifica de acuerdo con un tipo de símbolo.
- **Stemming:** Es un procedimiento que se utiliza para reducir una palabra a su raíz. Existen hoy en día diversos algoritmos que realizan *stemming*, como el caso del *Snowball*. Un *stem* (lema) es la porción que queda de una palabra después de retirar los afijos (es decir, los prefijos y los sufijos). Consiste en convertir todas las palabras parecidas a una forma común [4]. Entre algunas ventajas que proporciona el *stemming* es que reduce el tamaño del índice ya que el número de palabras es reducido y las variantes de la palabra se reducen a un concepto común eso mejora la importancia de la recuperación de información.

3.1. Técnicas y Librerías

API de *Twitter* 1.1. Ya que la información que fluye a través de *Twitter* es muy valiosa especialmente para nuevas investigaciones sobre el comportamiento

o descubrimiento de nuevos patrones que siguen los usuarios, uno de los objetivos de *Twitter* es lograr un equilibrio entre el fomento de desarrollo interesante y protección de los derechos tanto de *Twitter* y los usuarios. Para conceder un ambiente de desarrollo recto ha creado ciertas características para la utilización de esta tecnología en su versión 1.1 encontramos los siguientes puntos de interés:

- El usuario tiene 15 minutos después de la última ejecución dentro de *Twitter*, permitiendo 15 solicitudes por cada ventana. También entre otras convocatorias utilizadas recibirá un impulso de 180 solicitudes por ventana.
- Un requerimiento básico para todas las aplicaciones es la autenticación sobre todas sus peticiones con *OAuth* 1.0. Esto no sólo permite la visibilidad a prevenir conductas abusivas, sino que también nos ayudará a comprender mejor cómo las categorías de aplicaciones están utilizando la API. Este es un conocimiento para satisfacer mejor las necesidades de los desarrolladores a medida que se continúe evolucionando la plataforma. También toda la autenticación requiere contexto de usuario.
- Las aplicaciones cliente tienen un límite simbólico de 100,000 usuarios por *token*.

LinqtoTwitter. El proveedor *LINQ* sus siglas *Language Integrated Query* utiliza en su sintaxis un estándar para las consultas e incluye llamadas a métodos de cambios a través de la API de *Twitter*, lo ha implementado para su conexión con su API, se utiliza la sintaxis de *LINQ* estándar para las consultas e incluye llamadas a métodos de cambios a través de la API de *Twitter*. Entre las plataformas que soporta se incluyen *.NET* 3.5, *.NET* 4.0, *.NET* 4.5, *Silverlight* 4.0, *Windows Phone* 7.1, *Windows Phone* 8, *Window* 8 y *Windows Azure* [5].

Autenticación OAuth. *Open Authorization* es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite la autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web [6]. Fue lanzado hacia finales de 2007 y define un mecanismo para que una aplicación web (cliente) pueda acceder a la información de un usuario en otra (proveedor) sin tener que informar a la primera del usuario y contraseña.

Algoritmo Naïve Bayes. Es un clasificador probabilístico fundamentado en el teorema de Bayes, quien utilizó las matemáticas para encontrar la mejor clasificación y así descubrir el conocido teorema de la probabilidad [7]. Es un método importante no sólo porque ofrece un Análisis cualitativo de los atributos y valores que pueden intervenir en el problema, sino porque da cuenta también de la importancia cuantitativa de esos atributos. Su forma es simple y al mismo tiempo sorprendentemente eficaz es ampliamente utilizado en áreas como la clasificación de texto y filtrado de *spam*. Se han introducido una gran cantidad de modificaciones, por la estadística, minería de datos, aprendizaje automático, etc [8]. Un modelo bayesiano es fácil de construir, esto lo hace especialmente útil

para grandes bases de datos. Gracias a su simplicidad, este clasificador es bastante utilizado, especialmente en la recuperación de la información y manejo de texto en la web.

UClassify. El servidor Clasificación *uClassify* es impulsado por llamadas *XML* que permite a los usuarios crear y entrenar cualquier clasificador arbitrario. Se puede utilizar para categorizar las páginas web o cualquier otra tarea que requiera clasificación automática de texto. Está diseñado para manejar grandes cantidades de datos y puede procesar millones de documentos diarios [9]. El núcleo es un clasificador bayesiano Naïve con un par de pasos que mejora la clasificación. Muchos otros modelos sólo pueden responder Sí o No, pero no indican el grado. Los resultados que muestra este clasificador son probabilidades [0-1] [9] de un documento que pertenecen a cada clase. Esto es muy útil si se desea establecer un umbral para las clasificaciones. El uso de este modelo también hace que sea muy escalable en términos de tiempo de CPU para la clasificación.

Snowball. Es una librería construida en el lenguaje *Java*, en donde los algoritmos resultantes pueden representarse fácilmente. El compilador traduce a un *script* de *Snowball* (un archivo. *Sbl*) y dentro de él se crea un hilo seguro para subprocesos *ANSI* que puede ser ejecutado en *C* o un programa *Java*. Cada *script Snowball* produce un archivo y su cabecera correspondiente (con extensiones *.c* y *.h*) [10]. Se ha inventado en el lenguaje de *Snowball* reglas de algoritmos que contienen una serie de *stemmers* de lengua extranjera y vocabularios estándar de las palabras y sus equivalentes derivados donde se proporciona la raíz de cada palabra.

4. Trabajo Propuesto

Una vez conocidas las herramientas, librerías, algoritmos y términos relacionados con la lexicografía, podremos comenzar a crear una herramienta que propone el desarrollo de una aplicación que sea capaz de realizar un análisis léxico sobre los *tweets* de *Twitter*, en este caso para validar el sistema nos enfocaremos en la selección de los *tweets* basados en noticias relacionadas con ciencia y tecnología, a fin de efectuar sobre ellos diversos Análisis tanto cuantitativos como cualitativos. El elevado número de documentos de prensa disponibles hace inviable una selección manual, por lo que se decide aplicar un clasificador que seleccione de forma automática las noticias o documentos que traten de una u otra forma sobre temas relacionados en estas dos áreas. Asimismo se propone al usuario la creación de nuevas categorías descubriendo las características sobre algún perfil relacionado a una cuenta de *Twitter*.

4.1. Planteamiento del Problema

Pese a la gran cantidad de información y trabajos de investigación que encontramos en la web, hasta la fecha no se ha encontrado ningún analizador léxico

que ayude a clasificar las noticias en *Twitter*. Por esta razón se pretende diseñar y construir una aplicación que permita obtener todos los datos públicos que están involucrados en una cuenta; en este caso se proyecta recoger todos aquellos *tweets* que involucren las áreas de ciencia y tecnología, de forma masiva e individual, esto con el fin de poder procesar el *tweet* haciendo uso de técnicas de *stemming*, tokenización, y clasificación del *tweet* descubriendo la correcta categorización entre las dos áreas. Se trata también de que el mismo usuario pueda crear y generar su propia clasificación dependiendo el área de su interés.

4.2. Objetivo General

Construir un analizador léxico que permita el ingreso a una cuenta de *Twitter* con el apoyo de diferentes técnicas que ayuden a identificar y clasificar *tweets* referentes a las áreas de ciencia y tecnología y que también el usuario pueda realizar las mismas operaciones de clasificación utilizando su propia clasificación.

4.3. Análisis del Sistema

El proceso que se pretende llevar a cabo describe un flujo comenzando desde lo que será el corazón o núcleo del sistema con la creación de un sitio web conectado a través de un *webservice* que responderá a las peticiones del sitio web cliente. Para la autenticación con *Twitter* se utilizó *OAuth* que hasta la fecha es la forma más común de autenticación de recursos en aplicaciones de *Twitter*, continuando con la iteración se decide utilizar el lenguaje de programación *C#* ya que es un lenguaje de programación independiente y de adecuación para escribir aplicaciones de cualquier tamaño; es el caso de este proyecto donde se desarrolla una aplicación web con la combinación de *Visual Studio .NET* 2010 utilizando el *framework* 4.0, la biblioteca *LinqtoTwitter* y la API de *Twitter*. La Figura 1 describe el flujo de nuestro sistema.

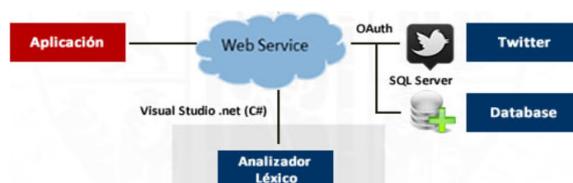


Figura 1. Flujo del Sistema

Comenzando con el desarrollo del sistema nos enfocamos en el Análisis léxico basado en la clasificación de los *tweets* en las áreas de Ciencia y Tecnología; estableciendo parámetros a seguir para realizar cálculos y obtener algún descubrimiento sobre casos particulares. Nos centramos en las siguientes premisas:

1. Estudio sobre terminología utilizada en *Twitter*.
2. Análisis de entidades.
3. Observación sobre la recuperación de la información.
4. Visualización sobre los resultados obtenidos de los casos de estudio.

Se trabajará con cada una de las palabras contenidas dentro de un *tweet*, considerando como palabra una secuencia ininterrumpida de letras [11]. Se removerán las palabras contenidas en una lista estándar denominada palabras vacías para el español. El sistema realizará esta limpieza que es necesaria para la extracción de palabras de un *tweet*, lo que aplicaremos en nuestro Análisis y que se detalla a continuación:

- Eliminación de acentos.
- Conversión a minúsculas.
- Eliminación de cualquier carácter que no sea alfanumérico.

El texto resultante se someterá a un proceso de normalización o *stemming*, para el cual se debe trabajar con la librería basada en el *stemmer de SnowBall*; y como último punto también se deben eliminar aquellas palabras cuya longitud sea menor a cuatro caracteres.

4.4. Diseño del Sistema

El sistema consta de una serie de ventanas que básicamente están distribuidas en los siguientes 5 pasos principales:

Inicio. La ventana inicial describe la bienvenida y una breve explicación comentando el objetivo principal referente al Análisis léxico sobre los *tweets* de *Twitter*, indicando también que es una aplicación con un diseño sencillo y código entendible. Podemos ver esta ventana en la Figura 2.



Figura 2. Pantalla de Bienvenida

Autenticación. Este paso se realiza con la utilización de *OAuth* como mecanismo de autenticación para *Twitter* proporcionando un método para que los clientes accedan a los recursos del servidor en nombre del propietario (como un cliente diferente o un usuario final). Seguido de la autenticación podremos ver el resumen de la cuenta visualizando los resultados: nombre de usuario, total de amigos, seguidores, actualizaciones y favoritos.

Verificando la conexión a través del Habla. Una vez autenticados con la cuenta personal de *Twitter*, la aplicación se conecta a través del *web service* mostrando la pantalla para la prueba de la conexión con *Twitter* y para hacer la aplicación más dinámica se ha decidido trabajar con el habla.

Análisis léxico. Esta pantalla detalla una parte de nuestro objetivo principal. En ella encontraremos varias características que nos ayudaran a identificar el proceso que debemos seguir para realizar el Análisis léxico sobre un *tweet* que podremos seleccionar para posteriormente ser evaluado.

Clasificación y Trabajo con Diccionarios. Aquí se definen los diccionarios que han sido creados con la ayuda de la herramienta *wordnet* y posteriormente desarrollados en un formato *XML* para que puedan ser leídos por el clasificador y categorizar adecuadamente los *tweets* seleccionados. En esta pantalla encontramos básicamente un botón principal Clasificador que se encarga de clasificar los *tweets* de forma automática según las categorías ciencia y tecnología. Este botón depende de la selección que se haya hecho en la pantalla anterior, es decir que si se quieren clasificar los *tweets* de forma masiva se pueden utilizar los que se encuentran dentro de la base de datos o si se quiere clasificar en tiempo real mediante la conexión con *Twitter*.

Gestión de Tweets. Esta sección de la aplicación web cuenta con una serie de características donde se puede trabajar con diferentes cuentas de usuario aún sin estar autenticados donde podrán buscar los *tweets* sobre cualquier perfil que no sea privado.

En esta pantalla podremos ingresar la mención o el usuario de alguna cuenta y agregar la cantidad de *tweets* que desee visualizar sobre la cuenta solicitada. Además, la aplicación también permite guardar los *tweets* seleccionados en un archivo de texto plano o en directamente en la base de datos, y permite crear y entrenar un nuevo clasificador. Ver Figura 3.

4.5. Desarrollo del Sistema

A fin que el Análisis y el diseño del sistema funcionen correctamente, se definieron varios proyectos, donde cada uno de ellos contiene una funcionalidad específica, las cuales se detallarán en esta sección.

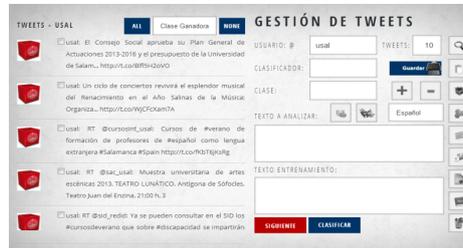


Figura 3. Resumen de cuenta

Conexión con *Twitter*. Como lo hemos mencionado anteriormente, la conexión con *Twitter* se realiza con la ayuda de las credenciales proporcionadas por *OAuth*. Esta conexión se realiza a través de las siguientes líneas de código (ver Listado 2.1)

Listado 2.1. Código conexión

```
static WebAuthorizer auth;
public static \emph{Twitter}Context obtenerContexto(ref WebAuthorizer _auth)
{
    const string OAuthCredentialsKey = "OAuthCredentialsKey";
    \emph{Twitter}Context twitterCtx;
    IOAuthCredentials credentials = new SessionStateCredentials();
    if (credentials.ConsumerKey == null || credentials.ConsumerSecret == null)
    {
        credentials.ConsumerKey =
            ConfigurationManager.AppSettings["twitterConsumerKey"];
        credentials.ConsumerSecret =
            ConfigurationManager.AppSettings["twitterConsumerSecret"];
    }
}
```

Consulta Sobre los Detalles de la Cuenta. La Tabla 1 describe los detalles que pueden ser obtenidos de una cuenta de *Twitter*.

Clasificación Masiva. La siguiente codificación (Listado 2.2) muestra la clasificación que se realiza masivamente en la pantalla Diccionarios de la aplicación web.

Listado 2.2. Clasificación Masiva de los Tweets

```
var listtweets =
from tweet in twitterCtx.Status
where tweet.Type == StatusType.Home && tweet.Count == 10
select tweet;

foreach (var tw in listtweets)
{
    string tweetText = limpiarCadena(tw.Text);
}
```

```

List<uClassifyResponse> uRes =
    ClassificarTexto(tweetText, "TechnologyorScience", false, true);
string classifyName = ((Math.Round(uRes[0].percentage, 2) >
    Math.Round(uRes[1].percentage, 2)) ? uRes[0].className :
    ((Math.Round(uRes[0].percentage, 2) < Math.Round(uRes[1].percentage,
    2) ? uRes[1].className : "None"));

tweetList.Add(new Tweet(nombre: classifyName, imageURL:
    tw.User.ProfileImageUrl,
ultimoTweet: limpiarCadena(tw.Text), seguidores:
    Convert.ToInt32(tw.User.FollowersCount),
estados: Convert.ToInt32(tw.User.StatusesCount), siguiendo:
    tw.User.Following, retweeted:
    Convert.ToBoolean(tw.Retweeted));
}

```

Trabajando con *Stemming*. Consiste en una biblioteca de clases que no es más que una dll para poder ser utilizada con *.NET*. Específicamente se ha desarrollado la traducción a *.NET* de la librería que está basada en el lenguaje *java*. Una vez que tenemos nuestra propia dll, podremos utilizarla específicamente para la limpieza de los *tweets* o el contenido que se pretenda analizar.

La librería cuenta con alrededor de quince idiomas diferentes entre ellos el inglés y español, que son básicamente con los que trabajaremos para realizar la clasificación de los *emphTwitter* ya que nuestros diccionarios están basados en estos dos lenguajes. Utilizamos el *Stemming* porque es uno de los métodos más precisos para agrupar palabras con un significado similar, que corresponde al caso de nuestros dos clasificadores basados en ciencia y tecnología.

Creación de Diccionarios. El segundo caso define otra biblioteca de clases que consiste en gestionar los diccionarios que se refieren a la definición de una librería para la creación y manipulación de los archivos en *XML* que son la clave fundamental en el proyecto donde están contenidas todas las palabras para la clasificación en las áreas de ciencia y tecnología basadas en los *tweets*. Los

| Nombre | Propósito | Tipo de Dato | Requerido |
|------------------|---|------------------|-----------|
| EndSessionStatus | Respuesta de la solicitud para finalizar la sesión | String | No |
| RateLimitStatus | Clasificación del texto limitada | RateLimit Status | No |
| Settings | configuración de la cuenta | Settings | No |
| Totals | totales actuales | Totals | No |
| Type | Tipo de cuenta consulta | Account Type | No |
| User | Usuario devuelve la veracidad de las Credenciales de la Consultas | User | No |

Tabla 1. Filtros/ Parámetros, detalles de la cuenta.

diccionarios han sido creados con la ayuda de *WordNet*, que contiene una gran base de datos léxica en el idioma Inglés donde encontramos sustantivos, verbos, adjetivos y adverbios que se agrupan en conjuntos de sinónimos cognitivos (*synsets*), cada uno expresando un concepto distinto. Los términos en *Synsets* están vinculados entre sí mediante relaciones conceptuales, semánticas y léxicas [12]. Gracias a esta herramienta se logró la extracción de las palabras relacionadas con ciencia y tecnología que a su vez han pasado por un proceso de traducción de forma manual, y para poder ser utilizadas en nuestro sistema se realizó la construcción de los diccionarios en formato *XML*.

5. Resultados

Los resultados están basados en el estudio de un total de 3690 *tweets* que han sido almacenados en la base de datos creada. Para comprobar que el algoritmo de clasificación de texto Naïve Bayes y las técnicas léxicas funcionan correctamente en el sistema, se decidió realizar dos estudios, uno sin utilizar la técnica de *emphstemming* y el segundo basándonos en esta técnica y al mismo tiempo utilizando todo el tratamiento de limpieza que realiza la aplicación.

Clasificación de *tweets*:

- Ciencia: 2050 *tweets*.
- Tecnología: 627 *tweets*.
- Otros considerados como ambiguos: 1013 *tweets*.

Prueba Sin Utilizar Stemming. De la clasificación sin utilizar las técnicas de Análisis léxico como el *stemming* o alguna otra técnica de limpieza sobre los *tweets*, el algoritmo Naïve bayes logra una exactitud de 52.14 %, y una precisión por clase con los siguientes valores (Tabla 2): En la Figura 4 podemos visualizar

| | Ciencia | Tecnología | Ninguno | Totales |
|------------|---------|------------|---------|-------------|
| Ciencia | 1133 | 604 | 313 | 2050 |
| Tecnología | 185 | 364 | 78 | 627 |
| Ninguno | 301 | 285 | 427 | 1013 |
| Totales | 1619 | 1253 | 818 | 3690 |

Tabla 2. Matriz de Confusión sin Stemming.

los resultados gráficamente. Como resultado porcentual generado por el clasificador se encontró que de los 2050 *tweets* de ciencia, se clasificaron correctamente 55 %, equivalente a 1133 *tweets*. Para la clase de tecnología se obtuvo un resultado positivo del 58 % que corresponde a 364 sobre un total de 627 *tweets*.

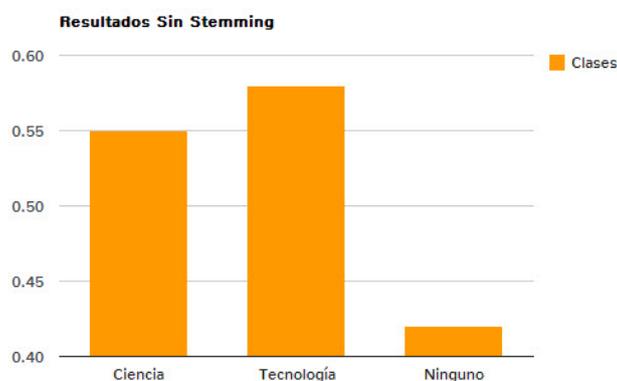


Figura 4. Resultados Sin Utilizar Stemming.

Finalmente en los *tweets* sin clasificación se obtuvo una precisión del 42%. Esto quiere decir que de 1013 *tweets* únicamente se clasificaron exitosamente 427 *tweets*. La Tabla 3 puntualiza el resultado según la precisión y el *recall*.

| | | | |
|------------------|----------------|-------------------|----------------|
| Precisión | Ciencia - 55 % | Tecnología - 58 % | Ninguno - 42 % |
| Recall | Ciencia - 70 % | Tecnología - 29 % | Ninguno - 52 % |

Tabla 3. Precisión y Recall, Sin *Stemming*.

Prueba Utilizando *Stemming*. Utilizando las técnicas de *stemming*, limpieza y tokenización, la matriz de confusión dio como resultado una precisión del 84% junto con la generación de los siguientes resultados (Tabla 4):

| | Ciencia | Tecnología | Ninguno | Totales |
|------------|---------|------------|---------|-------------|
| Ciencia | 1745 | 170 | 135 | 2050 |
| Tecnología | 32 | 583 | 12 | 627 |
| Ninguno | 145 | 82 | 786 | 1013 |
| Totales | 1922 | 835 | 933 | 3690 |

Tabla 4. Matriz Utilizando *Stemming*.

En la Figura 5 podemos observar los resultados generados al utilizar *Stemming*. El clasificador Naïve Bayes indicó que de los 2050 *tweets* catalogados

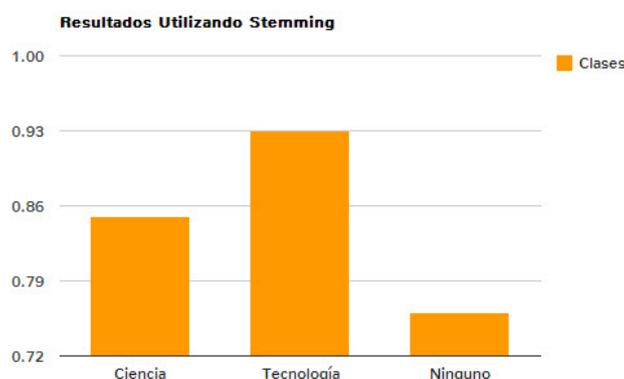


Figura 5. Resultados Utilizando *Stemming*.

como ciencia, se clasificaron 1745 correctamente dando como resultado porcentual 85 % para la clase tecnología un 93 % teniendo en cuenta que de los 627 se han clasificado 853 *tweets* correctamente. En último lugar los *tweets* fichados como ambiguos 786 han sido clasificados correctamente dando como porcentaje un 77 % (ver Tabla 5).

| | | | |
|------------------|----------------|-------------------|----------------|
| Precisión | Ciencia - 85 % | Tecnología - 93 % | Ninguno - 77 % |
| Recall | Ciencia - 91 % | Tecnología - 70 % | Ninguno - 84 % |

Tabla 5. Precisión y Recall, utilizando *Stemming*.

5.1. Interpretación de los Resultados

En la primera prueba el clasificador obtiene una precisión del 52.14 % contra la segunda prueba que dio un resultado de 84 % pre-procesando los datos con *Stemming*. Sin embargo, la precisión en el campo de la Recuperación de Información difiere de las definiciones de exactitud y precisión en otras áreas.

Por tanto para determinar la exactitud del clasificador en ambos conjuntos, se utiliza la medida F1. A continuación se describe en qué consiste la precisión y la recuperación.

- Precisión: es el número de resultados correctos dividido por el número de todos los resultados devueltos.
- Recuperación: es el número de resultados correctos dividido por el número de resultados que deberían haber sido devueltos.

La puntuación de F1 se puede interpretar como una media ponderada de la precisión y la recuperación, donde una puntuación F1 alcanza su mayor valor en 1 y peor puntuación en 0. La Tabla 6 muestra las fórmulas que se han utilizado para el cálculo de los resultados. Las Tablas 7 y 8, muestran la clasificación del

| | |
|------------------|---|
| Accuracy | $(\text{true positives} + \text{true negatives}) / (\text{total examples})$ |
| Precision | $(\text{true positives}) / (\text{true positives} + \text{false positives})$ |
| Recall | $(\text{true positives}) / (\text{true positives} + \text{false negatives})$ |
| F1 score | $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$ |

Tabla 6. Fórmulas.

número total de empltweets verdaderamente bien clasificados ya que la búsqueda del texto en los documentos seleccionados proporcionan una única medición para el sistema.

Ciencia 88 %
Tecnología 80 %
Ninguno 81 %

F1 83 %

Tabla 7. Resultados utilizando *Stemming*.

Ciencia 61 %
Tecnología 38 %
Ninguno 46 %

F1 48 %

Tabla 8. Resultados sin *Stemming*.

6. Conclusiones

De los resultados obtenidos podemos concluir que la utilización del *Stemming* como técnica de preprocesamiento de los datos mejora considerablemente el desempeño del clasificador, el Análisis léxico para analizar *tweets* resulta de gran utilidad para determinar a que categoría pertenece la información publicada por los usuarios.

El clasificador Bayesiano presenta un excelente rendimiento, debido principalmente a su robustez frente al ruido. Es un algoritmo muy práctico y sencillo de utilizar, especialmente porque hace múltiples suposiciones para simplificar el problema. Los cálculos probabilísticos realizados fueron acertados, la probabilidad de cada una de las palabras $P(W)$ fue condicionalmente independiente dado el valor de las clases (Ciencia, Tecnología o Ninguno), así mismo con las entradas sobre nuevas clases también se reflejó un número de aciertos aceptable. El clasificador también demostró ser muy potente en cuanto al volumen de los datos que pueden procesar en tan poco tiempo. Resultó muy ventajoso clasificar cada uno de los *tweets* por su contenido en lugar de hacerlo por sus *hashtags*, ya que no todos los *tweets* tienen *hashtags*, y en muchas ocasiones los mismos usuarios les asignan una categoría que no es del todo cierta, dado eso y a lo dinámico que puede resultar la utilización de un *hashtag* (es difícil predecir cuál será su estructura). Como resultado se determinó que era más útil para la investigación trabajar con el contenido del *tweet*.

La enorme cantidad de información y su crecimiento exponencial en *Twitter* representa un reto para el Análisis y la recuperación de información, de la investigación realizada, se deriva el desarrollo de una aplicación capaz de utilizar diferentes técnicas de minería de texto, la aplicación es un primer acercamiento al Análisis léxico en *Twitter*, la investigación puede despertar un gran interés en profundizar en esta temática.

Bibliografía

- [1] Ronald S Burt, M K, *Social Network Analysis Foundations and Frontiers on Advantage*. University of Chicago, University College London, University of Cambridge, 2013.
- [2] Semiocast, 2012. [Internet; descargado 20-abril-2013: <http://semiocast.com>].
- [3] J. Z. Maggie Johnson, *Lexical Analysis*. Stanford University, California, 2008.
- [4] K. D. Benavides, *Procesamiento de Texto*. 2008.
- [5] linqtotwitter, “2006-2013 microsoft,” 2013. [Internet; descargado 20-abril-2013: <http://linqtotwitter.codeplex.com>].
- [6] OAuth, 2007. [Internet; descargado 20-abril-2013: <http://oauth.net>].
- [7] M. Bramer, “Introduction to classification: Naïve bayes and nearest neighbour,” in *Principles of Data Mining, Undergraduate Topics in Computer Science*, pp. 21–37, Springer-Verlag, London, 2013.
- [8] V. K.-H. XindongWu, “Top 10 algorithms in data mining,” in *IEEE International Conference on Data Mining*, p. 37, Verlag London Limited, Springer, 2007.
- [9] J. Kagström, *uClassify Classification Server Manual*. Spring, SMid Sweden: Mid Sweden University, 2005.
- [10] M. Porter, *Strategy and the Internet*. Harvard Business Review, 2001.
- [11] García, C., Alonso J. L, “Clasificación automática de documentos. un caso práctico,” tech. rep., Instituto Universitario de estudios en Ciencia y Tecnología. España: Universidad de Salamanca, 2012.
- [12] G. A. Miller, 2012. [Internet; descargado 16-mayo-2013: <http://wordnet.princeton.edu>].

Soluciones Visuales Interactivas aplicadas a Grandes Volúmenes de Datos de Entornos 3D de Aprendizaje y Prácticas

Juan Cruz Benito¹, Roberto Therón Sánchez¹ y Emiliana Pizarro Lucas²

¹ Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{juancb, theron}@usal.es

² Servicio de Inserción Profesional, Prácticas y Empleo, Universidad de Salamanca. Patio de Escuelas 1. 37008, Salamanca, España
mili@usal.es

Resumen Este documento presenta el trabajo de investigación asociado al desarrollo de una propuesta de Visualización de Información apropiada para representar grandes cantidades de datos sobre movimientos de usuarios entre regiones de un Mundo Virtual de Aprendizaje y Prácticas y los patrones de exploración que en ellos se producen.

Este trabajo abarca todas las fases que incluyen en la propuesta de visualización: desde la identificación de las fuentes de datos, la extracción y procesado de la información, la aplicación de algoritmos de Minería de Datos para extraer conocimiento de la información, hasta la propuesta final de una solución visual que represente la información y el conocimiento que encierran los datos, y el conjunto de características y mecanismos que debe incorporar dicha visualización.

Keywords: Visualización de Información, Hierarchical Edge Bundles, HEBATT, Mundos Virtuales, Teleports, Minería de Datos

1. Introducción

Las tecnologías interactivas *online* o disponibles a través de Internet ofrecen la posibilidad de llegar a multitud de personas, sin importar la edad, cultura o nivel social. Una de esas tecnologías, es la conocida como Mundos Virtuales, o Entornos Virtuales Multi-Usuario (EVMU o MUVES). En un ellos, el usuario dispone de *alter ego* virtual o *avatar* que le representa en este entorno simulado. A través de estos *avatar*, al usuario se le proporcionan una serie de posibilidades de comunicación, colaboración, e interacción casi de cualquier tipo, tanto con el entorno virtual 3D como con el resto de usuarios que tienen presencia en dichas plataformas. Además, ofrecen un entorno *immersivo*, donde los usuarios experimentan una sensación de estar dentro de las situaciones y los eventos que ocurren, de un modo totalmente distinto a cómo se perciben en otras plataformas como la Web, u otros servicios informáticos en la Red.

El uso de este tipo de plataformas da lugar a la generación de grandes cantidades

de diversos tipos de datos, desde datos relacionados con la comunicación entre usuarios, a datos relacionados con los patrones de uso en objetos, o como es el caso del análisis, datos relacionados con el movimiento entre distintos territorios virtuales por parte de los usuarios. Debido al carácter inmersivo de la experiencia virtual de los usuarios, estos datos que se generan suelen ser indicadores fiables sobre distintos aspectos de la personalidad o comportamiento de los usuarios, ahí radica el interés del análisis de este tipo de entornos.

2. Objetivos

El objetivo principal de este trabajo de investigación es el estudio, comprensión y aprovechamiento de los datos sobre los movimientos de usuarios entre distintas regiones de un Mundo Virtual de Aprendizaje y Prácticas, para poder definir una visualización que permita a los distintos *stakeholders* implicados en el análisis de datos de un Mundo Virtual, descubrir conocimiento oculto en los grandes volúmenes de datos que se generan con información sobre los movimientos de usuarios entre distintos territorios virtuales, y que esta visualización pueda servir como Sistema de Soporte de Decisiones [1] sobre la actividad que realizan los usuarios dentro de un sistema 3D.

Además, se pretende dar un nuevo punto de vista sobre las posibilidades de visualización de datos extraídos de un Mundo Virtual, proponiendo un sistema bien construido, con un diseño atractivo, y que resulte útil y sencillo para el usuario, que abandone la línea típica de investigación de las visualizaciones de movimientos dentro de un Mundo Virtual [2] [3].

3. Fuentes y tratamiento de Datos

Para la obtención de datos de un Mundo Virtual de Aprendizaje y Prácticas, es necesario establecer conocer las fuentes de datos que se deben emplear en la extracción, y qué estrategia de extracción de información es conveniente aplicar [4] [5]. Una vez obtenidos los datos deseados del Mundo Virtual, es necesario procesarlos para extraer conocimiento de ellos. El procesamiento puede ser mediante técnicas clásicas, como puede ser la aplicación de procedimientos estadísticos simples, o mediante técnicas de minería de datos. En estas técnicas, destaca el algoritmo *Predictive Apriori*, el cual permite obtener patrones y reglas ocultas en los datos [6] [7], que más tarde se emplearán como herramienta de apoyo en las tareas de análisis, de modo que faciliten la comprensión del conjunto de datos y la obtención de conocimiento por parte del usuario.

El modelo de datos que se ha usado para el diseño de la visualización, ha sido extraído del Mundo Virtual de Aprendizaje y Prácticas USALSIM [8] [9].

4. *HEBATT: Hierarchical Edges Bundle Applied to Teleport Tracking*

Como se ha comentado previamente, el objetivo principal de este trabajo es proporcionar una visualización que permita analizar los movimientos entre territorios virtuales, y además aportando una solución novedosa y útil. Como resultado de este proceso de investigación, se ha llegado a la visualización denominada *HEBATT: Hierarchical Edge Bundles Applied to Teleport Tracking*. Como su propio nombre indica, se parte de la propuesta de Danny Holten sobre la visualización de datos jerárquicos *Hierarchical Edges Bundle* [10] [11]. En esta visualización de tipo radial, se codifica mediante variables visuales en forma de *arco* una jerarquía de datos, y se trazan líneas de tipo *spline* entre las distintas instancias de datos de dicha jerarquía que tengan relaciones entre sí [12].

Partiendo de esta idea resumida sobre el tipo de visualización *Hierarchical Edge Bundles*, se puede decir que en el conjunto de datos que se utiliza en el caso del Mundo Virtual, la jerarquía de datos se compone de una serie de islas o regiones, los grupos de usuarios que se integran en ellas, y los usuarios que pertenecen a esos grupos. Las líneas (curvas) que unen los distintos puntos de la estructura jerárquica representan los distintos movimientos que realizan los usuarios entre las distintas regiones.

La distribución de la jerarquía en el caso concreto de los datos extraídos del

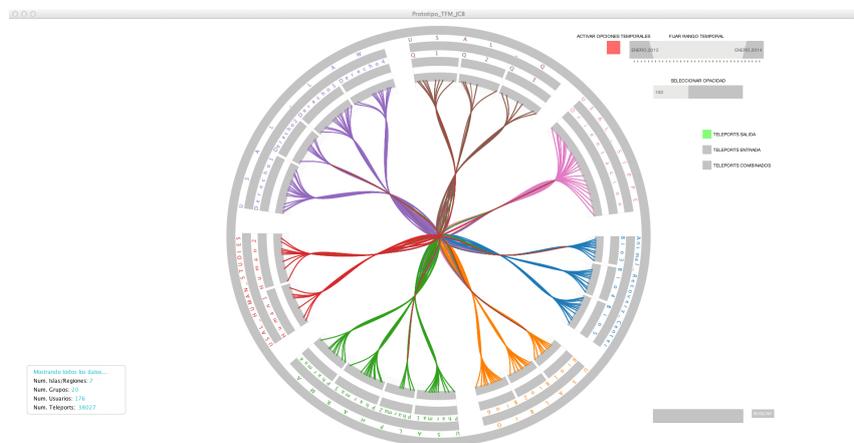


Figura 1. Vista general de la interfaz de la visualización HEBATT

Mundo Virtual USALSIM se realiza a través de cuatro niveles de arcos: un nivel superior que encierra toda la visualización y representa el conjunto completo, es decir, la vista general de todos los datos, de modo que si se selecciona en

cualquier momento ese arco exterior, se pasa a una vista general, como se ve en la figura 1. El nivel inmediatamente inferior a este que engloba todos los datos, es el que se corresponde con los distintos terrenos o islas que componen el Mundo Virtual; que en el caso del Mundo Virtual USALSIM, son siete territorios. Por debajo de ese nivel de la jerarquía y arcos de la representación visual, se presentan las distintas instancias de grupos de usuarios que existen en el Mundo Virtual. Cada una de las instancias de grupo se sitúa bajo el arco que representa la región o isla que sirve como base o localización de las actividades de aprendizaje y prácticas que realizan los usuarios de dicho grupo. Por último, el nivel más bajo de la jerarquía de datos, y nivel más bajo en la estructura radial, es para los arcos que representan a los distintos usuarios. En el caso de los usuarios hay un pequeño detalle a tener en cuenta: un usuario puede pertenecer a varios grupos, con lo cual, en una misma isla un usuario puede representarse varias veces, en función de la cantidad de grupos que pertenecen a esa región en los cuales se encuentra inscrito.

Como se observa en la figura 1, la interfaz de la aplicación cuenta con diversos elementos visuales que se organizan por toda la pantalla [13]. En la zona central se encuentra la visualización *HEBATT* propiamente dicha, la cual cambiará en función de la interacción que se de con el usuario. En la zona derecha de la interfaz se encuentran diversos elementos que permiten distintos tipos de interacción con la visualización, entre estos elementos se observan distintos grupos de botones, *sliders*, *sliders con funcionalidades de rango*, o un campo que permite introducir texto y realizar búsquedas en los datos. El uso de estos elementos se explicarán de forma más detallada en el apartado *Mecanismos de Interacción en la Visualización HEBATT* (4.9).

La zona izquierda de la visualización se ha reservado para la visualización de información textual relacionada con los datos que se estén mostrando dentro del *HEBATT*. Este texto se codificará mediante colores, que ayuden al usuario a la relación de los elementos visuales (mediante la característica de etiquetado, como se explicará en el punto 4.8.

En los siguientes sub-apartados se comentará qué tareas de análisis soporta la visualización y cómo es el comportamiento de la visualización en función de los datos que se quieran visualizar.

4.1. Tareas de Análisis que debe soportar la visualización *HEBATT*

Antes de mostrar el comportamiento de la visualización en cada caso concreto de uso, es preciso comentar qué tareas de análisis se permite a los usuarios:

- *Visualización general de los teleports realizados en cualquier instante de tiempo*
- *Visualización general de los teleports realizados en un instante de tiempo concreto*
- *Visualización de los teleports de salida de todas las regiones que componen el Mundo Virtual*
- *Visualización de los teleports de entrada a cualquiera de las regiones que componen el Mundo Virtual*

- Visualización de los teleports de entrada y salida de cualquiera de las regiones
- Visualización de los teleports de salida de una región que realizan los usuarios inscritos en un grupo
- Visualización de los teleports de salida de una región que realiza un usuario concreto
- Visualización de los teleports de salida efectuados por los usuarios de una región concreta en un rango de tiempo seleccionado
- Visualización de los teleports de salida efectuados por los usuarios de un grupo concreto en un rango de tiempo seleccionado
- Visualización de los teleports que realiza un usuario en un rango de tiempo seleccionado
- Visualización de los patrones de teleports de salida de una región
- Visualización de los patrones de teleports de entrada de una región
- Visualización de los patrones de teleports de un usuario
- Visualización de los teleports que ha realizado un usuario durante su tiempo de uso del sistema, pudiendo recorrerlos a través de una escala temporal

4.2. Tratamiento visual y lógico del volumen de datos a representar

En la propuesta de visualización *Hierarchical Edge Bundles (HEB)* original de Holten [10], ya se planteaban una serie de cuestiones que preparaban a la visualización para soportar grandes cantidades de datos, y se hacían diversas recomendaciones sobre cómo implementar la visualización. A continuación se comentan cuáles de las recomendaciones de Holten han sido utilizadas y qué otros mecanismos se han empleado en la visualización para gestionar grandes cantidades de datos:

1. **Uso de *splines*:** Holten, en sus publicaciones acerca de la visualización *HEB*, recomienda encarecidamente usar líneas que usen puntos de control y tensores para la formación de su curvatura, de esta forma es posible que se agrupen las líneas de forma efectiva y no saturen la visualización de líneas que no aporten información y oculten conocimiento. En el caso del prototipo software implementado, se ha hecho uso de líneas *splines* continuas [14], que usan tensores y puntos de control en su formación, aunque no permiten el control de estos tensores para la deformación de la curva en función de las necesidades que se tengan en la visualización.
2. **Uso de características de transparencia:** De la misma forma que en la característica anterior, Holten también recomienda el uso de características de transparencia, de forma que se pueda distinguir dentro de los conjuntos agrupados de líneas cuáles están compuestos por una cantidad mayor o menor de líneas (de relaciones en los datos), como se puede comprobar en la figura 2.
3. **Uso de líneas *resumen*:** Para la composición de la agrupación de los ejes hay dos opciones: una consiste en tener estructuras de datos que contengan cada teleport que se haya producido en el entorno virtual y toda su información asociada. La otra opción, es perder información como la referente a la

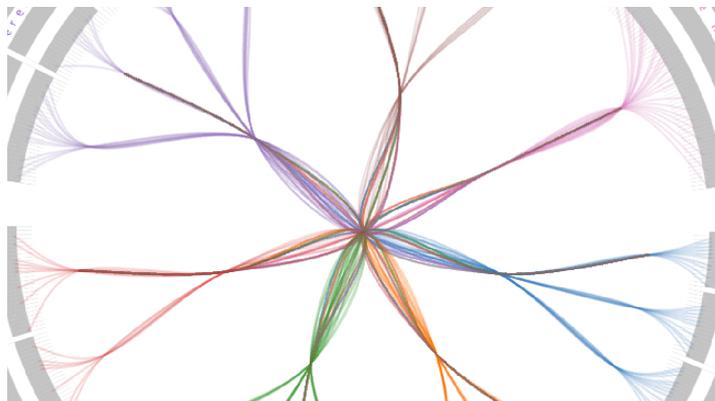


Figura 2. Transparencia aplicada en la agrupación de líneas de la visualización HE-BATT

fecha u hora de los teleports, y agrupar por cada tipo de teleport (definido por origen, destino y usuario), para así dibujar una sola línea *resumen*, que pueda expresar visualmente la cantidad de líneas que componen esta línea resumen. Esta representación visual se expresa también mediante la transparencia u opacidad de la línea, y esta opacidad se calcula de la siguiente forma:

$$\text{Opacidad Línea} = \text{factor Transparencia} * \text{num. de teleports del mismo tipo}$$

Por lo tanto, el resultado visual de estas líneas *resumen* es igual que en el caso de dibujar todas las líneas que componen cada agregación, y la carga de trabajo que se realiza en cuanto al dibujado, es menor en una cantidad proporcional al número de líneas que no se dibujan.

4. **Uso de múltiples estructuras de datos:** Como se indica parcialmente en el punto anterior de dibujado de líneas *resumen*, para poder tener un rendimiento mejor de la visualización, es necesario implementar distintas estructuras de datos que ayuden a mejorar el rendimiento de la visualización, de modo que a través de la segmentación y agrupación conveniente de los datos se pueda abarcar el dibujado de grandes conjuntos de datos. En la implementación de este prototipo se han usado tanto estructuras *ArrayList* con conjuntos completos de teleports y teleports de tipo resumen, como estructuras *HashMap* para acceder a variables que se usan de forma recurrente de un modo más rápido que con los algoritmos clásicos de búsqueda.

4.3. La visualización *HEBATT* y la escala temporal

Como se ha comentado anteriormente, entre los datos que se obtienen en la extracción de teleports de los archivos log de OpenSim, se encuentra información relevante a fechas y horarios. Por lo tanto esta información puede (y *debe*) ser

utilizada para poder realizar visualizaciones de datos segmentados por características temporales. El autor MacEachren, en 1995 [15], planteaba una serie de preguntas que toda visualización que trabaje con datos debería poder responder. Estas preguntas básicamente ofrecen una guía para el análisis de datos con características temporales, y se encargaban, por ejemplo, de evaluar la existencia de un elemento en un momento dado, o cómo se distribuye la existencia de un objeto en el tiempo, qué tasas de cambios aparecían de los datos, o qué secuencias de cambios se daban, etc.

Tomando este conjunto de preguntas como guía de desarrollo, se han implementado varias características de manipulación de la visualización, de modo que se puedan segmentar los datos de la visualización en función de sus características temporales. Estas características son dos principales:

1. **Selector del rango temporal a mostrar:** A través de un selector de rangos (un elemento de interfaz similar a un *slider* pero con dos cabezas deslizantes), se ha implementado un sistema que permite visualizar los teleports de cualquier tipo en función del rango temporal seleccionado. Este selector se encuentra en la parte superior derecha de la interfaz. De esta forma se puede visualizar, en función del interés del analista, *qué conjuntos de teleports tienen lugar en el rango temporal seleccionado*, así como ir moviendo ese rango temporal en la escala del selector e ir viendo la *evolución temporal en los teleports* y extraer conclusiones sobre estas visualizaciones, como por ejemplo, *qué usuarios son los más activos, qué cantidad de teleports se asocian a un grupo en un tiempo concreto, en qué épocas hay una mayor carga movimientos o uso en el Mundo Virtual, si los teleports tienen características cíclicas en determinados usuarios*, etc. El rango temporal mínimo que se puede seleccionar es de un mes, el máximo dos años.
2. **Slider temporal de los teleports que realiza un usuario en el Mundo Virtual:** A través de la funcionalidad y el *slider* que se describe en la sección 4.7, es posible extraer información sobre todos los movimientos que ha realizado un usuario en el Mundo Virtual, estos movimientos se presentan ordenados temporalmente.

A través de este mecanismo se pueden realizar buena parte de las tareas de análisis que plantea MacEachren como, por ejemplo, *¿un usuario realiza movimientos cíclicos a lo largo del tiempo?*, *¿Con qué frecuencia temporal realiza teleports el usuario?*, *¿Existen cambios significativos en los teleports en función de la época del año?*, *¿Qué cantidad de teleports realiza el usuario en cada época?*, etc.

4.4. Representación de teleports de salida

En este apartado se analizará el comportamiento de la visualización en el caso de la representación visual de los teleports de salida, sea cual sea el filtrado que se aplique sobre la visualización: *teleports de salida de una isla*, *teleports de salida relacionados con un grupo de usuarios*, y *teleports de salida relacionados con cada usuario concreto del Mundo Virtual*.

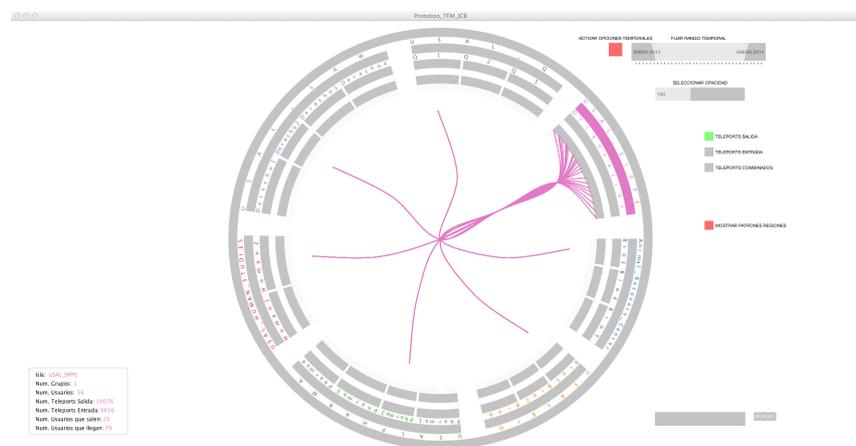


Figura 3. Visualización de los teleports de salida relacionados con una isla o región del Mundo Virtual

En la figura 3 se observa cómo se comporta la visualización en el caso de que se seleccione una región concreta y se quieran visualizar los teleports de salida. En la parte derecha de la interfaz, como se comentó anteriormente se encuentran una serie de elementos de control, y se observa que se ha seleccionado (recuadro en color verde) la visualización de los *Teleports Salida*.

En este caso (la visualización de los teleports de salida de una región) se observa cómo del conjunto de arcos de usuarios relacionados con los distintos grupos y una región concreta, surgen un conjunto de líneas curvas, codificadas con el color correspondiente a esa región (que se puede observar en el color de las etiquetas textuales de la región y los grupos que pertenezcan a ella). Estas líneas representan únicamente los teleports que han realizado los usuarios que pertenecen a grupos de aprendizaje y prácticas relacionados con esa región, no se observan los teleports de salida que han realizado otros usuarios que hayan pasado por la región, debido a que ese comportamiento no cumpliría los principios de la jerarquía de datos y la visualización jerárquica empleada.

En la parte derecha de la visualización, se observa una figura cuadrada roja acompañada de una inscripción que pone *Mostrar Patrones Teleport Región*. Este elemento, si se activa, muestra los patrones o reglas de movimiento que parten de esa región hacia otras, estos patrones de movimiento se han obtenido mediante los procesos de Minería de Datos (apartado 3), y se comentará su visualización en el apartado 4.6.

En el caso de los teleports de salida de un grupo, la visualización toma un aspecto parecido al caso de las regiones, aunque las líneas curvas que representan los distintos teleports surgen de los arcos que representan a los usuarios que pertenecen al grupo que se está analizando, estas líneas también estarán codificadas con el color correspondiente al grupo de usuarios, que coincide con el color asignado a

la región a la que pertenece el grupo. En el caso de este grupo de usuarios, se puede ver que se han movido a todas otras regiones del Mundo Virtual (incluso de su región a su región). Nótese que las líneas de llegada de un teleport a una región, se distinguen porque no llegan a tocar ningún arco perteneciente a una región, ya que el único que podrían tocar sería el de la región (un teleport no puede llegar a un grupo o a un usuario, sólo a un territorio), y si se estirase la línea hasta el arco de la región podría hacer perder parte de la simplicidad a la visualización, dificultando la comprensión de datos.

En el caso del análisis visual de los teleport de salida de un usuario (figura 4), el

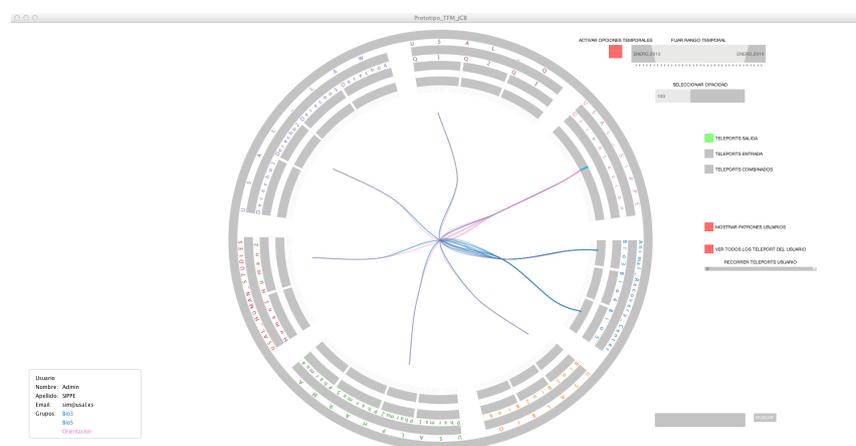


Figura 4. Visualización de los teleport de salida de las regiones a las que pertenece un usuario

comportamiento de la visualización es distinta, ya que muestra todos los teleport de salida de las distintas regiones a las que puede pertenecer un usuario. Es decir, si un usuario pertenece a varias regiones, incluso a varios grupos por región, se mostrarán los teleport que salen de los arcos de ese usuario en cada región, viendo además a qué islas o regiones se dirige. A través de la codificación por colores en función de la región de donde salga el teleport, es posible distinguir a qué territorios se dirige el usuario en función de la isla en la que se encuentre. En este caso se observa que los elementos de control de la parte derecha de la interfaz cambian respecto a los otros casos. Cuando se analizan los teleport de un usuario se ofrecen dos opciones más, la visualización detallada de todos los teleport de un usuario, y la visualización de los patrones de teleport de dicho usuario; las visualizaciones que resultan de activar estas opciones se explicarán a continuación (apartados 4.6 y 4.7).

4.5. Representación de teleports de entrada

En el caso de que un usuario quiera analizar los teleports de entrada a una región, deberá seleccionar dicha opción en los elementos de control de la interfaz, que sólo funcionarán en el caso de que el usuario quiera conocer los teleports de entrada a cualquier región (teniendo activado el visionado de todos los datos), o en el caso de que esté evaluando una región concreta. En los casos del análisis de usuarios y grupos, la opción de visualizar los teleports de entrada no funcionará, debido a que, como se comentó anteriormente, no es posible llegar a un grupo o a un usuario, sólo se llega a un terreno.

Como se puede observar en el figura 5, cuando se selecciona una región y la

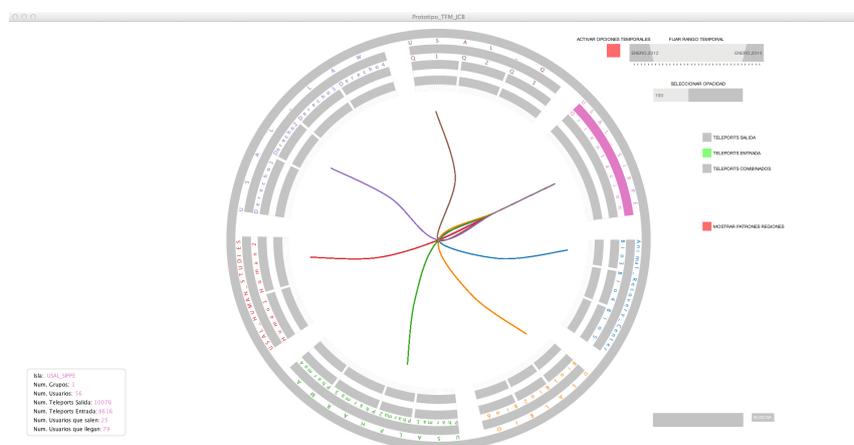


Figura 5. Visualización de los teleports de entrada a una región

opción de ver los teleports de entrada, se muestra una vista donde aparecen distintas líneas codificadas con los colores de las regiones de los que parten los teleports. Por lo tanto, a la región seleccionada llegarán una serie de líneas que parten de las distintas regiones del Mundo Virtual desde donde se haya llegado a la región seleccionada.

4.6. Visualización de los patrones de teleport

Como se apuntaba anteriormente, en las vistas de análisis de usuarios y regiones, se ofrece la opción de visualizar sus patrones de teleport, obtenidos mediante los algoritmos de Minería de Datos, por lo tanto ha sido necesario diseñar cómo un conocimiento extraído de esa forma podría ser representado [16].

En el caso de los patrones de teleport de usuarios, se ha optado por la representación a través de las líneas que representan el origen y destino de esos teleports. Estas líneas son de un color distinto al resto (color `#bcbd22`), y su ancho es

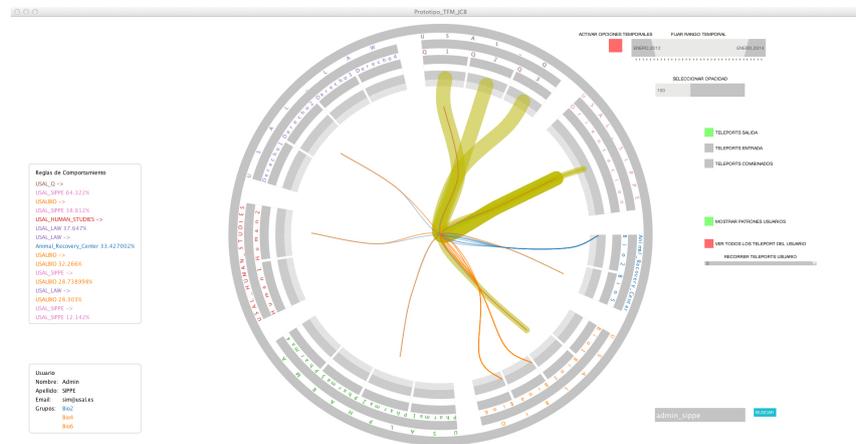


Figura 6. Visualización de los patrones de movimiento en el caso de los usuarios del Mundo Virtual

proporcional a la probabilidad de teleport entre las dos regiones que unen. De esta forma, cuando se representan visualmente los patrones de teleport, se muestran las líneas de teleport del usuario, de modo normal, y las que representan los patrones de movimiento, de modo que el contraste por el color y tamaño hacen que el usuario se fije rápidamente en ellas. Un ejemplo de esta representación visual se puede observar en la figura 6. Hay que señalar que en esta vista, sólo se representan, de igual modo que en los teleport de salida, los teleport de las islas a las que pertenece el usuario, no se pintan los teleport de las regiones a las que no está registrado el usuario. También es necesario apuntar que en la parte izquierda de la interfaz de usuario se mostrará la información textual acerca de las 10 reglas de comportamiento más relevantes para ese usuario, con el texto codificado como en otros casos con los colores de las regiones a las que hagan referencia.

En el caso de la visualización de los patrones de movimientos entre regiones, la representación visual se realiza de forma distinta. En este caso, cuando el analista elige ver los patrones de movimientos, desaparece de la visualización el conjunto de teleports que salen o entran a esa región seleccionada. En su lugar se dibuja un esquema de puntos y líneas curvas, de modo que cada punto, centrado en el arco de la región a la que se corresponde, se dibuja con una codificación en función del color de la región a la que representa. Una vez establecida la referencia entre los puntos correspondientes a las regiones del entorno virtual que estén presentes en las reglas o patrones de teleport, se dibujan las líneas que unen la región de origen con las de destino. En este caso las líneas se codificarán con el color de la región de origen, y representarán la probabilidad de ocurrencia de ese patrón de teleport a través de su ancho (figura 7). De igual forma que en los patrones de teleport referentes a los usuarios, en la parte izquierda de la pan-

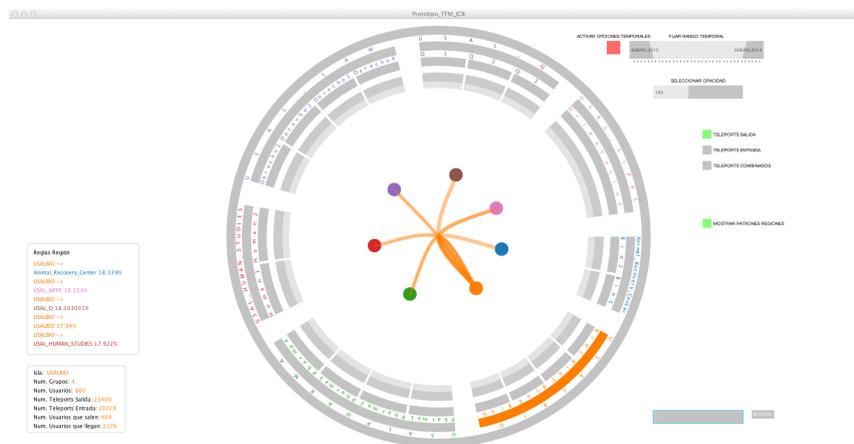


Figura 7. Visualización de los patrones de movimiento entre regiones del Mundo Virtual

talla se mostrará la información textual sobre esas reglas, también codificando las regiones implicadas en cada regla con los colores de la región a la que hacen referencia.

4.7. Visualización detallada de los teleports de un usuario

En la vista de análisis sobre los teleports de los usuarios, se ofrece una opción de visualización detallada de todos los teleports que ha realizado un usuario a lo largo de la existencia de su *avatar* en el entorno 3D. Cuando se activa la opción *Ver todos los teleports del usuario*, disponible únicamente en la vista de teleports de un usuario concreto, se activa un *slider* en la interfaz, a través del cual, mediante el deslizamiento por la barra horizontal y su escala, es posible visualizar paso a paso y en orden cronológico todos los teleports que ha realizado un usuario. De este modo es posible ver la evolución *espacio-temporal* del usuario, pudiendo comprender en función de las fechas y el teleports que realiza, qué le mueve a realizar cada uno de ellos (para llegar a este nivel de descubrimiento de conocimiento, el usuario que analice estos conjuntos de teleports debe tener un amplio conocimiento previo de las actividades teórico-prácticas que existen y qué características tiene el entorno virtual). Como complemento a la visualización de líneas que van de una región a otra, se hace uso de información textual, aunque en este caso en la parte derecha de la interfaz, justo debajo del *slider* usado para hacer las transiciones sobre el conjunto de teleports. En esta información textual se incluye una descripción acerca del teleports que se está mostrando; con una descripción que incluye la región de origen, la región de destino, el usuario que la realiza, y la fecha y hora del teleports para que se comprendan perfectamente las características del teleports, y se pueda conocer si tiene relevancia o no en las tareas de análisis.

4.8. Codificación de colores

Como señalan múltiples autores, como Colin Ware [17] o Edwar Tufte [18], una correcta aplicación del color en la visualización puede hacer que una visualización de información sea más clara, más agradable para el usuario, y sobretodo más útil. Del mismo modo, Healey [19] sostenía que no se deben mostrar más de diez colores simultáneamente en la pantalla, ya que los usuarios dejan de diferenciarlos de forma clara y se pierde información. Por ello, se ha planteado una interfaz donde nunca se lleguen a visualizar más de diez colores, esto es posible mediante una cuidada organización en la distribución de elementos visuales y su asignación de colores.

En este trabajo de investigación, el color ha tenido una gran influencia, ya que



Figura 8. Vista detalle sobre la codificación del color en la información textual de cada tipo de elemento de la jerarquía de datos

se ha hecho uso de la capacidad de *Etiquetado* que posee (Ware [17]). Esta capacidad implica que un color es un buen recurso para asociar objetos visuales, y hace que el usuario pueda relacionar distintos atributos de objetos a simple vista y sin tener que realizar un complejo proceso mental de *emparejado* de entidades visuales. En cuanto a esta característica de etiquetado, cabe destacar que se ha aplicado de forma que un mismo color representa siempre la misma semántica en los datos, es decir, cada isla o región del entorno virtual, así como los grupos y usuarios que pertenecen a ella, tienen un único color de etiquetado, de modo que el usuario sea capaz de, sólo visualizando el color, relacionar el concepto con la isla, grupo o usuario del que se está mostrando la información (en la figura 8 se observa como incluso la información textual de cada tipo de datos se codifica por los colores correspondientes). Cada teleport que salga de la jerarquía de datos de una región, tendrá asociado un color, de modo que, por ejemplo, es posible distinguir la dirección de un teleport en función del color de la línea que representa el teleport.

En cuanto a la paleta o gama de colores disponibles en la visualización (#1f77b4, #ff7f0e, #2ca02c, #d62728, #9467bd, #8c564b, #e377c2, #c2c2c2, #cbcd22,

#17becf), es necesario comentar que no es una escala determinada simplemente por características de diseño, sino que se trata de una escala desarrollada de tal forma que la diferencia entre las distintas tonalidades y colores que la conforman son lo suficientemente distintas como para que un usuario (sin problemas de visión del color) sea capaz de diferenciarlas sin problema. Esta escala ha sido propuesta por Mike Bostock y se conoce como *Categorical Colors* [20].

La aplicación de estos colores, a nivel de diseño concuerda con el objetivo de desarrollar una interfaz clara, sencilla y visualmente atractiva. Por ello se ha optado por el uso de colores claros para los datos que se muestren siempre, y colores vivos para centrar la atención sobre los datos que deben ser considerados de inmediato por el usuario o analista.

4.9. Mecanismos de Interacción en la Visualización *HEBATT*

A lo largo de este trabajo de investigación, se ha tenido especial cuidado con todos los aspectos que influyen en la calidad de la visualización, desde la codificación de variables visuales, la muestra de información [21], las herramientas de análisis que se permiten, etc. Dentro de este cuidado de los elementos que aseguran la calidad de la visualización, se encuentran unos muy especiales: *los Mecanismos de Interacción*. Estos mecanismos son métodos de interacción con la visualización, de modo que permiten realizar de una forma más sencilla las distintas tareas de análisis [22].

Entre los distintos Métodos de Interacción que se incluyen en la herramienta visual *HEBATT*, destacan:

- **Seleccionar:** Permitir seleccionar un elemento visual que se quiere analizar, hace que el usuario pueda generar visualizaciones de datos segmentadas que le facilitan la comprensión de la información que se le presenta. El método de interacción *seleccionar* es el que permite que cuando un usuario pinche en un arco de región, grupo o usuario, la visualización cambie para mostrar sólo los datos relacionado con el elemento sobre el que ha hecho clic.
- **Explorar:** El mecanismo de interacción *explorar*, permite al usuario modificar la visualización de modo que pueda cambiar los datos que desee sin mostrar el resto. Este mecanismo se implementa a través de los distintos elementos *deslizadores* o *sliders*, ya que en varios casos en los que se pueden aplicar (movimiento en la línea temporal de los teleport, y muestra sucesiva de los teleports de un usuario), mediante el movimiento del cabezal de este elemento se va modificando la vista, de modo que el usuario puede ir buscando la ocurrencia de datos que desee, sin que le molesten el resto de datos que no tengan relación con la selección que hace el usuario.
- **Filtrar:** Los filtros permiten al usuario seleccionar qué datos desea observar y cuáles no, ya que todos los datos que no cumplan el filtro seleccionado por el usuario no se mostrarán. En la visualización *HEBATT* hay distintos métodos de filtrado:
 - **Filtrados temporales:** Del mismo modo que se puede hacer un uso exploratorio del selector temporal, este sirve también para filtrar los datos de

movimientos en el espacio virtual en función de las fechas. Si se ajusta un rango de fechas, y se fija en algún punto de la línea temporal, se está realizando un filtrado por fecha, de modo que en la visualización sólo aparecerán los datos que se correspondan con ese rango de fechas definido.

– **Filtrados por tipo de teleport:** En el caso de la visualización de datos general y por regiones, es posible especificar qué tipo de teleport se quiere visualizar (entrada, salida o combinados). Esto se indica mediante un filtrado implementado a través de un elemento de tipo *radioButton*, que sólo permite que esté seleccionado un filtro por tipo de teleport en un instante concreto de tiempo.

– **Filtrados para patrones de comportamiento:** De forma similar al *filtrado por tipo de teleport*, en las vistas que en las que se analizan los datos de las regiones o lo usuarios, es posible seleccionar la opción *Mostrar Patrones (de usuario o región)*. En este caso este filtrado que permite mostrar o no los patrones de comportamiento es distinto al resto de filtrados, ya que en función del elemento donde se seleccione el filtrado, la vista de datos anterior desaparecerá o no. En el caso de los usuarios, los datos referentes al usuario seleccionado no desaparecen, sino que la visualización de los patrones de comportamiento *completa* la vista de los datos. Sin embargo, en el caso de la muestra de patrones de movimientos en las regiones, los teleport de la región se ocultan para mostrar únicamente la codificación visual de estos patrones de teleport entre regiones.

– **Filtrados mediante búsquedas:** El analista que utilice la herramienta de visualización, puede realizar filtrados mediante la introducción de texto en la caja de entrada textual habilitada para tal efecto. Mediante la correcta escritura del nombre de alguna región, grupo o usuario, el sistema filtrará la información que se está mostrando para pasar a representar visualmente sólo la información correspondiente al elemento que haya coincidido con la búsqueda.

- **Modificar la interfaz:** En este caso, al usuario se le permiten hacer modificaciones en los elementos visuales, cambiando así su aspecto para revelar si es posible datos que se encuentren ocultos o no sean claros. Esta modificación de la interfaz se realiza a través del *slider* “*Seleccionar Opacidad*” que permite al usuario seleccionar la opacidad o transparencia de las líneas que representan los conjuntos de teleports y así descubrir si hay relaciones en los datos que no se vean de un primer vistazo .

5. Conclusiones

En este trabajo de investigación se ha desarrollado un nuevo tipo de representación visual de los movimientos de los usuarios entre regiones de un Mundo Virtual de Aprendizaje y Prácticas, con lo cual se ha cumplido el objetivo que se planteaba inicialmente. En cualquier caso, no se puede asegurar que esta visualización sea totalmente válida, ya que no ha sido probada por usuarios reales, ni analistas de datos. Ese es un aspecto de los que se realizaría en futuras iteraciones en el proceso de investigación, ya que es fundamental validar la herramienta

mediante el uso, de modo que se pueda comprobar si los conceptos teóricos aplicados son lo suficientemente claros como para que un usuario del tipo que sea (experto, no iniciado, etc.) sea capaz de usar la visualización sin ayuda y con un tiempo de entrenamiento lo más corto posible.

Esta visualización se ha desarrollado para el uso concreto en el caso de los teleports en un Mundo Virtual, pero de igual forma se podría aplicar a cualquier tipo de movimiento migratorio, ya sea generado en un sistema digital o en la realidad. Por ejemplo, un esquema similar al que se presenta en este trabajo, se podría proponer para visualizar los flujos migratorios (de personas, de animales, etc.) entre regiones y países del mundo, ya que mientras se cumpla en parte la jerarquía de datos la visualización se puede adaptar a otros casos.

Bibliografía

- [1] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson, "Past, present, and future of decision support technology," *Decision support systems*, vol. 33, no. 2, pp. 111–126, 2002.
- [2] K. Börner, W. R. Hazlewood, and S.-M. Lin, "Visualizing the spatial and temporal distribution of user interaction data collected in three-dimensional virtual worlds," in *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pp. 25–31, IEEE, 2002.
- [3] K. Börner and S. Penumathy, "Social diffusion patterns in three-dimensional virtual worlds," *Information Visualization*, vol. 2, no. 3, pp. 182–198, 2003.
- [4] J. Cruz, R. Therón, E. Pizarro, and F. J. García-Peñalvo, "Análisis de datos en mundos virtuales educativos," in *Actas del XV Simposio Internacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE13)*, 2013. En Prensa.
- [5] J. Cruz, R. Therón, E. Pizarro, and F. J. García-Peñalvo, "Knowledge discovery in virtual worlds usage data: approaching web mining concepts to 3d virtual environments.," in *Proceedings Fourth International Workshop on Knowledge Discovery, Knowledge Management and Decision Support (Eureka-2013)*, 2013. En Prensa.
- [6] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, pp. 207–216, ACM, 1993.
- [7] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.
- [8] E. Pizarro Lucas and J. Cruz Benito, "Usalsim: Inserción profesional, prácticas y aprendizaje en un mundo virtual," in *Buenas Prácticas de Empleabilidad de Estudiantes Universitarios*, pp. 87–99, Universidad Miguel Hernández, Ministerio de Educación, Cultura y Deporte, 2012.
- [9] E. P. Lucas, J. C. Benito, and O. G. Gonzalo, "Usalsim: Learning and professional practicing in a 3d virtual world," in *2nd International Workshop on Evidence-based Technology Enhanced Learning*, pp. 75–82, Springer, 2013.
- [10] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 741–748, 2006.
- [11] D. H. Holten, *Visualization of graphs and trees for software analysis*. PhD thesis, Technische Universiteit Eindhoven, 2009.
- [12] B. Cornelissen, D. Holten, A. Zaidman, L. Moonen, J. J. Van Wijk, and A. Van Deursen, "Understanding execution traces using massive sequence and circular bundle views," in *Program Comprehension, 2007. ICPC'07. 15th IEEE International Conference on*, pp. 49–58, IEEE, 2007.

- [13] B. Shneiderman and C. Plaisant, *Designing The User Interface: Strategies for Effective Human-Computer Interaction*, 4/e (New Edition). Pearson Education India, 2003.
- [14] “Processing: Drawing curves.” <http://processing.org/tutorials/curves/>. R. E. (14). Julio, 2013.
- [15] A. MacEachren, *How Maps Work*. Guilford, New York, 1995.
- [16] D. A. Keim, “Information visualization and visual data mining,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 1, pp. 1–8, 2002.
- [17] C. Ware, *Information Visualization: Perception for Design*. Elsevier Inc, 2013. ISBN: 978-0-12-381464-7.
- [18] E. R. Tufte, *Beautiful evidence*, vol. 23. Graphics Press Cheshire, CT, 2006.
- [19] C. G. Healey, “Choosing effective colours for data visualization,” in *Visualization’96. Proceedings.*, pp. 263–270, IEEE, 1996.
- [20] M. Bostock, “D3 categorical colors.” <https://github.com/mbostock/d3/wiki/Ordinal-Scales#categorical-colors>. R. E. (14). Julio, 2013.
- [21] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336–343, IEEE, 1996.
- [22] R. Therón and L. Fontanillo, “Diachronic-information visualization in historical dictionaries,” *Information Visualization*, 2013. En Prensa.

Análisis de integración de soluciones basadas en *software* como servicio para la implantación de ecosistemas tecnológicos corporativos

Alicia García Holgado¹ y Francisco José García Peñalvo¹

¹Grupo de investigación GRIAL - Instituto Universitario de Ciencias de la Educación (IUCE) - Departamento de Informática y Automática, Universidad de Salamanca.
Paseo de Canalejas 169. 37008 Salamanca, España
{aliciagh, fgarcia}@usal.es

Resumen El *software* como servicio y la orientación 2.0, con independencia del dominio, constituyen la realidad tecnológica de las instituciones. Existen aplicaciones *Open Source* potentes que cubren la mayoría de las necesidades TIC de cualquier entidad, tanto para su operatividad como para la gestión de sus procesos internos. Por tanto, el problema no reside en la falta de soluciones tecnológicas sino en lograr que la integración no sea *ad-hoc*. En este punto entra en juego el concepto de *mashup*, que considera un sistema de información web como una combinación de componentes de varias fuentes para conformar un servicio completo. Este trabajo busca cómo obtener *mashups* con una alta flexibilidad y adaptabilidad a las necesidades tanto del usuario final como del técnico o especialista. Este tipo de solución constituye un ecosistema tecnológico, donde cada uno de los componentes del sistema es un organismo con un ciclo de vida independiente, y el medio físico donde se relacionan abarcará la integración y los flujos de información entre dichos componentes. Para ello se estudian soluciones en diferentes ámbitos como empresas, la Administración Pública o instituciones educativas. Con los principales resultados y conclusiones de estos estudios se define un patrón arquitectónico para un ecosistema tecnológico de aprendizaje que se aplica en un nuevo dominio.

Keywords: Ecosistema Tecnológico, Ecosistema *eLearning*, *Software Open Source*, Servicios Web, Patrón Arquitectónico

1. Introducción

La constante evolución de la tecnología en el ámbito de la computación, más concretamente de las redes de información y los dispositivos que acceden a dichas redes, ha supuesto una evolución en el desarrollo de soluciones *software*. La aparición en 2004 de la Web 2.0 [1] supuso un cambio radical en la forma de percibir el uso de Internet. Las páginas web con contenido estático creadas por unos pocos usuarios con los conocimientos de HTML necesarios dejaban paso a una web dinámica en la que el espectador se convierte en creador, ya que puede aportar sus propios contenidos de manera sencilla.

Hace tan solo una década el desarrollo *software* se centraba en la definición de aplicaciones de escritorio que fueran compatibles con el sistema operativo de los usuarios finales. Actualmente tanto los usuarios como las empresas de *software* han ido abandonando este tipo de aplicaciones estancas para pasar a utilizar y desarrollar aplicaciones *software* basadas en la Web 2.0. Unido a esto, la alta penetración de los dispositivos móviles en la sociedad ha generado una gran demanda de aplicaciones accesibles desde cualquier tipo de dispositivo, lo que acentúa más la tendencia de las organizaciones de moverse hacia un modelo de *cloud computing* [2] [3] para cubrir sus necesidades, tendiendo al uso de soluciones *Software as a Service* (SaaS), un modelo de implementación *software* donde las aplicaciones están alojadas remotamente por el proveedor de la aplicación o servicio y puestas a disposición del cliente bajo demanda a través de Internet [4].

El cambio de paradigma, junto a la aparición de una gran variedad de *frameworks* para el desarrollo de este tipo de soluciones, ha provocado una explosión en el desarrollo de aplicaciones *software* basadas en la Web, así como de aplicaciones para dispositivos móviles. Existe una sobrecarga de soluciones *software* con un mismo objetivo, de tal manera que, en muchas ocasiones, el problema no reside tanto en el desarrollo de soluciones *software* sino en la elección de la mejor solución de las que ya existen.

Paralelamente a la vertiginosa evolución de la tecnología, el concepto de *Software Libre* [5] y desarrollos *Open Source* [6] ha tomado fuerza y cada vez son más las empresas e instituciones que apuestan por este tipo de desarrollos. Un ejemplo de ello es la estrategia de la Comisión Europea de promoción de *software Open Source* [7] [8] o la apuesta por el conocimiento y el uso del *software* libre del Gobierno de España mediante el Centro nacional de Referencia de Aplicación de las TIC basadas en fuentes abiertas (CENATIC) [9].

Las entidades, desde la pequeña empresa hasta la Administración Pública, generan una gran cantidad de conocimiento. La importancia de darle valor a dicho conocimiento ha ido en incremento a lo largo de los últimos años, unido a la creciente capacidad tecnológica para almacenar y difundir la información. Hoy en día, son pocas las empresas e instituciones que no tienen entre sus objetivos estratégicos realizar una correcta gestión del conocimiento que generan, de tal forma que este revierta en beneficios para la propia entidad. A grandes rasgos, los principales beneficios que el conocimiento generado puede reportar a una entidad, independientemente del tipo de entidad que sea, son dos. En primer lugar, una buena gestión del conocimiento y, por tanto, de la información, permite mejorar los flujos de trabajo y proporciona las vías necesarias para la toma de decisiones. En entidades de pequeño tamaño esta ventaja puede ser menos evidente pero si se toma una institución de gran envergadura o un grupo de empresas, la toma de decisiones es un proceso complejo cuyo resultado conlleva graves consecuencias si no se tienen en cuenta todas las variables. En segundo lugar, una buena gestión del conocimiento incluye procesos y métodos para darle visibilidad, si no a todo al menos a una parte, con el fin de llegar al público de interés. Una empresa o una institución que no tiene presencia en Internet, que

no muestra los logros conseguidos, las mejoras en las que invierte sus esfuerzos, tiene fecha de caducidad.

Las empresas e instituciones están realizando una fuerte apuesta por la orientación 2.0 y el *software Open Source* como pilares fundamentales para cubrir sus necesidades tecnológicas, tanto para su funcionamiento interno como para su visibilidad de cara al público.

Existe un gran abanico de soluciones *Open Source* que cubren las necesidades TIC básicas de una entidad, desde aplicaciones para la gestión de contenidos o de la formación hasta herramientas para la toma de decisiones o la planificación temporal. Por tanto, el problema no reside en el desarrollo de dichas soluciones, sino en la elección e integración de las mismas para que los flujos de información se adapten a las peculiaridades propias de cada entidad. Este tipo de soluciones, denominadas ecosistemas tecnológicos y cuya definición se plantea a continuación, permiten centrarse en la información, en el conocimiento, para dejar de lado la tecnología subyacente.

El término ecosistema tecnológico se utiliza para definir un conjunto de componentes *software* que se relacionan entre sí mediante flujos de información en un medio físico que proporciona el soporte para dichos flujos. Si se toma la definición más sencilla de ecosistema natural se habla sobre un conjunto de organismos vivos y un medio físico donde se relación. Si se observa ambas definiciones se ve una clara analogía entre un ecosistema natural y un ecosistema tecnológico, los componentes *software* desempeñan la labor de organismos vivos del ecosistema tecnológico, interactuando con otros organismos y viéndose condicionados por el medio físico que les rodea. Al igual que ocurre en un ecosistema natural, el ecosistema tecnológico cumple los tres principios básicos de la ética ecológica:

- En un ecosistema todos los seres vivos son interdependientes y se necesitan entre sí. Lo mismo ocurre en un ecosistema tecnológico, donde cada componente se relaciona con otros componentes. Si un componente es totalmente independiente entonces no forma parte del ecosistema.
- La estabilidad de los ecosistemas depende de su diversidad. Un ecosistema tecnológico es más estable cuantas más posibilidades y opciones ofrezca, pero manteniendo una armonía, unidad, seguridad y coherencia entre sí.
- Todas las materias primas son limitadas y existen límites en el crecimiento de todos los sistemas vivos. El ecosistema tecnológico debe crecer de manera controlada. Su evolución sin un fin concreto puede llevar a tener un ecosistema insostenible que no cumple con la finalidad para la que se desarrolló.

Cabe destacar que una de las características fundamentales de un ecosistema tecnológico es la capacidad del sistema para soportar los cambios y adaptarse a las necesidades de los usuarios a lo largo del tiempo. Para que esto ocurra los componentes *software* que forman el ecosistema deben ser capaces de adaptarse a los cambios. Según Darwin *“It is not the strongest of species that survive, nor the most intelligent, but the ones most adaptable to change.”* [10]. Traducido al contexto tecnológico, no pervive en el ecosistema aquel componente que es más robusto, o que aporta la funcionalidad clave, sino el que mejor se adapta a los cambios del conjunto de componentes con los que coexiste.

El ecosistema tecnológico se puede orientar a diferentes ámbitos. Si el ecosistema está orientado a la gestión de la formación se denomina ecosistema de aprendizaje o ecosistema *eLearning* [11] [12]. También puede orientarse a la creación de un espacio integral de gestión de la información interna/externa de la entidad o a la gestión de procesos internos, la implementación de herramientas y procesos de social media, o una combinación de estas soluciones.

El GRupo de Investigación en InterAcción y *eLearning* (GRIAL) de la Universidad de Salamanca (<http://grial.usal.es>), desde su nacimiento en 2006, ha trabajado en la mejora y evolución de los ecosistemas tecnológicos, de forma que la gestión estratégica del conocimiento y de la tecnología es una de sus principales líneas de investigación. Gracias al carácter interdisciplinar de sus miembros, procedentes de tres ámbitos de investigación bien diferenciados pero que se complementan (la Ingeniería Informática, la Educación y la Teoría de la Comunicación), el planteamiento de los ecosistemas tecnológicos ha podido llevarse a cabo en contextos muy variados, lo que ha permitido aprender de las experiencias e incorporar los resultados de investigación generados en las otras líneas de investigación que desarrolla el grupo, todas ellas estrechamente vinculadas entre sí de manera transversal (sistemas interactivos para el aprendizaje, las tecnologías para el aprendizaje, la ingeniería web y la arquitectura del *software*, la metodología del aprendizaje *online*, la calidad y evaluación en educación, la teoría de la comunicación) [13].

En las siguientes secciones se presenta una descripción del estado del arte sobre sistemas de información hasta llegar a la percepción actual de los ecosistemas tecnológicos y en particular de los ecosistemas para el aprendizaje (2), los diferentes casos de estudio analizados (3), la propuesta arquitectónica obtenida a partir del análisis de los casos previos (4) y por último las conclusiones obtenidas junto con las líneas de investigación futuras que se plantean como resultado del proceso de investigación llevado a cabo (5).

2. Estado del arte

Toda empresa necesita disponer de sistemas de información para la gestión de la propia empresa. A lo largo del tiempo, con la evolución del *software* y de las necesidades de las empresas, los sistemas de información destinados a estas han evolucionado considerablemente, pasando desde el *software* para la gestión contable y el *software* de control de stocks al MRP (*Material Requirement Planning*), pasando por el MRP II (*Manufacturing Resources Planning*), hasta llegar finalmente a los sistemas de información para la gestión de empresa actuales, los ERP (*Enterprise Resources Planning*) [14].

Un ERP según Esteves & Pastor [15] es “un sistema compuesto por varios módulos, tales como, recursos humanos, ventas, finanzas y producción, que posibilitan la integración de datos a través de procesos de negocios incrustados. Estos paquetes de *software* pueden ser configurados para responder a las necesidades específicas de cada organización”. El ERP proporciona la infraestructura de gestión de la información en los ecosistemas de negocio, los cuales se definen

como la red de compradores, proveedores y fabricantes de productos o servicios y su entorno de negocio [12]. Se trata, por tanto, de la parte tecnológica, digital, del ecosistema de negocio y por analogía, el ERP podría ser el ecosistema tecnológico o ecosistema digital de la empresa o institución. El ecosistema tecnológico o digital de la empresa sería un sistema compuesto por un conjunto de módulos que proporcionan la funcionalidad necesaria para gestionar los procesos de negocio internos y externos de la empresa, para permitir el flujo de información entre los diferentes componentes del sistema. Uden [12] extiende esta definición con conceptos tales como evolución y cooperación, de tal forma que un ecosistema digital es un entorno digital que permite la cooperación, el intercambio de conocimiento, el desarrollo de tecnologías *open* y *adaptive*, junto con la evolución del modelo de negocio, mediante la recombinación y evolución de los componentes digitales que forman el ecosistema. Se consideran como componentes digitales elementos tales como las aplicaciones, los servicios, el conocimiento, los procesos y los modelos de negocio y los módulos de aprendizaje.

Desde la Unión Europea se considera a los ecosistemas digitales como la evolución de las herramientas de *e-business* y los entornos de colaboración para redes de organización. Un ecosistema digital posee una arquitectura basada en componentes de *Open Source* que se combinan para trabajar de manera conjunta para permitir la evolución gradual del sistema mediante la aportación de ideas y nuevos componentes por parte de la comunidad [16].

Según Chang [17] el ecosistema natural se define como un entorno abierto, débilmente acoplado, agrupado por dominios, conducido por las necesidades, con agentes auto-organizados, donde cada especie es proactiva y responsable de su propio beneficio. Si se sigue esta definición, cada una de las características del ecosistema natural mencionadas se corresponden con características del ecosistema digital. Abierto hace referencia a la transparencia del entorno. Débilmente acoplado se define como las relaciones abiertas que se establecen entre especies o entidades en la comunidad virtual, es decir, las entidades se relacionan entre sí, pero no existe una fuerte dependencia entre las mismas. El dominio, que se define como un conjunto de especies que tienen en común el mismo tipo de vida o intereses parecidos, está presente dentro de un ecosistema digital donde los componentes se agrupan según los objetivos de gestión de información, claramente definidos por la entidad en la que se implementa el ecosistema. Conducido por las necesidades significa que las especies dentro del ecosistema digital colaboran para alcanzar un beneficio mutuo. Los agentes auto-organizado son las personas, las herramientas y las tecnologías de la información que dan soporte a los flujos de información entre los diferentes recursos, de tal forma que pueden tomar decisiones y cumplir responsabilidades para que dichos flujos tenga lugar de forma correcta. Los agentes participan activamente en la comunidad y son responsables de sus propias acciones.

Chang propone el ecosistema digital como un conjunto de agentes *software* inteligentes que se organizan y colaboran entre sí para gestionar la información según las necesidades existentes en cada momento dentro del ecosistema. Este planteamiento sustituye a los usuarios como elementos centrales del ecosistema,

de forma que este rol se le atribuye a los agentes inteligentes que, a pesar de poder llevar a cabo diferentes tareas y poseer características que les distinguen unos de otros, tienen un comportamiento predefinido y aunque puedan aprender a lo largo del tiempo y evolucionar no dejan de ser un componente *software* más. La interacción entre usuarios es un elemento fundamental en el ecosistema, deben existir tareas automatizadas, flujos de información que se adapten a las necesidades cambiantes del usuario, pero la interacción del usuario y la detección de necesidades, permiten que el ecosistema evolucione y se adapte más rápidamente a su contexto. Los agentes inteligentes pueden adaptarse, pero les falta la visión de una parte del sistema, la evolución de la propia entidad.

Basándose en la definición de Chang, Pata [18] elimina el concepto de agente inteligente como especies del ecosistema y plantea una definición mucho más biológica dentro del contexto tecnológico: “Es un entorno abierto, auto-organizado, que conecta ubicaciones geográficas y basadas en web, individuos, servicios de información basados en herramientas sociales, interacción en la red y herramientas de intercambio del conocimiento junto con recursos que ayudan a mantener las sinergias entre las personas, donde cada sujeto es proactivo y responsable de su propio beneficio.”

Aunque existen diferencias entre unos autores y otros, todos están de acuerdo en un punto fundamental, hay una clara relación entre las características de un ecosistema natural y un ecosistema tecnológico en cualquiera de sus variantes. De esta forma, la mayoría de los autores respaldan la definición de ecosistema tecnológico o de ecosistema de aprendizaje con la definición de ecosistema natural con diferentes matices.

Cabe destacar la manera en la que Laanpere [19] aborda el tema. Este autor se centra en los conceptos y deja de lado la funcionalidad o la naturaleza de los elementos que representan estos conceptos. Laanpere representa los ecosistemas naturales y los ecosistemas digitales mediante conceptos relacionados entre sí, de tal manera que el ecosistema digital es una extensión del ecosistema natural. Todos los conceptos que se utilizan para describir un ecosistema natural permiten describir un ecosistema digital. De esta forma, según lo expresado en el propio diagrama, un ecosistema *eLearning* digital estará formado por elementos bióticos, elementos abióticos, el entorno y los agentes.

Si se observan estas y otras definiciones, hay tres elementos que forman parte de todo ecosistema natural y que están presentes en todas las definiciones [20]:

- Organismos, cada uno de los cuales intenta cubrir sus funciones básicas: comer, reproducirse y respirar.
- El medio físico en el cual los organismos llevan a cabo sus funciones básicas.
- El conjunto de relaciones entre los organismos, entre sí y con el medio.

Lo mismo ocurre en un ecosistema tecnológico, aunque con diferentes matices, en todas las definiciones se tienen una serie de componentes *software* que se relacionan entre sí en un medio físico que proporciona el soporte para dichas relaciones.

El ecosistema tecnológico se puede orientar a diferentes ámbitos. Si el ecosistema está orientado a la gestión de la formación se denomina ecosistema de aprendizaje o ecosistema *eLearning*.

Pirie [21] plantea en el 2004 el ecosistema *eLearning* como la unión entre la comunidad de aprendizaje y la empresa mediante un *Learning Management System* (LMS). Considera inherente al desarrollo de este tipo de ecosistema la integración con los recursos humanos y el portal web de la empresa, lo que permite a los empleados, desde el propio ecosistema, llevar a cabo actividades relativas al contexto laboral tales como gestión de los planes médicos o consultar los programas de formación que cubran las competencias relacionadas con el puesto de trabajo y la carrera profesional del empleado. Pirie introduce un concepto muy interesante, el uso del ecosistema *eLearning* para la toma de decisiones gracias a la información proporcionada por el ecosistema para definir planes de formación o mejorar la gestión de las actividades de toda la empresa.

Brodo [22] define ecosistema como un conjunto de interacciones sistémicas entre grupos de organizaciones en las industrias que se superponen y entrecruzan para crear un equilibrio entre los flujos de oferta y demanda. Si se toma dicha definición como punto de partida, se indica que el término ecosistema de aprendizaje se utiliza para describir todos los elementos necesarios para implementar una solución *eLearning*, de forma que dichos elementos pertenecen a tres categorías: proveedores de contenidos, consultores e infraestructura. A pesar de utilizar el concepto de ecosistema, la solución no se diferencia de las soluciones *eLearning* tradicionales donde existen unos contenidos, una metodología y un *software* encargado en proporcionar el entorno de gestión del aprendizaje, el LMS. La interacción entre los elementos así como los flujos de información no son parte del ecosistema.

Más adelante, Uden [12], que parte de la base de las definiciones de Pirie y Brodo, propone una aproximación a la definición de un ecosistema *eLearning* en cinco grandes bloques, lo que refuerza dos aspectos fundamentales de cualquier ecosistema, independientemente de la orientación del mismo, la integración y una base teórica o metodológica:

- La identificación de las necesidades de la organización que permiten la definición de una estrategia *eLearning*.
- La elección del LMS correcto que se pueda integrar con las herramientas ya existentes y que evite el uso de sistemas propietarios.
- La definición del diseño pedagógico que sustenta la formación.
- Los contenidos que van a estar disponibles dentro del ecosistema, además se debe tener en cuenta el soporte a los estándares definidos en cuanto a desarrollos de objetos de aprendizaje con el objetivo de incrementar la reusabilidad de los contenidos.
- La facilidad de adaptación frente a cambios.

Ficheman [23] propone un esquema bastante simple de un ecosistema de aprendizaje, con usuarios y contenidos como factores bióticos y *hardware*, *software* y tecnología de red como elementos abióticos del ecosistema. En un ecosistema natural los individuos de una misma especie se agrupan en poblaciones,

y varias poblaciones se agrupan en una comunidad. Ficheman propone que las poblaciones en un ecosistema de aprendizaje se forman a partir de los roles de los usuarios, docente, tutor, técnico, etc., y las comunidades son las organizaciones que agrupan a los diferentes roles y tipos de contenido.

Según Littlejohn [24], la combinación de las herramientas sociales con los ecosistemas de aprendizaje permite plantear ecosistemas de aprendizaje colectivo. El ecosistema *eLearning* se centra en los recursos y las interacciones como medio para llegar a ellos y crear nuevo conocimiento. Los recursos funcionan como nexo de unión entre las personas que participan en el ecosistema, pero el centro del aprendizaje es la interacción que se genera alrededor de estos recursos, de tal manera que gracias a esa interacción se puede generar nuevo conocimiento, que permite que el ecosistema de aprendizaje se mantenga y evolucione.

Ark [25] plantea el ecosistema *eLearning* como una evolución del LMS cuyo fin es cubrir aspectos que influyen en el paradigma de aprendizaje, tales como contenidos adaptativos, aprendizaje social, perfiles globales, motores inteligentes, gran cantidad de contenidos en abierto y la explosión de aplicaciones móviles.

Independientemente del enfoque que se le de al ecosistema, ya sea de aprendizaje centrado en recursos o interacción entre agentes inteligentes, existe un conjunto de componentes tecnológicos que forman el ecosistema. Adomavicius [26] propone una serie de sencillos pasos para identificar la tecnología utilizada en el ecosistema tecnológico:

- Identificar la tecnología central o el punto de inicio del ecosistema tecnológico y definir el contexto para poder identificar el resto de tecnologías necesarias.
- Identificar las tecnologías que complementan a la central.
- Identificar los componentes tecnológicos finales.
- Identificar las tecnologías para dar soporte y crear la infraestructura necesaria para las anteriores tecnologías.

Según plantea Haymes [27], los ecosistemas tecnológicos influyen en la aceptación de una nueva tecnología. Cada vez se tiende más a la integración de diferentes herramientas para cubrir las necesidades de los usuarios. Si una tecnología emergente no es capaz de integrarse con tecnologías cuyo uso está extendido y aceptado por un gran número de usuarios y cuya funcionalidad puede complementar a la de la nueva tecnología, entonces dicha tecnología terminará por caer en desuso. Haymes añade un patrón para evaluar nuevas tecnologías, de tal forma que las tecnologías utilizadas en el ecosistema deberían cumplir dicho patrón y el conjunto global, el ecosistema, también:

- Debe hacer algo que sea evidente para los usuarios.
- Debe ser fácil de utilizar.
- Debe convertirse en algo esencial en la vida del usuario.

Finalmente, Uden [12] organiza la tecnología elegida en tres capas: infraestructura que sería una base tecnológica genérica, una capa que implementa la adecuación del ecosistema al entorno de negocio y, por último, la implementación de un modelo de negocio específico.

Aquí entra en juego la escalabilidad del sistema. El ecosistema tecnológico, al igual que el ecosistema natural, debe estar en continua evolución, adaptándose a los cambios del entorno y permitiendo que los organismos que forman parte del mismo puedan adaptarse a dichos cambios. El ecosistema tecnológico centrado en la gestión del conocimiento en las empresas e instituciones tiende a crecer al igual que lo hace el contexto en el que se encuentra. El crecimiento puede tener lugar de dos formas, bien debido a que hay un mayor número de usuarios que interactúan con el ecosistema, o bien debido a que los componentes del ecosistema tienen que evolucionar, se incorporan componentes adicionales para cubrir nuevas necesidades o se cambia unos componentes por otros. En ambos casos el conocimiento gestionado, si el ecosistema está bien definido y cubre las necesidades de la entidad, crecerá de manera continua a lo largo del tiempo.

Cada vez son más las empresas que utilizan servicios en la nube o *cloud computing* para mejorar la escalabilidad de sus servicios y poder hacer frente al crecimiento de la demanda de manera continua o puntual [2]. Resulta por tanto lógico que las ventajas del uso de una infraestructura que siga el paradigma del *cloud computing* pueden extenderse a los ecosistemas tecnológicos y concretamente a los ecosistemas *eLearning* [28].

Para permitir la escalabilidad en un ecosistema tecnológico basado en *cloud computing* se debe hacer uso de la virtualización, de tal manera que no exista una barrera física entre los recursos *hardware* y la gestión de los mismos. La virtualización de la infraestructura sobre la que se sustenta el ecosistema tecnológico se realiza mediante tecnologías que permiten la monitorización y administración de los recursos, las Máquinas Virtuales [29].

Bo et al. [30] realizan una propuesta de arquitectura basada en *cloud computing* para los ecosistemas *eLearning*, aunque también es extensible a otros ecosistemas tecnológicos. Estos autores plantean una arquitectura en tres capas, desde la infraestructura hasta la aplicación, y un conjunto de módulos que permiten:

- Monitorizar los recursos utilizados en el *cloud*.
- Gestionar automáticamente los recursos del *cloud* en función de las estrategias *eLearning* definidas y el uso de los recursos para llevar a cabo dichas estrategias.
- Gestionar manualmente algunas políticas del *cloud*.
- Reasignar los recursos a partir de las órdenes recibidas por los módulos de monitorización y políticas.

Este conjunto de módulos, además de asegurar la escalabilidad, permiten disponer de mecanismos que detecten problemas antes de que afecten al funcionamiento del sistema de cara al usuario, y además aseguran la calidad del servicio y la disponibilidad del mismo.

3. Casos de estudio

Se introducen someramente tres casos de estudios que sirven como experiencia acumulada y evolución tecnológica del concepto de ecosistema digital de

aprendizaje. En cada uno de los casos ha realizado el análisis DAFO cuyo resumen comparativo se recoge en la Fig. 2 y que servirá de base para la definición de la propuesta arquitectónica objeto del presente trabajo.

3.1. Universidad Virtual

Dentro del contexto de la Universidad de Salamanca, surge en 2008, de la mano del Vicerrectorado de Innovación Tecnológica el proyecto Estrategia Digital 2010 con el objetivo de “definir, adquirir e implementar la infraestructura física y lógica adecuada para que los procesos de formación, investigación y gestión se puedan llevar a cabo utilizando la tecnología como una herramienta o un medio que los facilite o posibilite, sin que por ello esta tecnología suponga una barrera añadida a su desarrollo. Este objetivo se afronta desde una perspectiva estratégica que concluya con una solución integrada, coherente y accesible” [31].

Se plantea la definición e implantación del ecosistema tecnológico institucional para gestionar todo el conocimiento de la Universidad, tanto de forma interna, para lo que proporciona los medios necesarios para llevar a cabo los procesos de formación, investigación y gestión, como de forma externa, ya que aporta visibilidad a todo el conocimiento que posee una universidad con la trayectoria de la Universidad de Salamanca, con casi 800 años de historia. Son muchas las implicaciones, técnicas, metodológicas y de gestión, que conlleva un proyecto de esta envergadura por lo que este caso de estudio se centra únicamente en cuatro de los principales hitos del proyecto Estrategia Universidad Digital 2010 [31]: el repositorio institucional que permite la gestión documental de todo el conocimiento producido dentro de la institución, GREDOS (Gestión del Repositorio Documental de la Universidad de Salamanca) (<http://gredos.usal.es>); el nuevo portal web institucional que proporciona la infraestructura necesaria para dar soporte a la gestión de la información pública de todos los órganos, institutos, facultades, departamentos, etc. existentes dentro de la Universidad (<http://www.usal.es>); el Campus Virtual de la Universidad de Salamanca, Studium (<http://studium.usal.es>), puesto en marcha desde la Universidad Virtual; y el gestor de *blogs*, Diarium (<http://diarium.usal.es>), para proporcionar a la comunidad universitaria un espacio personal donde poder gestionar su propio conocimiento.

3.2. GRIAL 2.0. Una propuesta de integración de servicios y aplicaciones web en un portal académico personalizable

La definición y desarrollo del ecosistema tecnológico de GRIAL se plantea como un portal académico 2.0 dentro del proyecto de fin de carrera de García Holgado [32]. Se parte de la base del *know-how* adquirido a través de la experiencia con el ecosistema tecnológico de la Universidad de Salamanca para plantear un ecosistema formado por cuatro componentes principales, el portal web del Grupo (<http://grial.usal.es>), la plataforma de *eLearning* (<http://grial.usal.es/polis>), el gestor de *blogs* (<http://grial.usal.es/agora>)

y el repositorio documental (<http://grialospace.usal.es:443>), conectados entre sí a través de una serie de flujos de información.

El ecosistema de GRIAL supone un paso adelante respecto al ecosistema institucional desarrollado en el marco del proyecto Universidad Digital de Salamanca. Se han planteado soluciones a las debilidades detectadas, intentando potenciar las fortalezas y ofreciendo una mayor consistencia y adaptabilidad frente a la evolución de todo el sistema.

3.3. Tagging, Recognition and Acknowledgment of Informal Learning Experiences (TRAILER)

El tercer caso de estudio se desarrolla en el marco del proyecto europeo TRAILER (*Tagging, Recognition and Acknowledgment of Informal Learning Experiences*) (<http://trailerproject.eu>) [33] [34] cuyo objetivo principal es la elaboración de un ecosistema para la gestión de competencias adquiridas mediante aprendizaje informal desde la perspectiva del usuario como de la institución o empresa.

Los resultados principales que se desea alcanzar una vez finalizado el proyecto son dos. En primer lugar, la definición y aplicación de un conjunto de metodologías y recomendaciones para la integración del aprendizaje informal en las instituciones y organizaciones. En segundo lugar, un ecosistema que proporcione el marco tecnológico adecuado para la gestión de competencias, especialmente aquellas adquiridas de manera informal, de tal forma que proporcione las herramientas necesarias para llevar a cabo dicha gestión desde el punto de vista de todos los actores involucrados, tanto los individuos como la institución a la que estos pertenecen.

En la Fig. 1 se muestran los cuatro elementos principales del ecosistema, el portfolio, el catálogo de competencias, el *Informal Learning Collector* (ILC) [35] y el entorno institucional que proporciona herramientas de toma de decisiones [36]. La integración de los diferentes componentes se ha realizado mediante servicios web.

El ecosistema tecnológico de TRAILER introduce grandes cambios tanto metodológicos como tecnológicos respecto a los planteamientos analizados en los casos de estudio previos.

4. Patrón arquitectónico para un ecosistema tecnológico de aprendizaje

Para llevar a cabo el análisis comparativo a partir de las matrices DAFO obtenidas en los casos de estudio, se ha realizado una tabla que contiene el análisis interno de cada uno de los ecosistemas tecnológicos planteados (Fig. 2). La tabla se ha dividido en tres columnas, una por cada ecosistema analizado, colocadas por orden temporal. Las debilidades y las fortalezas se han agrupado en filas, para lo que se ha utilizado como criterio una serie de características básicas entre las que se incluyen los cuatro ejes fundamentales. Para distinguir

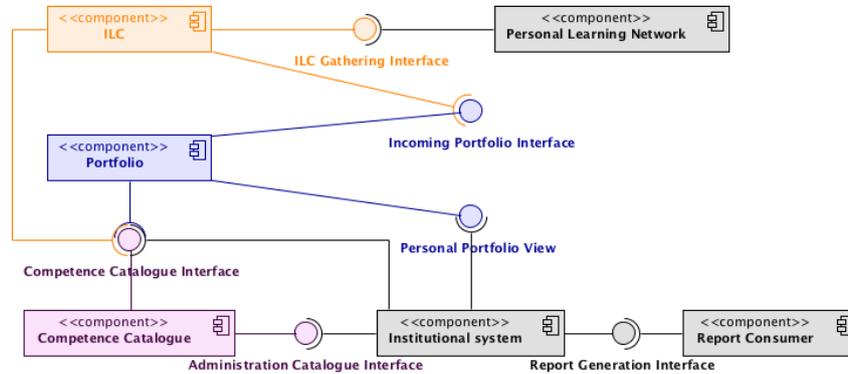


Figura 1. Arquitectura del ecosistema TRAILER [34]

las debilidades de las fortalezas, y poder ver la evolución del sistema de manera algo más visual, se ha utilizado el color rojo para indicar las debilidades y el negro para las fortalezas.

El análisis de las diferentes características a lo largo de los diferentes ecosistemas tecnológicos estudiados proporciona el marco adecuado para definir una serie de puntos en común que debe tener un ecosistema tecnológico:

- Establecimiento de una sólida base metodológica y de gestión de proyecto y riesgos.
- Definición clara de los procesos y *workflows* necesarios para gestionar el ecosistema.
- Gestión centralizada de usuarios, con cobertura tanto de la gestión de los datos como de la autenticación en todo el ecosistema.
- Gestión centralizada de los datos estáticos, para que puedan ser utilizados en los diferentes componentes del sistema.
- Integración de los componentes de manera transparente, para así asegurar la flexibilidad y la adaptabilidad del sistema frente a los cambios, es decir, se debe contar con un plan de aseguramiento de la evolución del ecosistema.
- Potenciación de la reutilización de los componentes del ecosistema, lo que a su vez se transforma en un beneficio para la evolución de los diferentes componentes.
- Integración a nivel de presentación que transmita unicidad, de tal forma que el usuario sea consciente de que se encuentra en el ecosistema.
- Fuerte componente social que permita la integración con herramientas sociales.
- Soporte para la toma de decisiones y para el análisis de los flujos de información que tienen lugar tanto dentro del ecosistema como provenientes del exterior y viceversa.
- Uso de *software Open Source* como base para el desarrollo de los componentes del sistema con el fin de beneficiarse de las ventajas de este tipo de *software*.

| | Universidad Virtual | GRIAL 2.0 | TRAILER |
|---------------------------|---|--|---|
| Metodología | El ecosistema tecnológico se sustenta sobre una sólida gestión técnica y metodológica involucrando a diferentes servicios de la Universidad coordinados desde el Vicerrectorado de Innovación Tecnológica pero con poder para la toma de decisiones acerca del componente del ecosistema en cuya definición, implantación y gestión están implicados. | El ecosistema tecnológico se sustenta sobre una sólida gestión técnica y metodológica proporcionada por los miembros del grupo de investigación. | El ecosistema tecnológico se sustenta sobre una sólida gestión técnica y metodológica que involucra a diferentes universidades europeas coordinadas por el grupo GRIAL de la Universidad de Salamanca. |
| Novedad | Aunque hay en el mercado soluciones software que cubren diferentes aspectos del ecosistema, las peculiaridades que tiene una universidad no están cubiertas por dicho software. El ecosistema tecnológico planteado permite establecer una conexión entre los diferentes componentes, lo que permite llegar a una integración en el nivel de gestión de los datos y adaptarse a las necesidades de la institución. | Aunque hay en el mercado soluciones software que cubren diferentes aspectos del ecosistema, las peculiaridades que tiene un grupo de investigación no están cubiertas por dicho software. El ecosistema tecnológico planteado permite establecer una conexión entre los diferentes componentes para proporcionar una funcionalidad adicional que no existe si se toman estos por separado. | No existe en el mercado ninguna solución software que permita la gestión integral de competencias, tanto adquiridas de manera formal como informal, dentro de una institución o empresa. |
| Usuarios | El sistema es dependiente de la gestión de usuarios existente. | A pesar de que la autenticación de usuarios sea centralizada, no existe una gestión centralizada integral de usuarios, lo que impide un control global de los mismos. El uso de un servidor único de autenticación de usuarios introduce una capa de seguridad adicional. | No existe una gestión centralizada de los usuarios, tanto a nivel de datos como a nivel de autenticación, lo que supone un paso atrás en la usabilidad del ecosistema. |
| Información | El repositorio permite un primer nivel de centralización de datos de diversa procedencia institucional (investigación, docencia, archivo institucional y acervo histórico), además de una clasificación y etiquetado semántico. Amplia difusión de los contenidos al integrar el repositorio con los principales rectorates. | Los datos del repositorio documental no son accesibles de manera transparente desde cualquier punto del ecosistema, sino que su utilización fuera del repositorio se hace de forma manual. Existe un primer nivel de centralización de los datos gracias al repositorio documental. | Existe una gestión centralizada de las competencias gestionadas dentro del ecosistema. |
| Integración | La integración entre los componentes no se ha realizado únicamente a nivel de lógica de negocio y de presentación, sino que ciertos flujos de información conllevan la integración a nivel de datos, lo que genera dependencias entre las aplicaciones. La integración a nivel de presentación no sigue una única identidad corporativa de tal forma que cada elemento, aunque se mantienen diseños similares y se sigue un diseño institucional, no proporcionaba unicidad. Interoperabilidad limitada a ciertas conexiones ad-hoc. | La integración a nivel de presentación mantiene la identidad corporativa pero sigue existiendo una falta de unicidad que muestre el ecosistema como un todo. La integración se ha llevado a cabo mediante el uso de servicios web, con lo que se mantiene la independencia entre los componentes y se proporciona una funcionalidad adicional al establecer ciertos flujos de trabajo. | La integración a nivel de presentación mantiene la identidad corporativa pero sigue existiendo una falta de unicidad que muestre el ecosistema como un todo. El entorno institucional ad-hoc es totalmente dependiente del resto de componentes, a pesar de que la integración se ha realizado mediante servicios web, existen fuertes dependencias a nivel de lógica de negocio. web, gracias a lo que se consigue una independencia entre los componentes y se proporciona una funcionalidad adicional al establecer ciertos flujos de trabajo. |
| Movilidad | No existe soporte para dispositivos móviles. | No existe soporte para dispositivos móviles. | El portafolio posee soporte para dispositivos móviles mediante una aplicación desarrollada ad-hoc. |
| Social | No existe integración con redes sociales, de tal forma que la capa social se limita a los blogs. | Integración de diferentes herramientas sociales como componentes satélite del ecosistema. | No existe una explotación del carácter social en la gestión de las competencias, lo cual es una de sus mayores oportunidades. |
| Toma de decisiones | La toma de decisiones se basa en informes ad-hoc no automatizados. | Los flujos de diseminación permiten un mayor impacto de la información generada en el ecosistema pero no existe una visión global del impacto conseguido mediante dichos flujos. No existe ningún tipo de monitorización sobre el funcionamiento del ecosistema, no disponiendo de la información necesaria para tomar decisiones relativas a la evolución del mismo. No proporciona las herramientas necesarias para establecer flujos de trabajo relacionados con la toma de decisiones. | Existe una visión global de los flujos de información gracias a las herramientas de toma de decisiones incorporadas en el entorno institucional. |
| Evolución | La solución planteada a la hora de satisfacer los requisitos de información en el portal web introduce una fuerte resistencia al cambio. A pesar de realizar un desarrollo por módulos sustentado en un CMS muy flexible, la gran cantidad de funcionalidad que se implementó no tuvo en cuenta permitir la evolución del sistema. La versión de Drupal que hay disponible actualmente mejora considerablemente la versión utilizada en su momento en el portal web, pero la actualización del mismo a una nueva versión supondría un desarrollo desde cero. La gran envergadura del ecosistema requiere de un gran trabajo de mantenimiento y de un constante seguimiento para asegurar el correcto funcionamiento tanto desde el punto de vista metodológico como técnico. La gestión de usuarios basada en un sistema independiente a cada uno de los componentes que forman el ecosistema tecnológico permite asegurar que el sistema crezca y evolucione, que permite añadir nuevos componentes o cambiar los existentes con el fin de ajustarse a las necesidades cambiantes que surgen a lo largo de los años. comunidad universitaria. | Las soluciones software desarrolladas para satisfacer los requisitos de los diferentes componentes del sistema permiten la evolución del mismo, de tal forma que la actualización de alguno de los elementos software utilizados conlleva un esfuerzo mínimo de desarrollo. | La arquitectura planteada permite la evolución de cada uno de los componentes del ecosistema de manera independiente e introducir nuevos componentes propios de la institución o empresa que despliegue el ecosistema TRAILER. |
| Reutilización | Algunos de los desarrollos ad-hoc llevados a cabo no permite ningún tipo de reutilización. | Los desarrollos ad-hoc, a pesar de cubrir necesidades particulares del ecosistema GRIAL, pueden ser reutilizados en otros entornos. | El catálogo de competencias desarrollado ad-hoc puede ser reutilizado en otros entornos. |
| Open Source | El uso de software Open Source permite realizar el desarrollo dentro de la propia Universidad con todas las ventajas que implica, tanto económicas como estratégicas. | El uso de software Open Source permite realizar el desarrollo dentro del propio grupo de investigación con todas las ventajas que implica, tanto económicas como estratégicas y de adquisición de conocimientos. | El uso de software Open Source permite desarrollar el ecosistema manteniendo unos costes mínimos y siguiendo la política de uso de software libre promovida dentro de la Unión Europea. |
| Desarrollo | El desarrollo incremental del ecosistema permite poner en explotación los diferentes componentes a medida que se realizaba su despliegue e integración. | El desarrollo incremental del ecosistema permite poner en explotación los diferentes componentes a medida que se realizaba su despliegue e integración. | El desarrollo por componentes permite reducir los tiempos de desarrollo y de mantenimiento del ecosistema global. |
| Despliegue | La resistencia al cambio propia de cualquier despliegue de nuevas tecnologías se ha tenido en cuenta y se ha cubierto mediante una serie de planes estratégicos de formación, asesoramiento y evaluación y gestión de la calidad. | La resistencia al cambio propia de cualquier despliegue de nuevas tecnologías se ha cubierto mediante una serie de manuales así como un soporte técnico continuo a través de correo electrónico. | La resistencia al cambio propia de cualquier despliegue de nuevas tecnologías se ha tenido en cuenta y se ha cubierto mediante una serie de talleres y experiencias piloto que han permitido definir los planes de explotación y puesta en marcha del ecosistema en aquellas instituciones y empresas interesadas en darle valor a las competencias adquiridas de manera informal por sus estudiantes o empleados. |

Figura 2. Análisis comparativo de los ecosistemas tecnológicos

Cabe destacar la posibilidad de la modificación del código, el soporte de una extensa comunidad de desarrolladores, así como la reducción de costes.

- Definición de las estrategias y los planes de formación e inmersión necesarios para facilitar la aceptación del ecosistema por parte de sus usuarios finales.

Tomando como punto de partida las características que debe tener un buen ecosistema tecnológico, se ha definido un patrón arquitectónico de cinco capas que permita sentar las bases del ecosistema teniendo siempre presente la flexibilidad y adaptabilidad de sus componentes con el objetivo de permitir la evolución de todos ellos (Fig. 3).

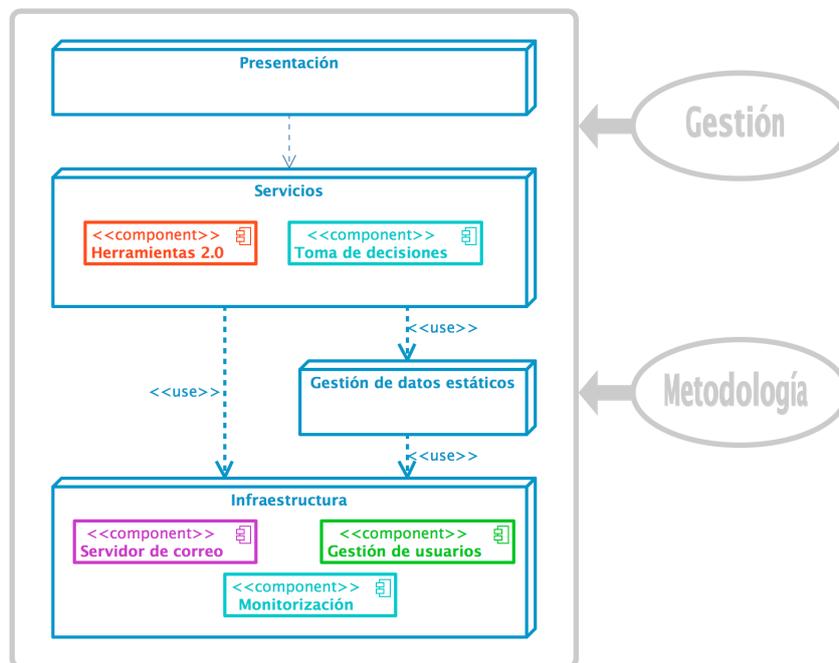


Figura 3. Propuesta de patrón arquitectónico

5. Conclusiones

Los casos de estudio que se han analizado en este trabajo han permitido obtener una visión global de los ecosistemas tecnológicos, tanto desde un punto de vista teórico como práctico. Poder estudiar y analizar estas experiencias desde todas las perspectivas proporciona una visión que no se tenía antes de realizar

este trabajo. Cada caso de estudio, en el momento que se definió y desarrolló se había basado en los casos previos pero no existía un trabajo que agrupara todo este conocimiento.

El patrón arquitectónico propuesto supone un paso adelante en la evolución de los ecosistemas tecnológicos. La posterior experiencia llevada a cabo en la Administración Pública para gestionar todo el conocimiento que se genera dentro de la misma ha demostrado que el patrón propuesto funciona bien aunque se han detectado algunos puntos débiles que se deben mejorar.

El concepto de ecosistema tecnológico así como el patrón o patrones arquitectónicos sobre los que se sustenta, deben mantenerse en constante evolución de tal forma que cada vez se adapte mejor a las necesidades cambiantes de los usuarios y las entidades sin abandonar la apuesta por el desarrollo *Open Source*.

Este trabajo sienta las bases para la realización de una futura tesis doctoral enmarcada en el contexto de los ecosistemas tecnológicos. Las posibles líneas de investigación que quedan abiertas tras este estudio son las siguientes:

- Mejora en la integración de soluciones tecnológicas *Open Source* con el objetivo de dejar de lado la integración *ad-hoc* de los componentes de un ecosistema.
- Determinar soluciones sobre la gestión de la vista en un ecosistema, transmitiendo una uniformidad que hasta ahora no se ha logrado por las dependencias de interfaz que existen en los diferentes componentes.
- Necesidad de un mayor grado de flexibilidad y adaptabilidad a los diferentes problemas y usuarios.
- Sistemas de toma de decisiones a partir del análisis de flujos de información.

6. Agradecimientos

El presente trabajo ha sido financiado con el apoyo de la Junta de Castilla y León dentro del proyecto GR47 “eLearning sin barreras: Nuevos paradigmas de comunicación, servicios y modalidades de interacción para la formación en línea” (<http://grial.usal.es/grial/node/80>) y de la Comisión Europea dentro del proyecto 519141-LLP-1-2011-1-ES-KA3-KA3MP “Tagging, Recognition and Acknowledgment of Informal Learning Experiences (TRAILER)” (<http://trailerproject.eu>).

Bibliografía

- [1] T. O'Reilly, "What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software," *Communications & Strategies*, vol. 1, p. 17, 2007.
- [2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [3] A. Weiss, "Computing in the Clouds," *netWorker*, vol. 11, pp. 16–25, December 2007.
- [4] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.
- [5] R. M. Stallman and G. E. Manual, "Free Software Foundation," *El proyecto GNU-Fundación para el software libre*, 1999.
- [6] O. S. Initiative, "The Open Source Definition." <http://opensource.org/osd>.
- [7] E. Commission, "European Union Public Licence (EUPL)." <http://joinup.ec.europa.eu/software/page/eupl>.
- [8] E. C. D.-G. for Informatics (DIGIT), "Strategy for internal use of OSS at the EC." http://ec.europa.eu/dgs/informatics/oss_tech/index_en.htm.
- [9] E. y. T. Ministerio de Industria and J. de Extremadura, "Centro Nacional de Referencia de Aplicación de las TIC basadas en fuentes abiertas." <http://www.cenatic.es>.
- [10] C. Darwin and G. Beer, *The origin of species*. Oxford University Press, 1951.
- [11] M. A. Conde, F. García-Peñalvo, M. J. Rodríguez-Conde, M. Alier, and A. García-Holgado, "Perceived openness of Learning Management Systems by students and teachers in education and technology courses," *Computers in Human Behavior*, 2013.
- [12] L. Uden, I. T. Wangsa, and E. Damiani, "The future of E-learning: E-learning ecosystem," in *Digital EcoSystems and Technologies Conference, 2007. DEST'07. Inaugural IEEE-IES*, pp. 113–117, IEEE, 2007.
- [13] F. J. García-Peñalvo, M. J. Rodríguez Conde, A. M. Seoane Pardo, V. Zangrando, and A. García Holgado, "GRIAL (GRupo de investigación en Interacción y eLearning), USAL," *Revista Iberoamericana de Informática Educativa*, 2012.
- [14] C. Ferran and R. Salim, "Enterprise resource planning for global economies," *Idea Group Inc (IGI)*, 2008.
- [15] J. Esteves and J. Pastor, "An ERP lifecycle-based research agenda," in *First International workshop in Enterprise Management and Resource. Planning: Methods, Tools and Architectures-EMRPS*, vol. 99, 1999.

- [16] E. Commission, "Digital Ecosystems: The New Global Commons for SMEs and local growth." <http://www.digital-ecosystems.org/doc/papers/d5-intro-for-the-press.pdf>, 2006.
- [17] E. Chang and M. West, "Digital Ecosystems A Next Generation of the Collaborative Environment," *the Eight International Conference on Information Integration and Web-Based Applications & Services*, vol. 214, pp. 3–23, 2006.
- [18] K. Pata, "Meta-design framework for open learning ecosystems." <http://tihane.wordpress.com/2011/06/09/meta-design-framework-for-open-learning-ecosystems-at-mupple-lecture-series/>, 2011.
- [19] M. Laanpere, "Digital Learning ecosystems: rethinking virtual learning environments in the age of social media." <http://es.slideshare.net/martlaa/digital-learning-ecosystems>, 2012.
- [20] M. Berthelemy, "Definition of a learning ecosystem." <http://www.learningconversations.co.uk/main/index.php/2010/01/10/the-characteristics-of-a-learning-ecosystem?blog=5>, 2013.
- [21] C. Pirie, "E-learning Ecosystems: The Future of Learning Technology." http://clomedia.com/articles/view/e_learning_ecosystems_the_future_of_learning_technology/2, 2004.
- [22] J. Brodo, "Today's Ecosystem of eLearning." http://www.enebuilder.net/salesmarketing/e_article000615779.cfm, 2006.
- [23] I. K. Ficheman and R. d. D. Lopes, "Digital Learning Ecosystems: Authoring, Collaboration, Immersion and Mobility," in *Advanced Learning Technologies, 2008. ICAIT '08. Eighth IEEE International Conference on*, pp. 371–372, 2008.
- [24] A. Littlejohn, "The Dynamics of Learning Ecosystems: literacies and resources." <http://littlebylittlejohn.com/learning-ecosystem/>, 2011.
- [25] T. V. Ark, "From LMS to Learning Ecosystem." <http://getttingsmart.com/2012/06/from-lms-learning-ecosystems/>, 2012.
- [26] G. Adomavicius, J. C. Bockstedt, A. Gupta, and R. J. Kauffman, "Technology roles in an ecosystem model of technology evolution," *International Journal of Information Technology and Management - IJITM*, 2005.
- [27] T. Haymes, "Toward a Model for a Technology Ecosystem." <http://www.nmc.org/news/toward-model-technology-ecosystem>, 2012.
- [28] G. Briscoe and A. Marinos, "Digital ecosystems in the clouds: towards community cloud computing," in *Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference on*, pp. 103–108, 2009.
- [29] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, p. 718, 2010.
- [30] D. Bo, Z. Qinghua, Y. Jie, L. Haifei, and Q. Mu, "An E-learning Ecosystem Based on Cloud Computing Infrastructure," in *Advanced Learning Technologies, 2009. ICAIT 2009. Ninth IEEE International Conference on*, pp. 125–127, 2009.

- [31] V. de Innovación Tecnológica, “Proyecto Estrategia Digital 2010.” <http://www.usal.es/webusal/node/1258>.
- [32] A. García Holgado, “GRIAL 2.0 Una propuesta de integración de servicios y aplicaciones web en un portal académico personalizable,” Master’s thesis, University of Salamanca, March 2011.
- [33] F. J. García-Peñalvo, M. á. Conde-González, V. Zangrando, A. García-Holgado, A. M. Seoane Pardo, M. Alier, N. Galanis, F. Brouns, H. Vogten, D. Griffiths, A. Mykowska, G. Ribeiro Alves, and M. Minović, “TRAILER project (Tagging, Recognition, Acknowledgment of Informal Learning Experiences). a Methodology to make visible learners’ informal learning activities to the institutions,” *Journal of Universal Computer Science*, 2013.
- [34] F. J. García-Peñalvo, V. Zangrando, A. García-Holgado, M. á. Conde-González, A. M. Seoane Pardo, M. Alier, J. Janssen, D. Griffiths, A. Mykowska, and G. Ribeiro Alves, “TRAILER project overview: Tagging, Recognition and Acknowledgment of Informal Learning Experiences,” in *Computers in Education (SIEE), 2012 International Symposium on*, pp. 1–6, IEEE, 2012.
- [35] M. Alier, N. Galanis, E. Mayor, J. Piguillem, M. J. Casany, F. J. García Peñalvo, and M. Á. Conde González, “Del ple al lms+portfolio: Integración de elementos de aprendizaje informal en moodle + mahara,” in *Moodle-Moot Madrid 2012. Globalización y Socialización (Madrid, España, 19-21 de septiembre de 2012)*, pp. 24–29, 2012.
- [36] F. J. García-Peñalvo and M. á. Conde González, “Knowledge management and decision making based on informal learning activities in business,” in *Proceedings of the 2nd Global Innovation and Knowledge Academy (GIKA 2013) (Valencia, Spain, July 9-11, 2013)*, 2013.

Modelado Matemático para la Generación de un Mallado Espacial con Aplicación a Nano-Estructuras Semiconductoras

Sergio García¹, José Rafael García-Bermejo² e Ignacio Íñiguez-de-la-Torre¹

¹ Departamento de Física Aplicada, Universidad de Salamanca. Plaza de la Merced, s/n. 37008 Salamanca, España
`{sergio_gs,indy}@usal.es`

² Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos, s/n. 37008 Salamanca, España
`coti@usal.es`

Resumen En este trabajo se presentan distintos métodos matemáticos para resolver la ecuación de Poisson en 2D, concretamente los métodos de factorización LU, Jacobi, Gauss-Seidel y sobre-relajación sucesiva (SOR); para dos criterios de convergencia. Estos métodos serán implementados en un potente simulador Monte Carlo (MC) de dispositivos desarrollado por el Grupo de Investigación de Dispositivos Semiconductores de la Universidad de Salamanca (USAL) para poder analizar un diodo vertical de GaAs y un diodo autoconmutante de GaAs. El método de factorización proporciona una solución exacta del potencial en los distintos nodos en los que se discretiza el dispositivo electrónico. En lo tocante al tiempo de simulación, el método SOR es competitivo con la técnica de factorización LU. Se descarta la utilización de los métodos de Jacobi y Gauss-Seidel cuando se trabaja con matrices cuadradas de orden superior a 100 por los elevados tiempos de simulación que se precisan para los dos criterios de convergencia. Un dispositivo electrónico se discretiza a través de un mallado formado por contornos rectangulares. Dado que el número de nodos del mallado es un factor muy importante en el tiempo de simulación (concretamente en la resolución de la ecuación de Poisson), se presenta un algoritmo de mallado que respeta cierta restricción física (la longitud de Debye tiene que ser menor que el tamaño de las mallas en las diferentes regiones) y además, se evitan en las fronteras cambios bruscos en los tamaños de las mallas.

Keywords: Métodos Numéricos, Ecuación de Poisson, Simulación Monte Carlo, Elementos Finitos, Diferencias Finitas

1. Introducción

Una de las tareas más difíciles, sin tener en cuenta su física, a la hora de llevar a cabo el estudio de un dispositivo electrónico a través de una simulación Monte Carlo [1] consiste en determinar cada una de sus regiones donde se debe

resolver la ecuación de Poisson con las condiciones de contorno apropiadas. En este trabajo se hará una investigación de las diferentes formas de resolver la ecuación de Poisson en 2D mediante distintos métodos matemáticos [2], [3]. Se estudiará con detalle la discretización de la ecuación de Poisson mediante el método de diferencias finitas y se propondrán cuatro métodos matemáticos [4] para resolver tal ecuación, siendo estos el método directo de factorización LU y los métodos de relajación de Jacobi, Gauss-Seidel y sobre-relajación sucesiva (SOR).

Por otra parte, crear un mallado rectangular para definir la topografía de los dispositivos electrónicos no es una tarea sencilla. Especialmente si su geometría y/o composición de materiales es complicada, por intervenir factores físicos que hacen que ésta no sea homogénea para una estructura, y tenga que tener características especiales. Además, cuanto mayor sea el número de mallas, mayor será el tiempo de simulación; de ahí que sea importante la creación de un algoritmo óptimo de mallado que proporcione automáticamente el número de mallas horizontales y verticales para la definición del dispositivo electrónico. Se investigarán distintas formas para la creación del mallado y se diseñará un algoritmo para elaborar un mallado adaptativo respetando las restricciones físicas (la longitud de Debye), donde el número de nodos máximo vendrá impuesto por el usuario, definiendo así de forma optimizada la posición de los nodos para garantizar una alta precisión en la magnitud a analizar. Mediante el algoritmo diseñado se estudian dos diodos, un diodo vertical de GaAs y un self-switching diode (SSD) de GaAs.

Todas las simulaciones que se presentan en esta memoria han sido realizadas por medio del simulador Monte Carlo de dispositivos que posee el Grupo de Investigación en Dispositivos Semiconductores de la USAL.

2. La ecuación de Poisson dentro del simulador Monte Carlo

2.1. Aspectos generales del simulador Monte Carlo

El método de Monte Carlo en Electrónica [5] consiste en un tratamiento microscópico mediante el cual se estudia la dinámica [6], [7] de los portadores que se mueven en el interior de un material semiconductor sometidos a la acción de un campo eléctrico auto-consistente, que se actualiza cada cierto paso temporal resolviendo la **ecuación de Poisson** de acuerdo con la posición de los portadores y la distribución del dopaje, y de la red cristalina [7], [8], [9]. Para poder discretizar en el espacio y resolver la ecuación de Poisson en 2D, dada la dificultad que conlleva la obtención de una solución analítica que puede ser atribuida a la complejidad natural de la ecuación diferencial o a las dificultades que pueden ocurrir con las condiciones de contornos, se utilizan los métodos de elementos finitos o de diferencias finitas [2], [3]. Para obtener la dinámica de los portadores dentro de un semiconductor es necesario resolver la ecuación de Poisson

$$\nabla[\epsilon_0\epsilon_s\nabla\psi(r,t)] = -\rho(r,t), \quad (1)$$

donde ψ es el potencial eléctrico en cada punto, ρ la densidad de carga, ϵ_0 la constante dieléctrica del vacío y ϵ_s la permitividad relativa estática. Para resolver esta ecuación empleamos el método de las diferencias finitas, siendo necesaria la discretización del dispositivo electrónico estudiado en un mallado de n columnas y m filas. Este mallado no tiene por qué ser uniforme, lo que facilitará una mejor resolución del campo eléctrico en aquellos lugares en los que su inhomogeneidad sea más pronunciada. La ecuación de Poisson en el caso bidimensional se puede expresar como

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\frac{\rho(x, y)}{\epsilon_0 \epsilon_s}, \quad (2)$$

donde ϵ_s es constante en cada malla. Nuestro sistema tendrá $n \times m$ mallas de distintos tamaños y $(n+1) \times (m+1)$ nodos. La resolución de la ecuación de Poisson se lleva a cabo en dos fases. Una primera de asignación de carga, donde se calculan las densidades de carga eléctrica que se atribuyen a los nodos del dispositivo [7], las cuales se emplearán en la segunda fase, que consiste en la resolución del sistema de ecuaciones resultantes de la discretización de la Ecuación 2. Aplicando la discretización (método de diferencias centradas) y resolviendo el sistema resultante se determinan los potenciales asociados a cada uno de los nodos. En la Figura 1 se observa el esquema que se emplea para resolver la ecuación de Poisson. Con esta técnica se calcula el potencial asociado a cada uno de los nodos

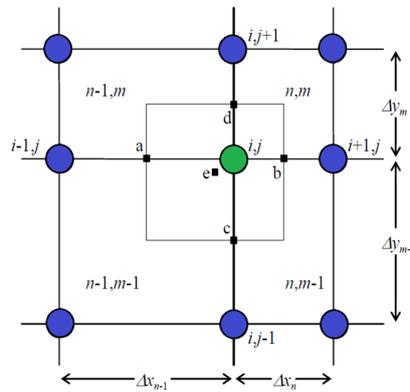


Figura 1. Esquema para la resolución de la ecuación de Poisson. Método de las diferencias centradas.

del mallado y el campo eléctrico asociado a cada una de las celdas. Aplicando la

discretización a la Ecuación 2, se obtiene como resultado

$$\begin{aligned}
& \psi_{i,j-1} \frac{2(\Delta x_n + \Delta x_{n-1})}{\Delta y_{m-1}} + \psi_{i-1,j} \frac{2(\Delta y_m + \Delta y_{m-1})}{\Delta x_{n-1}} + \psi_{i+1,j} \frac{2(\Delta y_m + \Delta y_{m-1})}{\Delta x_n} + \\
& \psi_{i,j+1} \frac{2(\Delta x_n + \Delta x_{n-1})}{\Delta y_m} - \psi_{i,j} 2 \left[(\Delta x_n + \Delta x_{n-1}) \left(\frac{1}{\Delta y_m} + \frac{1}{\Delta y_{m-1}} \right) \right] - \\
& - \psi_{i,j} 2 \left[(\Delta y_m + \Delta y_{m-1}) \left(\frac{1}{\Delta x_n} + \frac{1}{\Delta x_{n-1}} \right) \right] = - \left(\frac{\rho_{n-1,m-1} \Delta x_{n-1} \Delta y_{m-1}}{\epsilon_{n-1,m-1}} \right) - \\
& - \left(\frac{\rho_{n,m-1} \Delta x_n \Delta y_{m-1}}{\epsilon_{n,m-1}} + \frac{\rho_{n,m} \Delta x_n \Delta y_m}{\epsilon_{n,m}} + \frac{\rho_{n-1,m} \Delta x_{n-1} \Delta y_m}{\epsilon_{n-1,m}} \right). \quad (3)
\end{aligned}$$

Es de resaltar que los potenciales son magnitudes asociadas a los nodos (subíndices i, j) mientras que las permitividades y las densidades de carga se corresponden con las celdas (subíndices n, m) de la discretización. Aplicando esta discretización a todos los nodos de la red se obtienen $(n+1) \times (m+1)$ ecuaciones lineales.

2.2. Parámetros físicos para la simulación

Para que una simulación Monte Carlo reproduzca el comportamiento del dispositivo de forma correcta, es necesario que se satisfagan dos condiciones:

- El paso temporal está limitado por el inverso de la frecuencia de plasma f_p [10] y por el tiempo de relajación dieléctrico τ_d [11], teniéndose que verificar que $f_p dt < 2$ y $dt < \tau_d$.
- El tamaño de las mallas tiene que ser siempre menor que la longitud de Debye [12], [9]

$$\lambda_d = \sqrt{\frac{\epsilon_0 \epsilon_r k_B T}{q^2 n}}, \quad (4)$$

siendo q el valor absoluto de la carga del electrón, k_B la constante de Boltzmann, T la temperatura, $\epsilon_0 \epsilon_r$ la permitividad del material y n la concentración de portadores.

3. Métodos de resolución de la ecuación de Poisson. Análisis y comparación de resultados

Para resolver el sistema de ecuaciones dado por la discretización de diferencias finitas (Ecuación 3), los métodos típicos que se utilizan son: [4]:

- Métodos matriciales directos: gradiente conjugado, basados en eliminación Gaussiana y en factorización (LU).
- Métodos de relajación: Jacobi, Gauss-Seidel, sobre-relajación sucesiva (SOR) e implícito de dirección alternante.
- Métodos rápidos: de Fourier y de relajación cíclica.

En este trabajo se estudia el método directo de factorización LU y los métodos de relajación de Jacobi, Gauss-Seidel y SOR.

3.1. Métodos de resolución de la ecuación de Poisson

Métodos matriciales directos: factorización LU.

La descomposición o factorización LU [4], [13] busca expresar una matriz cuadrada regular como producto de una triangular inferior con números 1 en la diagonal, L , y otra triangular superior en la que en la diagonal no necesariamente tiene que haber números 1, U . Su utilidad inmediata radica en que para resolver un sistema de ecuaciones lineales $Ax = b$, si $A = LU$, el problema se reduce a resolver dos sistemas de ecuaciones triangulares. Una forma de conseguir esta factorización LU la constituye la propia eliminación de Gauss, que reduce la matriz original a una triangular superior mediante unas permutaciones y unas transformaciones definidas por una matriz elemental triangular inferior que tiene 1 en la diagonal. El sistema $Ax = b$ es por tanto $L(Ux) = b$. En este punto se introduce una nueva variable (por sustitución) $y = Ux$, obteniendo así el nuevo sistema $Ly = b$.

Una vez en este punto, se resuelve dicho sistema para la variable y mediante sustitución hacia adelante. Como paso final, usamos sustitución hacia atrás para resolver el sistema $Ux = y$. Es destacable, que los sistemas $Ly = b$ y $Ux = y$ son relativamente fáciles de resolver dado que se trata de matrices de coeficientes triangulares inferiores y superiores. Este método proporciona un resultado exacto de la variable x .

Métodos de relajación.

- Método de Jacobi

El siguiente método considerado será el que Carl Gustav Jacobi (1804-1851) desarrolló en 1845 [4], [13]. Su mecánica es muy sencilla donde los vectores en cada iteración (para un sistema cuadrado $n \times n$, suponiendo que $a_{ii} \neq 0$ y que $1 \leq i \leq n$) se generan de la forma siguiente:

$$x_i^{k+1} = \sum_{\substack{j=1 \\ j \neq i}}^n \left(\frac{-a_{ij}x_j^{(k)}}{a_{ii}} \right) + \frac{b_i}{a_{ii}}. \quad (5)$$

El algoritmo que representa el procedimiento iterativo de Jacobi para resolver el sistema de ecuaciones, parte de un punto inicial x^0 dado. La decisión de parar el proceso iterativo se puede basar en cualquier criterio que se estime adecuado, destacando los siguientes:

- Criterio 1 (C1): el número de iteraciones r requerido para reducir el error en un factor 10^{-p} partiendo de una solución inicial arbitraria es $r \approx \frac{1}{2}pJ^2$. El número de iteraciones es proporcional a J^2 , con J la dimensión de la matriz. Cuando se tengan matrices cuyas dimensiones sean mayores que 100×100 el método de Jacobi será muy lento.
- Criterio 2 (C2): se puede imponer a que la parada se produzca cuando

$$\frac{\|x^{(k)} - x^{(k-1)}\|_\infty}{\|x_{max}\|_\infty} < TOL, \quad (6)$$

donde el subíndice ∞ hace referencia a que la condición se ha de cumplir para todos los nodos. En este caso el error está referido a la diferencia entre dos iteraciones consecutivas, pero no respecto de la solución exacta.

■ Método de Gauss-Seidel

Una mejora del algoritmo de Jacobi es el método de Gauss-Seidel [4]. Admitiendo que la solución $x_i^{(k+1)}$ es una mejor aproximación que $x_i^{(k)}$, de la solución real, se podría plantear emplear soluciones de la ecuación i en la iteración $k + 1$ para determinar las soluciones de $x_j^{(k+1)}$ con $j > i$. Esto es, en vez de emplear en la iteración $k + 1$ todas las soluciones de la iteración anterior $x_i^{(k)}$ ($1 \leq i \leq n$), se emplearán los valores recientemente calculados en la iteración $k + 1$ (los $x_i^{(k+1)}$). En tal caso la fórmula de recurrencia ya no será la expresada en la Ecuación 5, sino la siguiente:

$$x_i^{k+1} = \sum_{j=1}^{i-1} \left(\frac{-a_{ij}x_j^{(k+1)}}{a_{ii}} \right) + \sum_{j=i+1}^n \left(\frac{-a_{ij}x_j^{(k)}}{a_{ii}} \right) + \frac{b_i}{a_{ii}}. \quad (7)$$

Para parar el proceso iterativo se consideran, al igual que en el método de Jacobi, dos alternativas

- Criterio 1 (C1): el número de iteraciones r requerido para reducir el error en un factor 10^{-p} partiendo de una solución inicial arbitraria es $r \approx \frac{1}{4}pJ^2$. El número de iteraciones es reducido un factor 2 con respecto al método de Jacobi.
- Criterio 2 (C2): la condición explicada en el método de Jacobi, que se corresponde con la Ecuación 6.

■ Método de sobre-relajación (SOR)

Este método de relajación emplea el concepto de vector residual [4], que se define así:

$$r_i^{(k+1)} = x_i^{(k+1)} - x_i^{(k)}, \quad (8)$$

$$x_i^{(k+1)} = x_i^{(k)} + r_i^{(k+1)}, \quad (9)$$

donde $x_i^{(k+1)}$ y $x_i^{(k)}$ pueden ser las soluciones de la ecuación i dadas por el método de Jacobi o por el de Gauss-Seidel. Evidentemente, se espera de cualquier método que el residual converja a cero con $k \rightarrow \infty$. Modificando la Ecuación 9 según

$$x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k+1)}, \quad (10)$$

donde $0 < \omega < 2$, se puede conseguir una mejora sustancial de la velocidad de convergencia. A los métodos donde $0 < \omega < 1$ se les llama de sub-relajación y a los métodos donde $\omega > 1$ se les llama de sobre-relajación. En nuestras simulaciones consideraremos que el valor de ω tiene que ser:

$$\omega \approx \frac{2}{1 + \pi/J}. \quad (11)$$

Para parar el proceso iterativo en este método se consideran los criterios siguientes:

- Criterio 1 (C1): el número de iteraciones r requerido para reducir el error en un factor 10^{-p} partiendo de una solución inicial arbitraria es $r \approx \frac{1}{3}pJ$. El número de iteraciones es lineal con J .
- Criterio 2 (C2): la condición explicada en el método de Jacobi, que se corresponde con la Ecuación 6.

3.2. Análisis de resultados

En la Figura 2 se muestra el esquema de la geometría de la estructura vertical simulada: diodo con notch ($n^+n^-nn^+$). La zona n_1^+ tiene una longitud de 100

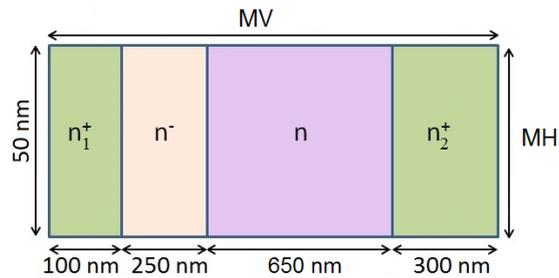


Figura 2. Esquema y geometría de la estructura simulada.

nm y la zona n_2^+ tiene una longitud de 300 nm. El notch n^- tiene 250 nm y la zona activa n tiene una longitud de 650 nm. La dimensión vertical de la estructura simulada es de 50 nm. Se considera una impurificación en el diodo de $n^+ = 1 \times 10^{24} \text{ m}^{-3}$, $n^- = 2 \times 10^{22} \text{ m}^{-3}$ y $n = 1 \times 10^{23} \text{ m}^{-3}$. Se estudia una estructura con 20 mallas verticales ($2 \times 50,0 \text{ nm}$, $3 \times 83,333 \text{ nm}$, $10 \times 65,0 \text{ nm}$ y $5 \times 60,0 \text{ nm}$) y el número de mallas horizontales (MH) variará en función del caso a estudiar.

Análisis de resultados en función de la precisión (p) del método iterativo para $MH = 2$ usando el criterio de convergencia (C1).

Para abordar este apartado se considera un diodo con 2 mallas horizontales. Recapitular que el criterio de convergencia (C1) es aplicado cuando se desea reducir el error en un factor 10^{-p} . Se parte de una solución arbitraria, x^0 inicial, del potencial en cada nodo, que en las simulaciones ha sido de 1 V.

- Método de Jacobi, Gauss-Seidel y sobre-relajación sucesiva frente al de factorización LU

En la Figura 3 (a) se representa la densidad de corriente frente al potencial aplicado entre extremos del diodo en simulaciones donde se ha empleado el

método de Jacobi y el LU. $\forall p, p \in [1, 5]$ se obtiene prácticamente la misma característica $I-V$ en comparación con el resultado logrado tras resolver la ecuación de Poisson por medio del método de factorización LU. En el recuadro de la Figura 3 (a) se representa con un diagrama de barras el tiempo requerido para efectuar la simulación empleando el método de Monte Carlo. Resolviendo la ecuación de Poisson mediante el método de factorización LU se logra el menor tiempo de simulación, $t^{LU} = 5687$ s. Por el contrario, si se utiliza el método de Jacobi se observa que el valor de p (que marcará la precisión del resultado) tiene un peso muy importante en el tiempo de simulación, observándose un comportamiento lineal creciente a medida que p es mayor. En este método y para esta estructura, considerar $p = 1$ proporciona un buen resultado de la característica $I-V$. Relativo al método de Gauss-Seidel se encuentra el mismo resultado de la característica $I-V$ en comparación con el LU. Se obtiene como ventaja, en comparación con la técnica de Jacobi, una reducción (de orden la mitad) en el tiempo empleado para realizar la simulación Monte Carlo. Para el método de sobre-relajación sucesiva (SOR), Figura 3 (b), para el cual se ha considerado que $\omega = 1,905$ (Ecuación 6) observamos que la densidad de corriente no es igual en todos los casos: para $p = 1$ se obtiene un resultado que no se corresponde con la característica encontrada con la técnica de factorización LU. Si se reduce el error en un factor 0,01 ($p = 2$) la característica $I-V$ sí coincide en la zona lineal pero en la zona de saturación no se obtiene el resultado esperado. Se ha comprobado que para otros parámetros de ω ($\omega = 1,2$ y $\omega = 1,5$) y otros valores de x^0 tampoco se alcanzan valores correctos de la $I-V$ ni para $p = 1$ ni para $p = 2$. Para $p = 3$ sí se obtiene la solución deseada tanto para la zona lineal como para la de saturación. Si se compara el tiempo tras emplear factorización LU con los correspondientes tras utilizar el método SOR (con $p = 1, 2, 3, 4$ y 5) no se obtiene una diferencia tan notoria como para los dos casos anteriores (Jacobi y Gauss-Seidel), siendo la mayor para $p = 5$: $t_{p3}^{SOR} - t^{LU} = 826$ s $\approx 13,77$ min.

Análisis de resultados para diferentes números de mallas horizontales (MH) con $p = 4$ usando el criterio de convergencia (C1).

A continuación se presentan los tiempos de CPU cuando se consideran diferentes números de mallas horizontales, $MH = 3, 4$ y 5 , para los métodos matemáticos presentados en este trabajo. Se utiliza siempre el criterio de convergencia (C1) y precisión $p = 4$. Se ha comprobado, aunque no se presentan los resultados en este resumen, que se obtiene en todos los casos las mismas características $I-V$ y el mismo potencial eléctrico. Se encuentra, para un diodo con $MH = 5$ mallas horizontales (126 nodos) que es el método de factorización LU el que proporciona mayor rapidez ($t^{LU} = 7452$ s $\approx 2,07$ h), siendo además, un método exacto de resolución. Con el método SOR se consigue que la simulación finalice en 2.74 h, tiempo más próximo al logrado con el método de factorización LU. El resto de métodos (Jacobi y Gauss-Seidel) la ralentizarían mucho ya

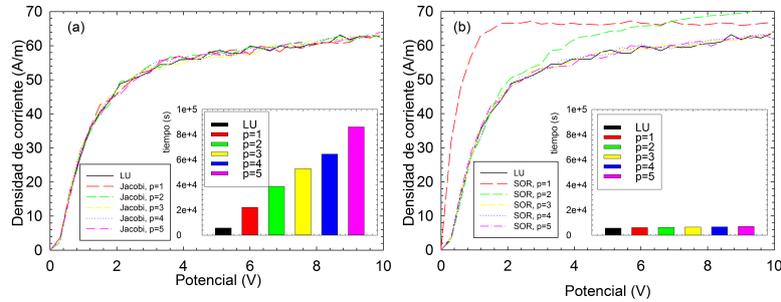


Figura 3. Densidad de corriente frente al potencial aplicado (C1). En el recuadro se representa el tiempo de simulación en función de p . (a) Método de Jacobi frente a la factorización LU y (b) método SOR frente a la factorización LU

que son necesarios días para que ésta finalice: $t_{MH=5}^{Jacobi} = 741400 \text{ s} \approx 8,58$ días y $t_{MH=5}^{GS} = 369260 \text{ s} \approx 4,22$ días. Parece claro pues que al aumentar el número de mallas el único método competitivo con factorización LU es el SOR.

Análisis del tiempo de simulación para $MH = 5$ y usando el criterio de convergencia (C2).

Pasaremos a realizar una optimización de los métodos iterativos donde se tendrá en cuenta el criterio 2 (C2) (Ecuación 6). Según la bibliografía [14], en dispositivos electrónicos el valor de la tolerancia comúnmente utilizado y admitido como suficientemente bueno es $TOL = 10^{-5}$. Es decir, la parada debe producirse cuando el error de la magnitud del potencial en todos los nodos, entre dos iteraciones consecutivas, sea menor que 10^{-5} . Si este criterio no se cumple, la parada se producirá cuando se alcance el número de actualizaciones r de acuerdo con el criterio 1 (C1), tomándose en este caso, $p = 4$. Analizando los tiempos de las simulaciones Monte Carlo de un diodo vertical con $MH = 5$ mallas para los métodos de Jacobi, Gauss-Seidel y SOR para los dos criterios de convergencia estudiados en este trabajo (C1) con $p = 4$ y (C2) donde $TOL = 10^{-(p+1)} = 10^{-5}$ se encuentra que:

- El método de factorización LU, que es un método exacto, proporciona el menor tiempo de simulación $t_1 = 7452 \text{ s} \approx 2,07 \text{ h}$.
- Si se utiliza el método de Jacobi considerándose el C2 se reduce en un $\sim 80\%$ el tiempo de simulación (6.7 días menos de simulación) respecto al C1.
- Si se utiliza el método de Gauss-Seidel considerándose el C2 se reduce en un $\sim 80\%$ el tiempo de simulación (3.29 días menos de simulación) respecto al C1.

- Si se utiliza el método de sobre-relajación sucesiva considerándose el C2 se reduce en un $\sim 1,5\%$ el tiempo de simulación (2.41 minutos menos de simulación) respecto al C1.
- El método SOR proporciona tiempos similares (9720 s) a la técnica de factorización LU.

Esto indica que en los métodos de Jacobi y de Gauss-Seidel el criterio C2 es considerablemente más ventajoso si se compara con el primer criterio (C1). Por el contrario, en el método SOR no existe tanta diferencia entre los tiempos para ambos criterios, indicando esto que para la mayoría de las iteraciones Monte Carlo, el criterio SOR finaliza cuando se alcanzan el número de iteraciones r de acuerdo al criterio 1 (C1).

4. Algoritmo óptimo de discretización de dispositivos electrónicos

Mediante la interfaz gráfica Jersecín [15] el usuario define un dispositivo electrónico indicando los objetos que lo componen (materiales, dieléctricos, contactos con volumen y/o contactos de superficies), y se genera un mallado del dispositivo no uniforme para la resolución de la ecuación de Poisson. Previo a este trabajo, para realizar la discretización de un objeto, el usuario debía indicar el número de filas y columnas en las que lo quería dividir. Con este algoritmo el usuario no tendrá que realizar esta tarea, ya que mediante el algoritmo óptimo de mallado que se propone a continuación, se creará de forma automática el mallado más adecuado para discretizar el dispositivo electrónico. Este algoritmo automático será implementado y probado en la Interfaz de usuario Jersecín. El algoritmo óptimo (V1) elaborará un mallado regular de forma que se respeten las restricciones físicas que impone la longitud de Debye λ_d (asociada a cada material) (Ecuación 4). Como ya se ha dicho anteriormente el tamaño de las mallas tiene que ser siempre menor que la longitud de Debye que es, para cada una de las regiones: $\lambda_d^{n^+} = 4,29$ nm, $\lambda_d^{n^-} = 30,35$ nm y $\lambda_d^i = 13,57$ nm. Además, el usuario tendrá control del número de nodos máximo que intervendrá en la definición del mallado. Por otra parte, es necesario garantizar que no se produzcan cambios bruscos en los tamaños de las mallas al pasar, en la estructura de una región a otra, ya que en este caso la solución de la ecuación de Poisson no sería la correcta. Esto se tiene en cuenta en el algoritmo óptimo versión (V2). Este algoritmo de mallado óptimo (para ambas versiones) se aplica para el estudio de un Self-Switching Diode (SSD) [16] de GaAs y de un diodo vertical de GaAs. Las simulaciones se han realizado teniendo en cuenta el método de factorización LU para resolver la ecuación de Poisson.

4.1. Algoritmo óptimo versión 1 (V1)

Previo a la elaboración del mallado óptimo, en el algoritmo antiguo el usuario tiene el control del número de mallas horizontales y verticales que desea en

cada región. La utilización de un algoritmo óptimo lleva asociado realizar cambios en el algoritmo antiguo ya que será necesario calcular para cada región la longitud de Debye de acuerdo con la Ecuación 4 para tenerla en cuenta a la hora de crear la dimensión de las mallas verticales y horizontales. Un dispositivo electrónico no tiene porqué estar formado únicamente por un material. Es más, en la definición del dispositivo electrónico puede haber involucrados dieléctricos (como es el caso de los SSDs) y/o contactos con volumen. Es por tanto necesario un criterio para definir la longitud de malla asociada a las zonas donde existan dieléctricos o contactos con volumen. En nuestro algoritmo, el tamaño de malla asociada a un dieléctrico será $\lambda_d(menor) \times 0,7$ y el tamaño asociado a una malla de contacto con volumen será $\lambda_d(menor) \times 0,6$. El usuario podrá elegir el número de nodos que quiere que intervengan en la definición del mallado, pero siempre existirá un número de nodos mínimo que garantice que se verifique la restricción de la longitud de Debye en cada celda de cada material. Este algoritmo siempre comenzará obteniendo el mallado óptimo. Si el número de nodos que desea el usuario (`nodos_user`) es mayor que el obtenido con el algoritmo óptimo, se disminuye la longitud de todas las mallas en un factor 0.95 y se vuelve a iterar hasta lograr que el número de nodos se aproxime a `nodos_user` sin sobrepasar dicho valor.

Aplicación del algoritmo óptimo (V1) en un SSD de GaAs.

El SSD [16] es una estructura nanométrica unipolar de dos terminales que, basándose en efectos electrostáticos, proporciona un efecto rectificador de la corriente debido a la asimetría que generan unas zanjas aislantes (dieléctrico) en forma de L que definen el canal conductor, Figura 4. Se simulará un SSD de

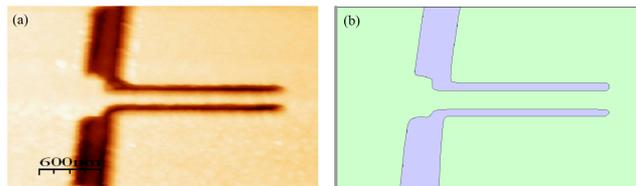


Figura 4. Self-switching diode de GaAs. (a) Imagen SEM del dispositivo, (b) definición del dispositivo mediante la interfaz Jersecin

GaAs para un dopaje de $0,35 \times 10^{23} \text{ m}^{-3}$. La permitividad del dieléctrico es $\epsilon = 1$ (aire). Las simulaciones han sido realizadas a temperatura ambiente. Se tiene en cuenta una densidad de carga superficial $\sigma_{max}/q = -0,455 \times 10^{16} \text{ m}^{-2}$.

En la Figura 5 (a) se representa el diodo discretizado con un mallado regular $MH = 50$ y $MV = 80$ mallas (4131 nodos). Dicha discretización se ha realizado por medio del algoritmo antiguo. Se observa la aproximación de la estructura

por contornos rectangulares regulares, en la cual no se obtiene una aproximación del todo fiel de la misma ya que las mallas están muy equiespaciadas y no es un mallado bueno. No se está respetando la restricción de Debye. Si se discretiza el diodo empleando el algoritmo de mallado óptimo (V1), Figura 5 (b), la aproximación por contornos rectangulares reproduce mejor el diodo. En este caso se tiene que $MH = 182$ y $MV = 178$ mallas (34547 nodos) y este mallado garantiza que sí se respete la restricción de la longitud de Debye. Si el usuario indica que

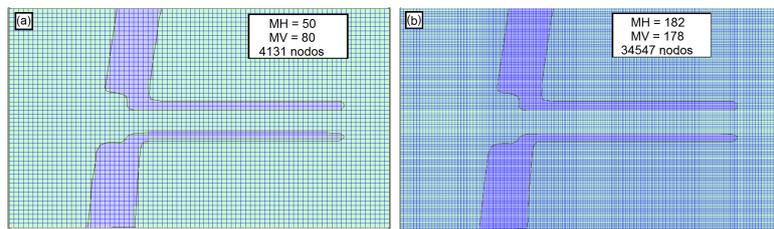


Figura 5. Self-switching diode de GaAs. (a) Pocos nodos, (b) caso óptimo.

desea realizar una discretización donde existan 53000 nodos, el algoritmo de mallado óptimo (V1) proporciona un mallado con $MH = 254$ y $MV = 205$ mallas (52530 nodos). En la Figura 6 (a) se representan las características $I-V$ tras realizar las correspondientes simulaciones Monte Carlo y el tiempo de simulación representado por diagramas de barras (Figura 6 (b)).

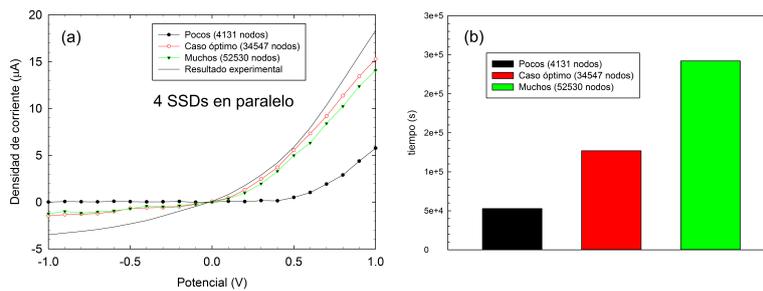


Figura 6. (a) Característica $I-V$ bajo las hipótesis de pocos nodos, caso óptimo y muchos nodos. Se representa también la curva experimental de la corriente. (b) Tiempo de simulación requerido en cada caso.

Para el diodo en estudio se dispone la característica $I - V$ experimental de 4 SSDs en paralelo [resultados que provienen de una comunicación privada con la Universidad de Manchester]. Como únicamente se simula un SSD, para que los resultados sean equiparables, se multiplica el valor de la magnitud de la densidad de corriente de la simulación por el factor 4. La simulación con el mallado con pocos nodos proporciona un resultado que no refleja el comportamiento real del diodo, siendo por tanto un resultado inaceptable y que no coincide para nada con la realidad. Si se compara el resultado de la simulación tras considerar el algoritmo óptimo (algoritmo en el que el usuario no ha tenido que indicar qué número de mallas que desea en cada región y el mallado se ha creado de forma automática) con el resultado experimental, se obtiene que en directa el acuerdo es muy bueno mientras que en inversa la discrepancia está asociada a corrientes de pérdidas a través del buffer porque el aislamiento eléctrico de la estructura fabricada no es perfecto. Con 34547 nodos se necesitó un tiempo de simulación de 1.46 días. Si el usuario desea que el mallado esté formado por aproximadamente 53000 nodos (muchos nodos), la simulación Monte Carlo proporciona prácticamente la misma característica $I - V$ que en el caso óptimo. Este mallado con muchos nodos tiene la desventaja de necesitar más tiempo de simulación, 2.8 días. Precisa prácticamente el doble de tiempo que cuando se considera el algoritmo óptimo para conseguir esencialmente el mismo resultado.

Aplicación del algoritmo óptimo (V1) en una estructura vertical de GaAs.

En este apartado se va a simular la estructura presentada en la Figura 2 bajo distintas hipótesis. Se recuerda que para que una simulación de un dispositivo electrónico sea correcta, una de las condiciones que se ha de satisfacer es que el tamaño de las mallas en cada región sea menor que la longitud de Debye ($\lambda_d^{n^+} = 4,29$ nm, $\lambda_d^{n^-} = 30,35$ nm y $\lambda_d^n = 13,57$ nm).

- Pocos nodos. El dispositivo se discretiza con un mallado homogéneo formado por $MH = 7$ mallas y $MV = 20$ mallas, 168 nodos.
 - Mallado horizontal: 7 mallas \times 7,1429 nm/malla. En las zonas n^+ no se satisface la condición de la longitud de Debye, ya que $\Delta_y > \lambda_d^{n^+}$.
 - Mallado vertical: 20 mallas \times 65 nm/malla. Para ningún caso se satisface la condición de la longitud de Debye, ya que $\Delta_x > \lambda_d^{n^+}, \lambda_d^{n^-}, \lambda_d^n$.
- Mallado óptimo. El dispositivo se discretiza con un mallado no homogéneo formado por $MH = 12$ mallas y $MV = 186$ mallas, 2431 nodos.
 - Mallado horizontal: 12 mallas \times 4,1667 nm/malla. En todas las zonas del diodo se verifica que $\Delta_y < \lambda_d^{n^+}, \lambda_d^{n^-}, \lambda_d^n$.
 - Mallado vertical
 - Zona n_1^+ : 32 mallas \times 3,125 nm/malla, $\Delta_x^{n_1^+} < \lambda_d^{n^+}$.
 - Zona n^- : 9 mallas \times 27,778 nm/malla, $\Delta_x^{n^-} < \lambda_d^{n^-}$.
 - Zona n : 49 mallas \times 13,265 nm/malla, $\Delta_x^n < \lambda_d^n$.
 - Zona n_2^+ : 96 mallas \times 3,125 nm/malla, $\Delta_x^{n_2^+} < \lambda_d^{n^+}$.

- Muchos nodos. El dispositivo se discretiza con un mallado no homogéneo formado por $MH = 17$ mallas y $MV = 274$ mallas, 4950 nodos. En este caso el usuario deseaba como máximo 5000 nodos.
 - Mallado horizontal: 17 mallas \times 2,9415 nm/malla. En todas las zonas del diodo se verifica que $\Delta_y \ll \lambda_d^{n^+}, \lambda_d^{n^-}, \lambda_d^n$.
 - Mallado vertical
 - Zona n_1^+ : 64 mallas \times 1,5625 nm/malla, $\Delta_x^{n_1^+} \ll \lambda_d^{n^+}$.
 - Zona n^- : 12 mallas \times 20,833 nm/malla, $\Delta_x^{n^-} \ll \lambda_d^{n^-}$.
 - Zona n : 69 mallas \times 9,4203 nm/malla, $\Delta_x^n \ll \lambda_d^n$.
 - Zona n_2^+ : 129 mallas \times 2,3256 nm/malla, $\Delta_x^{n_2^+} \ll \lambda_d^{n^+}$.

En la Figura 7 (a) se presenta la densidad de corriente frente al potencial aplicado entre extremos del diodo. Como se observa, la característica $I - V$ no coincide en todos los casos. La simulación realizada bajo la hipótesis del mallado con pocos nodos no está respetando la restricción impuesta por la longitud de Debye ya que el tamaño de las celdas es superior a λ_d y el dispositivo no está siendo discretizado correctamente, y por lo tanto no es una simulación precisa. Ahora bien, la simulación del diodo discretizado con el mallado óptimo ya respeta la condición de la longitud de Debye ($\Delta_x, \Delta_y < \lambda_d$) pero no proporciona el resultado correcto ya que si se aumenta el número de nodos, la característica $I - V$ cambia. Para los casos donde el número de nodos es 3480, 4304 y 4950, la característica estática de la densidad de la corriente es similar, indicativo de que sí se está reproduciendo el funcionamiento de la estructura correctamente. Para entender porqué la discretización con el mallado óptimo no reproduce el comportamiento del diodo apropiadamente, se analiza el campo eléctrico frente a la longitud del diodo, Figura 7 (b). Para el mallado óptimo en la frontera $n_1^+ - n^-$

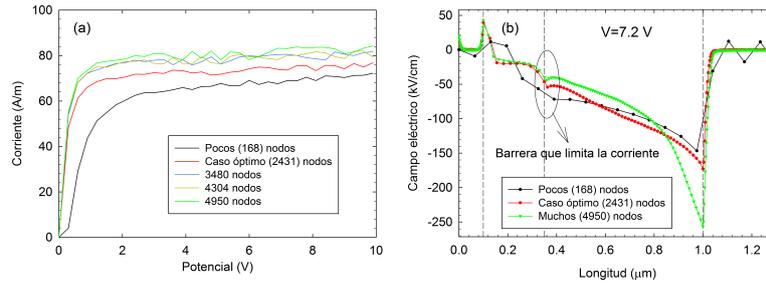


Figura 7. (a) Característica $I - V$ frente a la longitud (V1), (b) campo eléctrico medio frente a la longitud (V1). Se consideran las hipótesis: (1) mallado con pocos (968) nodos, (2) mallado óptimo (2431) nodos, (3) mallado con 3480 nodos, (4) mallado con 4304 nodos y (5) mallado con 4950 nodos

se pasa de tener mallas de $\Delta_x^{n_1^+} = 3,125$ nm a tener mallas de $\Delta_x^{n^-} = 27,778$ nm y en la frontera $n^- - n$ se pasa de tener mallas de $\Delta_x^{n^-} = 27,778$ nm a tener mallas de $\Delta_x^n = 13,265$ nm. Para la frontera $n^- - n$ el campo eléctrico no es el mismo para las tres hipótesis, Figura 7 (b). Esto es debido a que no se está resolviendo la ecuación de Poisson correctamente, y se encuentra que para el caso óptimo los electrones van a tener que superar una barrera mayor para poder acceder a la zona n en comparación con la barrera que tienen que superar en el caso de muchos nodos, donde se obtiene una densidad de corriente mayor. En la frontera $n_1^+ - n^-$ tampoco se está resolviendo la ecuación de Poisson correctamente ya que existe un salto muy brusco en el tamaño de las mallas. Un comportamiento análogo sucede en la frontera $n - n_2^+$, donde para el caso de mallado óptimo se pasa de tener mallas de $\Delta_x^n = 13,265$ nm a tener mallas de $\Delta_x^{n_2^+} = 3,125$ nm. La discretización con muchos nodos proporciona un resultado correcto pero a costa de perder tiempo introduciendo nodos en zonas donde no es necesario: $t_{3480} = 5,17$ h, $t_{4304} = 5,24$ h y $t_{4950} = 5,32$ h.

Por lo tanto, es necesario diseñar un nuevo algoritmo óptimo (C2) para evitar en las regiones frontera cambios demasiados bruscos en el tamaño de las mallas y obtener la solución correcta de la ecuación de Poisson en cualquier región del diodo.

4.2. Algoritmo óptimo versión 2 (V2)

Como se ha explicado en la última parte de la sección anterior, es necesario añadir alguna condición en el algoritmo óptimo (V1) para evitar tener saltos bruscos en los tamaños de mallas cuando haya regiones frontera. Estas condiciones se recogen en el algoritmo óptimo (V2) que será explicado con ayuda de la Figura 8.

- En primer lugar se aplica el algoritmo de mallado óptimo (V1) explicado anteriormente, ver Figura 8 (a).
- Si existe una región frontera, existirá un cambio de material y por lo tanto cambiará la longitud del tamaño de las mallas de acuerdo λ_d . La frontera limita mallas de tamaños r y s , donde $r < s$. Se pretende que no exista un cambio de malla brusco en la frontera. La malla limítrofe con la frontera de mayor tamaño, malla (1), se divide en k sub-mallas, donde $k = \text{ParteEntera}(s/r)$, y la malla vecina a ella, malla (2), se divide en t sub-mallas, donde $t = \text{ParteEntera}(s/(2r))$ (ver Figura 8 (b)).

Aplicación del algoritmo óptimo (V2) en una estructura vertical de GaAs.

Para la estructura presentada en la Figura 2 se aplica el algoritmo óptimo (V2) mediante el cual se evitarán cambios bruscos en los tamaños de las mallas en las regiones frontera además de cumplirse que el tamaño de las mallas es menor que λ_d . Se consideran las siguientes hipótesis:

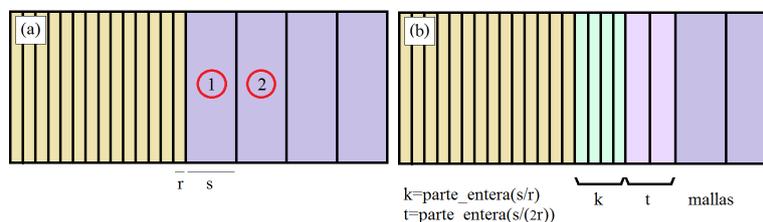


Figura 8. Diodo vertical. Discretización del dispositivo utilizando (a) el algoritmo óptimo (V1) y (b) el algoritmo óptimo (V2).

- Mallado óptimo. $MH = 12$ mallas y $MV = 200$ mallas verticales, 2613 nodos.
- Muchos nodos. $MH = 15$ mallas y $MV = 293$ mallas verticales, 4704 nodos. El usuario no desea tener más de 5000 nodos.

Para los dos tipos de mallado se observa un resultado muy similar en la característica $I - V$, resultados que no se presentan en este resumen pero sí se recogen en la memoria. Esto indica que el algoritmo de mallado óptimo (V2) está discretizando adecuadamente el dispositivo y la resolución de la ecuación de Poisson se resuelve correctamente en las regiones frontera. Para el mallado óptimo, en la transición $n_1^+ - n^-$ se pasa de tener mallas de $\Delta_x^{n_1^+} = 3,125$ nm a tener mallas de $\Delta_x^{n^-} = 3,0864$ nm, en la región $n^- - n$ se pasa de tener mallas de $\Delta_x^{n^-} = 13,89$ nm a tener mallas de $\Delta_x^n = 13,265$ nm y en la frontera $n - n_2^+$ el cambio de tamaño en las mallas es de $\Delta_x^n = 2,5390625$ nm a $\Delta_x^{n_2^+} = 2,3255$ nm. Esto indica que el mallado óptimo es bueno y sí se está resolviendo la ecuación de Poisson correctamente en las fronteras. La ventaja de considerar el mallado óptimo está en el tiempo de simulación, 5,08 h, frente al tiempo que tarda la simulación MC con el mallado con muchos nodos, 5,27 h. Cuando se consideran muchos nodos el diodo se discretiza de forma muy precisa en todas las regiones, lo que conlleva un aumento del tiempo de CPU. Es importante evitar cambios bruscos en los tamaños de las mallas en las fronteras, y en el resto de las regiones únicamente es necesario que el tamaño de las mallas sea menor que la longitud de Debye.

5. Conclusiones

Para finalizar, se recogen aquí las principales conclusiones obtenidas:

1. Tras profundizar en el estudio de la resolución de la ecuación de Poisson utilizando distintos métodos matemáticos se concluye que con la técnica de factorización LU se resuelve el problema de forma exacta precisándose menor

tiempo de simulación en comparación con las técnicas de Jacobi, Gauss-Seidel y sobre-relajación sucesiva (SOR). Es la técnica SOR la que compite con la factorización LU en tiempo de cálculo.

2. En las técnicas de Jacobi y Gauss-Seidel, cuando se ha utilizado el criterio de convergencia (C1) en el estudio del diodo vertical de GaAs ($MH = 2$ mallas y $MV = 20$ mallas, 63 nodos), para el caso donde se desea reducir el error en un factor 10^{-p} , basta considerar $p = 1$ para encontrar un resultado correcto de la simulación del dispositivo. Para la misma estructura se ha comprobado que el criterio de convergencia (C1) en el método SOR no proporciona buen resultado para $p = 1$ y $p = 2$.
3. Para el mismo criterio de convergencia (C1), si el dispositivo se discretiza con más mallas (caso en el que la matriz cuadrada es de orden superior a 84) el método de Jacobi y el de Gauss-Seidel proporcionan tiempos de simulación muy elevados y por lo tanto, su uso no es recomendable.
4. Si se tiene en cuenta el criterio de convergencia (C2) para un diodo vertical de GaAs ($MH = 5$ mallas y $MV = 20$ mallas, 126 nodos) para una tolerancia $TOL = 10^{-5}$, el tiempo de simulación se reduce en un $\sim 80\%$ para los métodos de Jacobi y Gauss-Seidel respecto al criterio (C1). Por el contrario, emplear el método SOR únicamente reduce en un $\sim 1,5\%$ el tiempo de simulación.
5. Tras investigar distintas formas de la creación del mallado se concluye que el algoritmo óptimo de mallado (V1) proporciona una discretización correcta en los SSDs donde únicamente en la definición del dispositivo electrónico interviene un material y un dieléctrico.
6. Si existen regiones que limitan diferentes materiales o un mismo material con diferentes impurificaciones, la restricción impuesta por la longitud de Debye no es suficiente. Es necesario evitar en las fronteras cambios bruscos en los tamaños de las mallas. Esto se ha tenido en cuenta en el algoritmo óptimo de mallado (V2). En las regiones donde no existe una frontera, considerar únicamente la restricción impuesta por la longitud de Debye es suficiente y tiene la ventaja de que se precise un tiempo menor en la simulación MC.
7. Se ha simulado un diodo vertical con los dos algoritmos óptimos del mallado, (V1) y (V2). Bajo estas hipótesis se demuestra que el algoritmo óptimo (V1) no reproduce de forma fiel el funcionamiento del diodo a menos que se discretice con muchos nodos. Esto evidentemente tiene una penalización en el tiempo de simulación. Por el contrario, si se emplea el algoritmo óptimo (V2) se observa que con la discretización óptima se reproduce de forma correcta el funcionamiento del diodo mejorándose en un $\sim 2\%$ el tiempo de simulación en comparación con la simulación empleando el algoritmo (V1) para 3480 nodos. Además de reducir el tiempo de cálculo se tendrá más precisión en las zonas frontera, aspecto que puede ser clave cuando se estudian fenómenos dinámicos.
8. Este trabajo me ha permitido comprender el funcionamiento del simulador Monte Carlo y me sirve de punto de partida para la elaboración de un simulador donde se incorpore una resolución quasi-3D de la ecuación de Poisson.

Además, se incorporará la ecuación de calor en el código Monte Carlo empleando el método SOR por la relativa sencillez de su implementación.

Bibliografía

- [1] C. Jacoboni and P. Lugli, *The Monte Carlo method for semiconductor device simulation*. Springer-Verlag, New York, 1989.
- [2] M. Cerrolaza, *El método de los elementos finitos para ingeniería y ciencias aplicadas: teoría y programas*. Universidad Central de Venezuela. Consejo de Desarrollo Científico y Humanístico, Venezuela, 2007.
- [3] A. Carnicero, *Introducción al Método de los Elementos Finitos*. Escuela Técnica Superior de Ingeniería ICAI. Universidad Pontificia Comillas, Madrid, 2001.
- [4] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical recipes. The Art of Scientific Computing*. Cambridge University Press, New York, 1989.
- [5] A. D. Boardman, *Computer simulation of hot electron behavior in semiconductors using Monte Carlo methods, in Physics Programs*. Physics Programs, Edited by A. D. Boardman, New York, 1980.
- [6] T. Kurosawa, "Proceedings of the international conference on the physics of semiconductors," *J. Phys. Soc. Japan Suppl*, vol. 49, p. 345, 1966.
- [7] T. González, "Análisis del ruido electrónico en materiales y dispositivos semiconductores unipolares mediante el método de monte carlo," tech. rep., Tesis Doctoral, (Universidad de Salamanca), Salamanca, 1994.
- [8] H. Rodilla, "Estudio de hemts basados en semiconductores de gap estrecho. desde los materiales al dispositivo," tech. rep., Tesis Doctoral, (Universidad de Salamanca), Salamanca, 2010.
- [9] A. Íñiguez-de-la-Torre, "Nanodispositivos de gan para generación de señales de thz. simulación monte carlo y experimental," tech. rep., Tesis Doctoral, (Universidad de Salamanca), Salamanca, 2012.
- [10] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles*. IOP, Bristol, 1988.
- [11] S. Tiwari, *Compound semiconductor device physics*. Academic Press, New York, 1992.
- [12] J. Mateos, "Modelización microscópica del ruido electrónico en estructuras mesfet y hemt submicrométricas. análisis numérico y experimental," tech. rep., Tesis Doctoral, (Universidad de Salamanca), Salamanca, 1997.
- [13] J. L. De la Fuente, *Técnicas de cálculo para Sistemas de Ecuaciones, Programación lineal y Programación Enteran*. Editorial Reverté, S.A., Barcelona, 1998.
- [14] D. Vasileska, *Numerical Solution of Poisson's Equation*. nanoHUB.org, online simulations and more, 2006.
- [15] S. García, "Desarrollo de una interfaz gráfica para la definición de la topología de dispositivos electrónicos en un simulador monte carlo," tech. rep., Proyecto fin de carrera, ITIS (Universidad de Salamanca), Salamanca, 2011.

- [16] J. Mateos, B. G. Vasallo, D. Pardo, T. González, “Operation and high frequency performance of nanoscale unipolar rectifying diodes,” *Appl. Phys. Lett.*, vol. 86, p. 212103, 2005.

Sistema Multiagente desplegado en un entorno Cloud Computing para gestión de zonas de aparcamiento

Daniel Hernández Alfageme¹ y Juan Manuel Corchado¹

¹Departamento de Informática y Automática. Universidad de Salamanca. Plaza de los Caídos s/n. 37008 Salamanca, España
{dan_her_alf, corchado}@usal.es

Resumen En este trabajo se realiza el diseño e implementación de un Sistema Multi-Agente integrado dentro de una plataforma de *Cloud Computing* para la gestión de zonas de aparcamiento en núcleos urbanos. El objetivo es facilitar la labor de vigilantes, conductores y gestores, mejorando los mecanismos existentes y tratando de solventar las carencias actuales en este ámbito. Para ello, se presenta el estado del arte en el área de Sistemas Multi-Agente y Organizaciones Virtuales de agentes, Cloud Computing y algoritmos de recorrido de grafos y de planificación. Como resultado se ha obtenido un Sistema Multiagente abierto desarrollado con la plataforma PANGEA y se ha diseñado un algoritmo específico para la creación de rutas óptimas de vigilantes.

Keywords: Agente Inteligente, Sistema Multi-Agente, Organización Virtual, Cloud Computing, Grafo Euleriano, Chinese Postman Problem

1. Introducción

Actualmente, la búsqueda de aparcamiento en el centro de las ciudades es una tarea difícil dada la creciente cantidad de vehículos existentes para un espacio que se mantiene constante, o incluso que se va reduciendo. Además, estos espacios se encuentran regulados por zonas de aparcamiento.

En este proyecto se plantea la creación de un sistema que facilite a los conductores la gestión del pago en estas zonas de aparcamiento, permitiéndoles ahorrar tiempo y dinero. Tiempo, a la hora de tener que localizar un expendedor y desplazarse hasta él o en el caso de tener que regresar para aumentar el tiempo de validez del ticket. Y dinero, a la hora de definir el tiempo de aparcamiento. Con el modelo de aplicación propuesto el usuario no necesita indicar la cantidad de tiempo que va a estar aparcado su coche, permitiéndole ahorrar dinero tanto en estimaciones de tiempo por exceso (tiempo de aparcamiento pagado y no utilizado) como por defecto (posibilidad de sanción).

La creación del sistema se plantea mediante la construcción de dos partes diferenciadas, ambas basadas en el diseño de una organización virtual de agentes. Por un lado, se encuentra la parte del servidor, compuesta por una aplicación

de administración de las zonas de aparcamiento regulado en una ciudad y una serie de servicios creados para el uso por parte de las aplicaciones clientes. Por otro lado, se crearán dos aplicaciones móviles (que se escapan del estudio de este trabajo) para el vigilante de las zonas de aparcamiento y para el conductor, respectivamente. Este trabajo se centra en la creación de la parte del servidor.

Actualmente, la tecnología móvil está a la orden del día. Tal es así que desde hace dos años la venta mundial de smartphones supera a la de los ordenadores [1]. Personas de todas las generaciones hacen uso en su vida cotidiana de dispositivos móviles inteligentes. Por otra parte, el término *Cloud Computing* hace referencia a un revolucionario paradigma computacional que permite ofrecer servicios software y hardware de forma distribuida y ubicua a través de Internet [2]. En este paradigma todo lo que puede ofrecer un sistema informático tradicional se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles “en la nube de Internet” sin necesidad de conocimientos previos o de instalación de software en sus ordenadores.

Como consecuencia de la introducción del *Cloud Computing* y las tecnologías móviles, el software tradicional no es capaz de adecuarse perfectamente a este tipo de sistemas, siendo necesario introducir nuevos enfoques que se adapten mejor a sistemas tan dinámicos y heterogéneos. En este sentido, la evolución de los **agentes inteligentes** continúa, adaptándose constantemente a las tecnologías emergentes en cada momento.

Las organizaciones virtuales de agentes [3] permiten modelar aplicaciones desde un punto de vista sociológico muy acorde al problema de gestión de aparcamientos que se pretende resolver, dado su carácter heterogéneo, en el que entidades muy diversas pasan a formar parte del sistema. Cada vez es más común encontrar agentes en áreas relacionadas con la Web, como Web Semántica [4] o Servicios Web [5] utilizando protocolos REST (*Representational State Transfer*). En concreto, los servicios REST proporcionan una arquitectura idónea para el intercambio de información y solicitudes [6].

Una vez introducido el problema a resolver, en la sección **2. Objetivos** se recogerán los objetivos de la investigación. En el apartado **3. Marco teórico** se analizan los principales aspectos teóricos y se realiza un análisis básico del estado del arte en agentes y SMA, Cloud Computing y algoritmos de recorrido de grafos y de planificación. En la sección **4. Propuesta del sistema** se detallan la arquitectura de SMA construida y la metodología y plataforma de desarrollo empleadas, así como el algoritmo de generación de rutas que se ha diseñado. Finalmente, en la sección **5. Conclusiones** se analizan los resultados obtenidos y se proponen algunas líneas futuras de investigación.

2. Objetivos

Los objetivos de este estudio proponen contribuir al ahorro de tiempo y costes, así como a una mejora indirecta de la calidad de vida a diversos colectivos relacionados con el ámbito de las zonas de aparcamiento (conductores, vigilantes, ayuntamientos, gestores). Para ello, se realizará una investigación en los méto-

dos, herramientas y tecnologías existentes en el estado del arte para proponer una solución óptima que permita solucionar, o mejorar en todo lo posible, los problemas y dificultades existentes actualmente en el campo de los aparcamientos en ciudades.

Con este propósito, se plantean los objetivos enumerados a continuación. Realizar un análisis del estado del arte en **Agentes Inteligentes**, revisando los conceptos y principales metodologías en el paradigma de agentes, sistemas multiagente y organizaciones virtuales. Investigar las tendencias actuales en **Cloud Computing** en el entorno empresarial y académico y los productos y tecnologías asociadas. Analizar las soluciones actualmente existentes para la generación de rutas óptimas en grafos, con el fin de poder aportar una solución específica para la **generación de rutas óptimas de vigilancia** en el contexto de este trabajo. Se diseñará e implementará una **arquitectura multiagente** basada en organizaciones virtuales que permita facilitar la gestión de las zonas urbanas de aparcamiento, ofreciendo también una serie de servicios tipo REST a aplicaciones móviles para conductores y vigilantes. Se proporcionarán **mecanismos de visualización** para representar aspectos de la gestión tales como seguimiento de rutas para vigilantes o zonas de aparcamiento. Finalmente, el sistema construido se integrará dentro de una **plataforma de Cloud Computing** existente, haciendo uso de los servicios ofrecidos por esta tecnología.

3. Marco teórico

3.1. Agentes, Sistemas Multi-Agente y Organizaciones Virtuales

Los agentes han sido empleado en numerosas áreas de la ciencia de la computación desde la segunda mitad del siglo XX (Inteligencia Artificial, Ingeniería del Software, Sistemas Distribuidos, Bases de Datos, etc.). Una de las primeras definiciones del término “**agente**” se debe a [7]: *Sistema computacional basado en hardware o (más habitualmente) basado en software que disfruta de las propiedades de autonomía, habilidades sociales, reactividad y proactividad..* Para unificar todas las definiciones existentes, se ha decidido definir el término agente como una entidad que posee las siguientes características: Autonomía, Situación, Reactividad, Pro-Actividad o Racionalidad, Habilidad social, Inteligencia, Organización, Movilidad y Aprendizaje.

Los agentes pueden colaborar entre ellos. En este momento surge el concepto de **Sistema Multiagente** (SMA), definido como “*la unión de dos o más agentes colaborando en su trabajo con el objetivo de resolver un problema*” [8]. Los sistemas multiagente poseen una computación asíncrona dado su carácter distribuido (descentralizado). No existe un control global, por lo que cada agente se centra en su conducta individual. Cada agente decide libre y dinámicamente, utilizando su autonomía y pro-actividad qué tareas debe efectuar y a quién asigna estas tareas [9], definiendo metas y planes que le permitan alcanzar sus objetivos.

Por otra parte, la analogía entre sociedades de agentes y sociedades humanas es clara. Por ello, aparece el concepto de **Organización Virtual** (OV), definido como “*un conjunto de individuos e instituciones que necesitan coordinar*

sus recursos y servicios dentro de unos límites institucionales". Una OV es un sistema abierto y eminentemente heterogéneo, formado por la agrupación y la colaboración de entidades donde existe una clara separación entre la forma y la función que define su comportamiento. En la actualidad, existe un amplio abanico de investigaciones enfocadas a la construcción de organizaciones virtuales, a la simulación social basada en agentes y al estudio de su comportamiento [10, 11].

3.2. Cloud Computing

Como ya se ha dicho, el *Cloud Computing* es una tecnología que permite ofrecer servicios a través de Internet. El NIST (*National Institute of Standards and Technology*), define este término como un “*modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio*” [12]. Los servicios que proporciona una plataforma de Cloud Computing pueden ser enmarcados en tres capas diferentes. Estas capas son:

- **Software como servicio (SaaS).** Constituye la capa más alta de los sistemas *Cloud*, y consiste en una serie de aplicaciones ejecutadas bajo demanda.
- **Plataforma como servicio (PaaS).** Se compone de un conjunto de herramientas o servicios de producción y desarrollo tales como Interfaces de Programación de Aplicaciones (API's) para las aplicaciones de la capa *SaaS*.
- **Infraestructura como Servicio (IaaS) o Hardware como Servicio:** comprende todos los servidores (máquinas físicas y virtuales), sistemas de almacenamiento, conexiones, dispositivos de red, etc.

Pilares Tecnológicos. La base tecnológica que ha permitido el nacimiento y expansión de la Tecnología Cloud se compone de los pilares fundamentales descritos a continuación:

- *Virtualización:* en informática, la virtualización se refiere a la abstracción de la estructura física subyacente de elementos como plataformas hardware, sistemas operativos, recursos de red o dispositivos de almacenamiento [13]. Supone la base para la migración de máquinas y la consecución de la elasticidad en los sistemas Cloud. Un ejemplo de sistema de virtualización es KVM [14].
- *Bases de datos distribuidas:* una base de datos distribuida es una colección de múltiples bases de datos relacionadas sobre una red de computadores. Las bases de datos distribuidas permiten que las plataformas cloud sean escalables y aportan seguridad a la información (replicación, partición y distribución física de la información, etc.).

- *Sistemas de ficheros distribuidos*: un sistema de ficheros distribuido es un sistema de ficheros que permite el acceso a ficheros desde múltiples hosts a través de una red [15], como si los ficheros se encontraran en la propia máquina del usuario. Los requisitos básicos que debe cumplir un sistema de ficheros distribuido son transparencia de localización, eficiencia, tolerancia a fallos, escalabilidad, consistencia, mecanismo de actualizaciones concurrentes y seguridad [16].
- *Balanceo de carga*: los balanceadores de carga funcionan como un proxy, en el sentido de que suponen un punto de acceso centralizado a los servicios para los usuarios y de que modifican las peticiones entrantes en el servidor distribuyéndolas entre un conjunto de proveedores de servicios con el objeto de evitar situaciones de sobrecarga. Para efectuar esta tarea hacen uso de algoritmos de distribución tipo *Round-Robin* o similares. Hay tres algoritmos importantes [17] para gestionar el balanceo de carga: el algoritmo *Honeybee Foraging* [18], *Biased Random Sampling* [19], y *Active Clustering* [20].

3.3. Algoritmos de Planificación y Recorrido de Grafos

Algoritmo de Fleury. El algoritmo de *Fleury* es el método más conocido para la obtención de ciclos y/o caminos eulerianos. El algoritmo de Fleury se aplica únicamente sobre grafos eulerianos (aquel que contiene algún camino o ciclo euleriano), y permite la obtención de un ciclo (camino) euleriano siguiendo el siguiente procedimiento:

1. Construir una lista vacía C que contendrá el ciclo construido.
2. Tomar un vértice cualquiera x y añadirlo a la lista C .
3. Tomar una arista a cualquiera incidente con $x \rightarrow y$, siempre que el grafo resultante de retirar la arista continúe siendo conexo (exceptuando al vértice x , que podría quedar inconexo si no quedan más aristas por recorrer incidentes con él).
4. Retirar a del grafo, y retirar también el nodo x si tras eliminar la arista a el nodo x resulta inconexo (tiene grado 0, i.e. no hay más aristas incidentes). Añadir el nodo y al final de la lista C .
5. Si se han recorrido todas las aristas (el grafo ha quedado vacío) devolver la lista C que contendrá el ciclo.
6. Si no, tomar $x = y$ y volver al paso 3.

El algoritmo anterior se puede aplicar en grafos no eulerianos que posean únicamente dos nodos con grado impar para obtener un camino euleriano abierto si se toma como vértice inicial x uno de los dos nodos con grado impar (el camino finalizará en el otro nodo con grado impar).

Problema del Cartero Chino. El principal problema del algoritmo de Fleury es que solo se puede aplicar a grafos eulerianos (todos los nodos tienen grado par), o a lo sumo que tengan dos nodos con grado impar. Para ello, en [21] se ideó una implementación que resolvía el problema de encontrar el ciclo de peso

mínimo que contiene todas las aristas de un grafo conexo, repitiendo únicamente aquellas aristas que sea preciso. Este problema se conoce como Problema del cartero chino.

Existen diferentes implementaciones que tratan de solucionar este problema. En [22] se propone una solución basada en algoritmos genéticos. Por otra parte, en [23] se plantea otra solución basada en redes neuronales estocásticas. En [24] se propone la solución de este problema utilizando para ello computación molecular. En [25] idearon una solución para un problema multi-objetivo donde cada eje del grafo puede tener varios valores para los pesos.

El algoritmo requiere emparejar los vértices de grado impar, de tal modo que la suma de los pesos (distancias) asociados a las parejas formadas sea mínimo. Dicho problema es un problema NP-hard o NP-complejo [26] (es decir, actualmente no se conocen algoritmos que lo resuelvan en tiempo polinomial) y se conoce como problema de emparejamiento mínimo (*Minimum Weight Matching*, *MWM*). Existen diversos algoritmos para lograr solucionar este problema.

El algoritmo *Blossom V* [27] se emplea para resolver el problema de *Minimum Weight Matching* utilizando colas de prioridad y teoría del espacio dual. Para resolver el problema utilizan teoría de optimización y programación lineal.

Algoritmo A*. Un algoritmo A* [28] encuentra la solución óptima (camino de coste mínimo), bajo ciertas condiciones, a un problema de planificación. Se trata de un algoritmo de búsqueda informado, pues utiliza una heurística para hallar la solución óptima. El problema de algunos algoritmos que utilizan heurística es que no consideran el coste real que tiene una determinada operación, llegando muchas veces a soluciones no óptimas.

El algoritmo A* tiene en cuenta el coste real del camino recorrido hasta un cierto punto unido a un valor aproximado del coste asociado al resto de pasos en planificación hasta la meta deseada (por un agente, en este caso). La función de evaluación utilizada por un algoritmo A* es de la forma $f(n) = g(n) + h'(n)$. $f(n)$ contiene el coste aproximado de ejecutar un plan desde una situación inicial a una situación final pasando por una situación intermedia n . $g(n)$ se refiere al coste real de ejecución del plan desde la situación inicial hasta la situación n , mientras que $h(n)$ es el coste de ejecución del plan desde la situación actual n hasta la situación objetivo. $h'(n)$ contiene la heurística, es decir, una estimación de $h(n)$. Para que el algoritmo A* encuentre la solución óptima, se debe cumplir que, en cualquier caso $h'(n) \leq h(n)$; se dice en este caso que la heurística es admisible. En otro caso, la solución encontrada podría no ser óptima, aunque el algoritmo convergería mas pronto.

El algoritmo A* es un algoritmo completo, es decir, encuentra la solución óptima siempre que exista y que la heurística sea admisible.

La complejidad espacio-temporal de un algoritmo A* es exponencial, por lo que en problemas muy grandes puede no resultar eficiente, aunque si la heurística es suficientemente buena los resultados pueden ser aceptables.

4. Propuesta del Sistema

4.1. Plataforma PANGEA

Para diseñar e implementar el sistema se ha utilizado PANGEA [29], un framework para el desarrollo de Sistemas Multiagente. PANGEA, *Platform for Automatic coNstruction of orGanizations of intElligent Agents* es una plataforma orientada a servicios que permite la implementación de Sistemas Multi-Agente abiertos mediante Organizaciones Virtuales de agentes para obtener el máximo aprovechamiento de la distribución de los recursos. Para lograrlo, todos los servicios son implementados como Servicios Web. Debido a esta orientación a servicios, diferentes herramientas modeladas con agentes que consumen Servicios Web pueden ser integradas desde la plataforma, independientemente de su localización física y de su implementación. La plataforma está especialmente diseñada para la aplicación práctica de los conceptos de organización, normas y roles.

4.2. Metodología GORMAS

Para el diseño del Sistema Multiagente se ha hecho uso de la metodología GORMAS [30]. GORMAS es una guía metodológica para el diseño de SMA abiertos desde la perspectiva de las organizaciones humanas. El análisis y diseño de mecanismos de coordinación y colaboración en sistemas abiertos se ve facilitado gracias al concepto de organización. GORMAS consta de un conjunto de fases que cubren el análisis, el diseño de la estructura organizativa y el diseño de la dinámica de la OV. Estas fases permiten especificar los servicios ofrecidos por la organización, su estructura interna y las normas que determinan el comportamiento.

El modelo GORMAS establece la funcionalidad como sistema un abierto, que incluye tanto los servicios que se deben publicitar como las políticas de adquisición y liberación de roles. Se determina qué funcionalidad debe ser implementada por agentes internos y cuál se publicita para poder ser suministrada por agentes externos. De este modo, se modela la dinamicidad de la organización como sistema abierto. En la Fig. 1 se muestra la topología del sistema.

Los roles que componen la OV se describen brevemente a continuación:

- **Usuario.** Representa el usuario del sistema. En este sentido, puede tratarse de un conductor o de un gestor.
- **Manager Zonas.** Es el rol encargado de definir y servir información (propiedades) sobre las diferentes zonas de aparcamiento.
- **Almacenamiento.** Este rol se encarga de realizar la integración de la información persistente con una plataforma de Cloud Computing.
- **Notificador.** Es el rol encargado de enviar diferente tipo de notificaciones asíncronas al rol usuario (sanciones, alarmas, etc.).
- **Pago.** Se trata del rol encargado de realizar la recarga de saldo para los conductores.

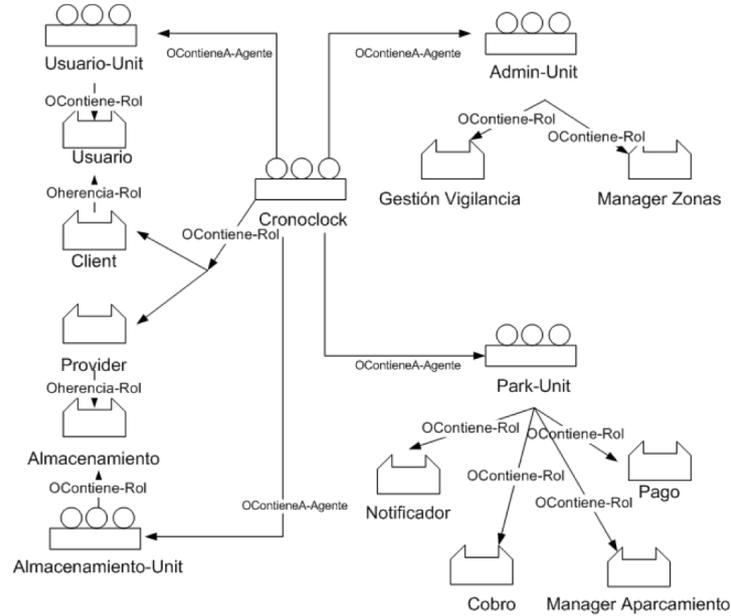


Figura 1. Modelo de estructura. Topología de tipo holarquía

- **Manager aparcamiento.** Este rol tiene un control específico sobre los aparcamientos realizados en un instante, sus ubicaciones, tiempos de estancia.
- **Cobro.** Este rol se encarga de realizar, periódicamente, el cobro de los aparcamientos
- **Gestión Vigilancia.** Este rol se encarga de realiza las labores de gestión de vigilancia (gestion de rutas, localización).

4.3. Optimización de Rutas

En base a la investigación realizada para resolver el problema del cálculo de rutas óptimas, se planteó la resolución de encontrar el ciclo de longitud mínima que recorre todas las aristas de un grafo siguiendo el procedimiento que se detalla a continuación.

1. Obtener los vértices de grado impar del grafo.
2. Si la lista de vértices de grado impar está vacía (el grafo es euleriano), ir al paso 4.
3. Sea I la lista de nodos de grado impar del grafo. Construir un grafo euleriano duplicando aristas. Para ello será necesario:
 - a) Aplicar el *algoritmo de Dijkstra* para obtener los caminos de longitud mínima entre cada par de nodos de I . Construir una matriz con dichos pesos.

- b) Resolver el problema de Minimum Weight Matching con los nodos de la lista I .
 - c) Para cada par de nodos (i, j) devueltos por el algoritmo de emparejamiento mínimo, duplicar en el grafo las aristas que componen el camino mínimo de i a j obtenido aplicando el *algoritmo de Dijkstra*.
4. Aplicar el algoritmo de Fleury al grafo euleriano.

De todas las operaciones anteriores, falta todavía por aclarar cuál es el método de resolución del problema de MWM. Para ello, en lugar de utilizar uno de los algoritmos existentes (*Blossom V*), se propone crear una solución propia, que supone uno de los principales puntos de investigación de este trabajo.

En la Fig. 2 se puede ver un diagrama de flujo del procedimiento a seguir con la solución aportada (para el subproblema de MWM). Este procedimiento corresponde al paso 2 del algoritmo global presentando unas líneas atrás.

La secuencia de pasos a seguir se describe a continuación. En primer lugar, se ha diseñado un algoritmo A* para la resolución del problema de emparejamiento mínimo. Las diferentes operaciones realizadas en el plan consisten en la realización de parejas de nodos. Este algoritmo utiliza la matriz de pesos para construir la función heurística $h'(x)$ sobre la que se construye el algoritmo, detallada en el Algoritmo 1.

Algoritmo 1 Función Heurística para MWM

Input: Lista de tamaño par de nodos a emparejar L

Input: Matriz bidimensional de pesos W para los nodos de L

Output: Una heurística admisible

$N \leftarrow \text{size}(L)$

$h \leftarrow 0$

for $i = 1 \rightarrow N/2$ **do**

$m \leftarrow$ Valor mínimo en W

$a \leftarrow$ Fila del valor m

$total \leftarrow total + m$

$W \leftarrow$ Resultado de eliminar la fila a de la matriz W

end for

return $total$

El algoritmo proporciona un resultado adecuado, pero ineficiente cuando el tamaño del problema crece. Por ello, se plantea la realización de un clustering de nodos cuando el tamaño del problema supere un tamaño máximo permitido, utilizando las coordenadas geográficas de dichos nodos. El clustering es una técnica no supervisada de minería de datos que ayuda a dividir un conjunto de datos en grupos de acuerdo a las características intrínsecas de éstos. Para realizar el clustering se utiliza el algoritmo *K-medias*, normalizando previamente la latitud y la longitud de cada punto. Una vez obtenidos los cluster, es posible aplicar a cada uno el algoritmo A* diseñado. El procedimiento de clustering se

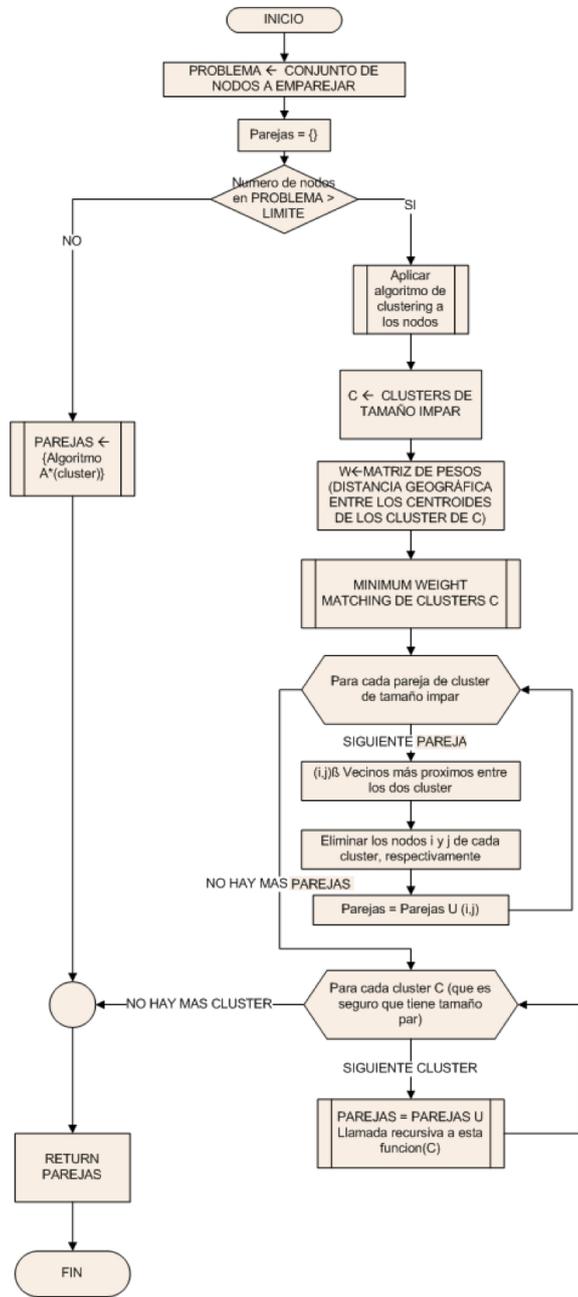


Figura 2. Solución ideada para resolver el problema MWM

aplica de manera recursiva mientras el tamaño de un cluster supere el tamaño máximo permitido.

Existe todavía un problema, y es que las técnicas de clustering forman grupos de tamaño arbitrario. El algoritmo A* y el problema MWM en general, requieren como entrada un número de nodos par. Existen algunas soluciones basadas en programación lineal para obtener clustering con restricciones de tamaño en los cluster [31]. La solución propuesta aquí consiste en resolver el problema de MWM para emparejar los cluster de tamaño impar, de forma análoga al MWM de los puntos. En este caso se utilizan los centroides de cada cluster para determinar las distancias (pesos) entre estos. Una vez emparejados los cluster, se extrae a cada pareja de cluster la pareja más cercana de nodos entre ellos (uno de cada cluster). De este modo, los clusters ya tienen tamaño par y se les puede aplicar el algoritmo A*.

La solución final al problema de MWM viene dada por todas las parejas formadas en cada cluster considerado, unidas a las parejas formadas al extraer nodos de los cluster de tamaño impar.

5. Conclusiones

La parte central de la investigación realizada, además del diseño y desarrollo de un SMA integrado en una plataforma Cloud y haciendo uso de técnicas de visualización, ha sido la creación de un algoritmo específico de optimización de rutas para el problema en cuestión.

El algoritmo A* diseñado inicialmente para la resolución del problema de emparejamiento mínimo (MWM) tiene la problemática de resultar ineficiente para la solución de problemas de tamaño medio/elevado. Por ello, se tomó la decisión de aplicar algoritmos de minería de datos, como es el clustering *K-medias* para simplificar el problema en subproblemas (divide y vencerás). Al aplicar el procedimiento de clustering y ser necesario obtener clusters de tamaño par, se optó por aplicar recursivamente el algoritmo A* de resolución del problema MWM (pero con una semántica diferente) a los cluster de tamaño impar.

Los resultados obtenidos muestran que es posible reducir una ejecución inicial de horas de duración a apenas unos segundos aplicando el algoritmo diseñado. Las técnicas empleadas son heurísticas y no garantizan una solución óptima, pero sí aportan una solución eficiente, y muchas veces óptima a un problema *NP-hard* para el que no existe una solución algorítmica.

5.1. Líneas futuras de investigación

A continuación se enumeran algunas posibles líneas de investigación futuras para el presente trabajo:

- Creación de algún método no heurístico que de una solución siempre óptima al problema de la ruta mínima para vigilantes, que mejore los algoritmos actualmente existentes. Cabe decir que los resultados obtenidos, a pesar de ser heurísticos, son bastante satisfactorios.

- Creación de rutas de vigilancia en base a otros parámetros no considerados como nivel de ocupación de zonas en diversas franjas temporales o en función del nivel de sanciones en cada zona, optimizando así la labor y efectividad del vigilante. Un enfoque posible sería tener en cuenta valores estadísticos sobre estos parámetros para obtener un peso en las aristas del grafo que no dependa únicamente de las distancias entre nodos.
- Creación de técnicas de visualización interactivas para los datos mencionados anteriormente (nivel de sanciones en función de zonas, horas o días de la semana, ocupación de zonas, etc).
- Información sobre plazas de aparcamiento libres.
- Búsqueda de parkings públicos.
- Integración con otro tipo de sistemas como recomendación turística o acceso a diferentes servicios de la ciudad (emergencias, restauración, ocio), etc.

Bibliografía

- [1] J. Edmondson, C. Mellon, W. Anderson, J. Loyall, J. Gray, J. White, and K. Schmid, "Special issue on next generation mobile computing," 2013.
- [2] K. Egami, S. Matsumoto, and M. Nakamura, "Ubiquitous cloud: Managing service resources for adaptive ubiquitous computing," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 123–128, 2011.
- [3] J. Ferber, O. Gutknecht, and F. Michel, "From agents to organizations: An organizational view of multi-agent systems," in *In: LNCS n. 2935: Procs. of AOSE'03*, pp. 214–230, Springer Verlag, 2003.
- [4] T. Berners-lee, J. Hendler, and O. Lassila, "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities," 2001.
- [5] J. Samper, "Ontologias para servicios web semanticos de informacion de trafico: descripcion y herramientas de explotacion," 2005.
- [6] R. Khare and R. Taylor, "Extending the representational state transfer (rest) architectural style for decentralized systems," in *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, pp. 428–437, 2004.
- [7] N. Jennings and M. Wooldridge, "Technology:foundations, applications and markets," 1995.
- [8] A. Mas, "Agentes software y sistemas multiagente : conceptos, arquitecturas y aplicaciones," ISBN 84-205-4367-5, pp. 29–64, 2005.
- [9] M. Wooldridge, "An introduction to multiagent systems," 2002.
- [10] C. Sandores and J. Pavon, "Simulacion social basada en agentes.," *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, pp. 71–78, 2005.
- [11] J. Iglesias, "Modelado automático del comportamiento de agentes inteligentes," *Departamento de Informática. Universidad Carlos III de Madrid*, 2010.
- [12] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.
- [13] M. Eisen, "Introduction to virtualization. the long island chapter of the ieee," 2011.
- [14] L. YamunaDevi, P. Aruna, D. Sudha, and N. Priya, "Security in virtual machine live migration for kvm," in *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pp. 1–6, 2011.
- [15] G. Silberschatz, *Operating System concepts, chapter 17 Distributed file systems*. 1994.
- [16] J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. West, "Scale and performance in a distributed file system," *ACM Trans. Comput. Syst.*, vol. 6, pp. 51–81, Feb. 1988.
- [17] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Advanced*

- Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pp. 551–556, 2010.
- [18] S. Nakrani and C. Tovey, “On honey bees and dynamic server allocation in internet hosting centers,” *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, vol. 12, pp. 223–240, Sept. 2004.
 - [19] O. A. Rahmeh, P. Johnson, and A. Taleb-bendiab, “A dynamic biased random sampling scheme for scalable and reliable grid networks,” 2008.
 - [20] F. Saffre, R. Tateson, J. Halloy, M. Shackleton, and J. L. Deneubourg, “Aggregation dynamics in overlay networks and their implications for self-organized distributed applications,” *Comput. J.*, vol. 52, pp. 397–412, July 2009.
 - [21] J. Edmonds and E. Johnson, “Matching, euler tours and the chinese postman,” *Mathematical Programming, Vol. 5*, pp. 88–124, 1973.
 - [22] H. Jiang, L. Kang, S. Zhang, and F. Zhu, “Genetic algorithm for mixed chinese postman problem,” in *Advances in Computation and Intelligence* (Z. Cai, C. Hu, Z. Kang, and Y. Liu, eds.), vol. 6382 of *Lecture Notes in Computer Science*, pp. 193–199, Springer Berlin Heidelberg, 2010.
 - [23] G. Tan, X. Cui, and Y. Zhang, “Chinese postman problem in stochastic networks,” in *Autonomic and Autonomous Systems and International Conference on Networking and Services, 2005. ICAS-ICNS 2005. Joint International Conference on*, pp. 78–78, 2005.
 - [24] B. S. E. Zoraida, “Dna algorithm employing temperature gradient for chinese postman problem,” in *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pp. 1–4, 2011.
 - [25] S. Prakash, M. Sharma, and A. Singh, “A heuristic for multi-objective chinese postman problem,” in *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, pp. 596–599, 2009.
 - [26] Z. Caner and T. Ekim, “Integer programming formulations for the minimum weighted maximal matching problem,” 2011.
 - [27] V. Kolmogorov, “Blossom v: a new implementation of a minimum cost perfect matching algorithm,” *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, 2009.
 - [28] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” 1968.
 - [29] C. Zato, A. Sanchez, G. Villarrubia, S. Rodriguez, J. Corchado, and J. Bajo, “Platform for building large-scale agent-based systems,” in *Evolving and Adaptive Intelligent Systems (EAIS), 2012 IEEE Conference on*, pp. 69–73, 2012.
 - [30] E. Argente, V. Julian, and V. Botti, “Mas modeling based on organizations,” in *Agent-Oriented Software Engineering IX* (M. Luck and J. Gomez-Sanz, eds.), vol. 5386 of *Lecture Notes in Computer Science*, pp. 16–30, Springer Berlin Heidelberg, 2009.
 - [31] F. Hoppner and F. Klawonn, “Clustering with size constraints,” 2008.

Aplicación Multiagente en Entornos Móviles para la Toma de Datos en Zonas de Aparcamiento

María Navarro¹ y Juan Manuel Corchado¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de la Merced s/n. 37008, Salamanca, España
{mar90ali94, corchado}@usal.es

Resumen En este artículo se presenta un trabajo de investigación en torno a las organizaciones virtuales, al reconocimiento de imágenes y a otros medios de identificación de objetos. Todo ello, tiene como objetivo el desarrollo de una aplicación de aparcamiento automático que ayude a los usuarios, y que además sea novedosa desde el punto de vista de la investigación actual. Se definirán diversas técnicas de investigación actuales y estudiaremos el problema de la integración de un sistema mediante organizaciones virtuales de agentes, el algoritmo para reconocer una matrícula mediante una fotografía y el uso del protocolo NFC para identificar un objeto del entorno.

Keywords: Organizaciones Virtuales, Protocolo NFC, Técnicas OCR, Redes Neuronales, Operaciones Morfológicas, Plataforma de Agentes, Clustering

1. Introducción

Hoy en día, el coche es una parte fundamental de nuestras vidas, no sólo para viajar, sino también para tareas cotidianas como ir al trabajo, ir de compras, etc. Frente a este aumento del uso de los vehículos, existen las limitaciones de tiempo y espacio de aparcamiento. Podemos considerar que el sistema de aparcamiento en zonas azules que tenemos hoy en día es poco eficiente e incómodo. El proyecto que se propone tratará de solventar estas dificultades para así mejorar la eficiencia.

Sin embargo, pese a la importancia de esta nueva aplicación que proponemos, el objetivo principal será la innovación en el propio desarrollo del software. A lo largo de este proceso se ha realizado un estudio exhaustivo sobre las diferentes tecnologías y sobre el comportamiento y la integración de agentes, base para futuros trabajos de investigación. Debido a la gran amplitud de este trabajo, se tuvo la necesidad de dividirlo en dos partes. En este artículo nos centraremos en la recogida, procesamiento y presentación de los datos, y con la interacción directa entre agentes internos y agentes externos como el usuario o el vigilante.

De forma general, el sistema desarrollado consiste en facilitar la tarea al usuario cliente y al vigilante. Así pues, existen dos posibles subdivisiones, según estemos pensándolo desde el punto de vista del cliente o del vigilante. El cliente

tendrá la posibilidad de gestionar sus pagos, recibir notificaciones si ha incurrido en alguna infracción o conocer la posición de estacionamiento de su vehículo en algún momento dado. Por su parte, el vigilante puede conocer en todo momento su propia posición, la ruta más óptima a seguir en su turno para recorrer las calles de forma eficiente y tiene acceso a los mecanismos de identificación del vehículo.

Dentro de esta identificación, cabe destacar la labor investigadora que se ha realizado, pues una de las propuestas de identificación es el reconocimiento de imágenes vía dispositivo móvil. Si bien existen ya estudios previos que intentan resolver de forma eficiente este problema [1], todavía hay dificultades a la hora de reconocer la matrícula, tales como la resolución de la imagen, el color, la iluminación o la carga en la computación. Algunas formas de solventar parcialmente estos problemas son el uso de cámara infrarroja fija, para conocer aproximadamente la posición de la matrícula y leer fácilmente los dígitos [2]. En nuestro caso, la posición de la cámara siempre es variable, así que la carga en la computación no debería ser muy alta para no ralentizar a la propia aplicación. Estas dificultades propuestas son la principal motivación para la investigación y mejora de las técnicas hasta ahora propuestas para el reconocimiento de matrícula utilizando un dispositivo móvil, así como la integración de este sistema dentro de un sistema multi-agente.

Este artículo se encuadra dentro de una línea de investigación en el desarrollo de organizaciones virtuales y en desarrollo de aplicaciones propias de Smart Cities, con el fin de facilitar las tareas al ser humano. El objetivo principal de este proyecto es la investigación en las técnicas en auge actualmente para su uso, así como la propuesta de nuevas técnicas para desarrollar mejoras en determinadas áreas del conocimiento como los agentes o el reconocimiento de imágenes. En particular, comprobaremos la adecuación de las organizaciones virtuales de agentes inteligentes en la resolución de estos problemas.

2. Estado del arte

El concepto de agente está presente en diversos campos de estudio, siendo diferentes sus características y comportamientos en cada uno. Aunque en los orígenes de este campo de estudio, solo se estudiaba el comportamiento de agente individual, la ciencia ha ido evolucionando y actualmente lo interesante es estudiar las organizaciones que los agentes pueden crear, los sistemas multiagentes [3].

En los desarrollos iniciales de sistemas multiagente, los diseñadores se centraron en el estudio del agente. No obstante, en los últimos años, el interés de los desarrolladores se ha dirigido a los aspectos organizativos de la sociedad de agentes. Debido a esta nueva concepción de los sistemas multiagente como sociedades u organizaciones virtuales, aparecieron nuevas metodologías [4]. Así, si el proceso viene guiado por la organización del sistema, se dice que la metodología es orientada a la organización (organizational-oriented methodology). Dentro

de estas, destacamos los trabajos realizados en GAIA [5] y GORMAS [6]. Nos centraremos en GORMAS por ser la metodología que se presenta en el artículo.

Esta metodología propone un nuevo modelo de Organización que permite describir la estructura, funcionalidad, normalización, dinamicidad y entorno de las organizaciones. Para ello, hace uso principalmente de los conceptos de unidad organizativa, servicio, entorno y norma. Este modelo de Organización Virtual (VOM) está compuesto por cuatro dimensiones:

1. Dimensión estructural: Describir los elementos del sistema y sus relaciones
2. Dimensión funcional: Detallar la funcionalidad específica del sistema, en base a servicios, tareas y objetivos;
3. Dimensión del entorno: Definir cada elemento del entorno al que se puede acceder mediante puertos de entorno
4. Dimensión normativa: Describir las normas de la organización, y los objetivos normativos que los agentes deben seguir.

Una vez realizado el diseño, es necesario llevarlo a la práctica mediante el uso de alguna plataforma de agentes que nos permita la comunicación e integración. En nuestro caso, hemos utilizado para el desarrollo de nuestra arquitectura el sistema PANGEA [7]. Este sistema es una plataforma orientada a servicios que fomenta al máximo la distribución de recursos. Por esta razón, todos los servicios son implementados como servicios Web, emulando una arquitectura cliente-servidor.

De manera general, la plataforma PANGEA incluye una serie de características que la hacen especialmente flexible [7] y adecuada para su despliegue e inclusión en proyectos de Smart Cities.

2.1. NFC

NFC son las siglas para referirse al concepto *Near Field Communication*. El NFC consiste en una interfaz de comunicación inalámbrica con una distancia de trabajo limitada a unos 10 cm, generalmente. La interfaz puede funcionar en varios modos. Los modos se distinguen si un dispositivo crea su propio campo de radio-frecuencia (RF) o un dispositivo recupera el poder desde el campo de RF generado por otro dispositivo. Si el dispositivo genera su propio campo se llama un dispositivo activo, de lo contrario se denomina un dispositivo pasivo.

El número de aplicaciones de este protocolo es muy amplio, y va desde la mera lectura de datos para conocer un teléfono o una dirección, hasta el intercambio de datos entre dos dispositivos NFC activos.

En el ámbito de la investigación, existen diversos estudios acerca de la posibilidad de aplicar la tecnología NFC para monitorización de ciertos parámetros relacionados con la medicina [8], o trabajos relacionado con el bienestar de las personas es el campo de la inteligencia ambiental [9] o estudios centrados en la seguridad, como las vulnerabilidades que el protocolo NFC puede presentar y sus posibles soluciones [10].

2.2. Reconocimiento de matrícula

A pesar de existir software comercial, el reconocimiento automático de matrículas está siendo investigado ampliamente [1, 2, 11]. Los sistemas comerciales en uso en la actualidad pueden obtener tasas de acierto superiores al 95 % en entornos controlados. En entornos en los que las condiciones de iluminación no están controladas (p.ej. mediante cámaras infrarrojas) o varía el tamaño orientación o escorzo de la placa, la tasa de acierto se reduce hasta el 75 %.

En el reconocimiento automático de matrículas podemos diferenciar dos fases: 1) Localización de la placa dentro de la imagen, paso crítico para la posterior identificación y 2) el reconocimiento y validación de los caracteres de la matrícula. Los métodos usados para localizar la matrícula dentro de la imagen son variados. Una aproximación usual es la de realizar una búsqueda por líneas. En cualquier caso, el resultado sería cada caracter por separado. Posteriormente, el reconocimiento de los caracteres segmentados es implementado a menudo mediante redes neuronales. La entrada de las redes puede ser una representación directa de los caracteres o indirecta, en la que los caracteres sufren alguna transformación previa [11]. Otros autores se decantan por la correlación, tal y como sucede en [12], aunque en este caso el procedimiento es modificado con ciertas reglas heurísticas para mejorar su tasa de acierto.

3. Descripción del sistema propuesto

Como ya se comentaba anteriormente, el sistema que se quiere construir está basado en organización de agentes. Cabe destacar que la parte más importante de nuestra investigación es la gestión de la identificación, pues es el lugar donde hemos centrado los estudios para el reconocimiento de la imagen, estudiando OCR junto con otros algoritmos de clustering y redes neuronales. También entra dentro de esta identificación el estudio del protocolo NFC. El primer paso en este apartado consistirá en la descripción del sistema basado en agentes, para posteriormente centrarnos en el objetivo de uno de ellos: la identificación de un objeto de entorno.

Antes de desarrollar una organización virtual, es necesario un diseño previo para distribuir la funcionalidad y definir la comunicación entre los distintos roles. Para ello, se ha empleado GORMAS [6] y, como ya anticipamos en el punto 3, PANGEA [7] ha sido nuestra plataforma final de desarrollo. Al desarrollar el diseño de la organización virtual y así establecer el modelo, es necesario analizar las necesidades y expectativas de los posibles usuarios del sistema. El resultado de este análisis serán los roles de los entes que participan en el sistema propuesto. En particular, se encontraron los siguientes roles:

- Usuario. Representa al usuario o cliente del sistema. En este caso, el usuario será el vigilante o el cliente. Los separamos porque el agente tendrá distinta funcionalidad según el rol que adquiera.
- Identificar Vehículo: Este rol se responsabiliza de llevar a cabo todo el proceso de identificación del vehículo, pudiendo ejecutar para ello diversos métodos según los parámetros de entrada.

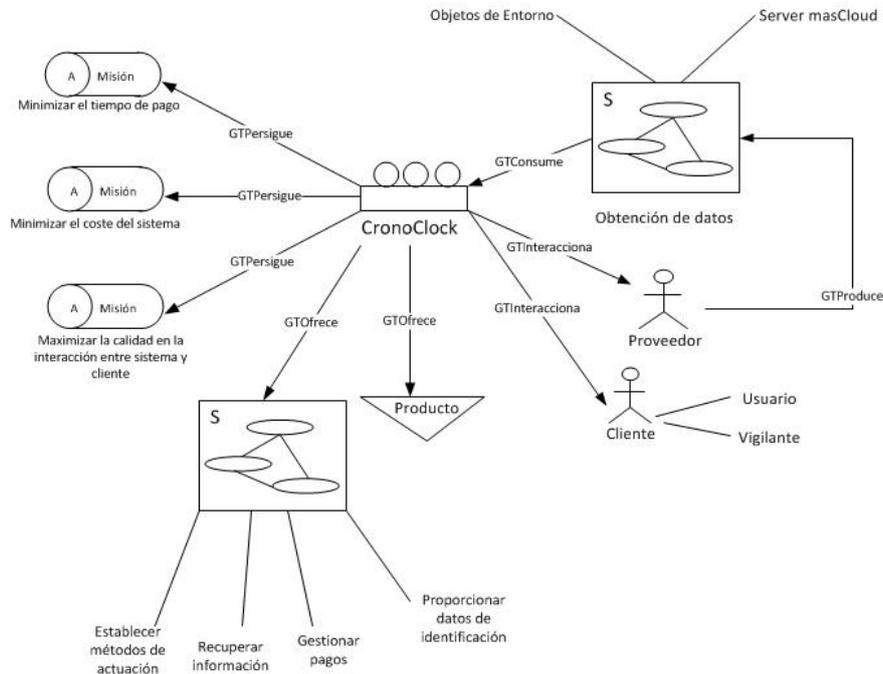


Figura 1. Diagrama GORMAS del modelo de organización (vista funcional). Se muestran los servicios del sistema, los roles potenciales del cliente y proveedor y la misión global del mismo

- **Notificador:** Es el encargado de presentar las comunicaciones del sistema. Puede llevar a cabo diversas notificaciones, en función de qué rol o agente se comunique con él. Este rol permite al usuario interactuar con el sistema, y es el encargado de recoger las valoraciones que pueda realizar dicho usuario.
- **Localizador:** Este proceso es responsable de localizar algunos objetos del entorno, como el vehículo o el usuario.
- **Infracciones:** Su responsabilidad es gestionar todo el sistema de infracciones que se producen cuando un cliente no tiene un comportamiento adecuado.
- **Comunicación:** Es el responsable de la comunicación entre la organización virtual del servidor central y dicha organización. Esta comunicación provoca la respuesta de ciertos agentes en función de los parámetros de entrada provenientes de la OV de dicho servidor.
- **Proveedor:** Este rol es un agente que tiene acceso y almacena toda o gran parte de la información necesaria para gestionar las acciones que gobiernan este sistema. Esta información la puede extraer a través de servicios como la obtención de datos del sistema masCloud [13], o mediante objetos del entorno tales como el tag NFC, imágenes o información suministrada por los roles del usuario.

- Supervisor: Un agente que ejerza este rol tendrá el control general del sistema. Analiza la estructura y sintaxis de todos los mensajes de entrada y salida del sistema.

Uno de los diagramas de la metodología es instanciar la vista funcional (misión) del modelo de organización, que representa los productos y servicios que ofrece el sistema, el tipo de entorno, la misión global que persigue y los grupos de interés a los que afecta. Puede observarse en la Figura 1.

3.1. Técnicas de identificación del vehículo

La gestión de la identificación del vehículo ha sido la parte central de nuestra investigación. En este proceso, se han explorado varias técnicas y se han elegido las que más se adaptaron a las necesidades. A continuación exploraremos cada una de ellas, comentando el proceso de estudio y desarrollo seguido.

El reconocimiento NFC es una de las técnicas que nos ha permitido reconocer un objeto del entorno, en nuestro caso, el vehículo. El objetivo es comunicar un dispositivo activo (dispositivo móvil) con un dispositivo pasivo (tag NFC). La ventaja de estos *tags* es que no necesitan ningún tipo de mantención ni baterías, se activan por el propio campo electromagnético.

Cada vehículo posee un número único que le identifica. Este número se graba de forma automática en la tarjeta y se cierra para que no se pueda volver a escribir. De esta forma, el usuario puede leer la tarjeta con su propio móvil e identificar el vehículo.

Por otro lado, para la identificación del vehículo mediante imagen se realizó una investigación acerca de diferentes programas ya en explotación, pero también sobre las facilidades que soporta la biblioteca OpenCV [14]. Actualmente, OpenCV implementa un módulo para interactuar con Android de forma más eficiente, aunque se pueden utilizar algoritmos nativos de la biblioteca tales como el algoritmo de las K-medias o las redes neuronales, sin aumentar notablemente el tiempo de ejecución. Por ello, se optó por esta herramienta como la principal para analizar y procesar la imagen para extraer el texto. El paso siguiente era procesar la imagen para extraer las características básicas de la misma y poder reconocer los dígitos. En concreto, teníamos por delante 4 etapas fundamentales:

1. Extraer la placa de la matrícula
2. Extraer cada dígito por separado
3. Reconocer el dígito
4. Comprobar la veracidad de lo reconocido

Para extraer la placa, se estudió la composición de la misma, considerando sólo las matrículas de los turismos (la mayoría de nuestros clientes). Su diseño es único para todo el país. En la parte izquierda lleva una franja azul con el símbolo de la Unión Europea y la letra E de España, y contiene tres letras que van desde la BBB a la ZZZ, y 4 dígitos que van desde el 0000 hasta el 9999. Este conjunto de caracteres es la clave necesaria para la identificación de cada vehículo. Cabe

destacar que se han suprimido las cinco vocales, y las letras Ñ, Q (por ser fácil su confusión con la letra N y el número 0, respectivamente). Al ser una placa blanca con letras en negro, nos facilita en cierta medida el tratamiento de la imagen. El objetivo del preproceso es obtener una imagen binarizada, donde sólo existan los colores negros y blancos.

Existen muchas técnicas para mejorar la calidad de la imagen y binarizarla. De hecho, muchas veces el éxito del algoritmo de reconocimiento depende en gran medida de este pre-proceso y de la extracción de dígitos. En este estudio se han probado diversas formas de procesar la imagen, aunque la que mejor resultados ha dado ha sido la que describiremos a continuación.

Para minimizar la carga dentro del dispositivo móvil, el primer paso es realizar una transformación de la imagen a color a la imagen en escala de grises, para posteriormente, estudiar la eliminación de ciertas imperfecciones en la imagen o ruido. Para ello, se utilizó una técnica de suavizado de contornos, de forma el color de un punto dependiera de forma parcial de los puntos de alrededor. Así, podríamos eliminar manchas u otras imperfecciones que dificulten la identificación de la placa. Además de este suavizado, es necesario aplicar alguna transformación morfológica. Existen varias, que se han estudiado y se ha elegido la que mejores resultados nos ha dado. El proceso total se muestra en la Figura 2.



Figura 2. Esquematización del proceso de transformación de la imagen para la identificación de la placa de la matrícula

Para extraer los dígitos, se siguió un proceso similar. En la imagen 3 mostramos el resultado final.



Figura 3. Esquemización del proceso de transformación de la imagen para la identificación de los dígitos de la matrícula

Para reconocer el dígito se estudiaron dos técnicas. Por un lado, se contemplaron diversos algoritmos de clustering y finalmente se optó por el estudio del algoritmo del vecino más cercano. Por otro, se entrenó una red neuronal y se observaron los resultados obtenidos. Para ambos casos se ha tenido que crear un conjunto de datos de entrenamiento, y otro de prueba. Para el entrenamiento, se crearon 8 ejemplos de cada letra, teniendo en cuenta que pueden tener diversas inclinaciones (incluidas en el set de entrenamiento). Con este set de entrenamiento, se entrenó la red neuronal y el algoritmo de clustering, y después se probó con diversas imágenes recogidas de ejemplos reales, para estudiar cuál es el nivel de precisión y el error total cometido.

4. Resultados

El grado de precisión a la hora de identificar el vehículo con NFC siempre es del 100%. Las pruebas han sido realizadas con cuatro tipos de tags NFC de distintas tecnologías, según la capacidad (0.5K, 1K o 2K), siempre con idénticos resultados. Los posibles problemas de seguridad no se han tenido en cuenta, debido a la brevedad de la comunicación entre el tag y el dispositivo móvil. Aun así, es conveniente estudiar las nuevas innovaciones para proteger este tipo de comunicación. Este estudio queda ya en líneas futuras. Hubo dos dificultades a la hora de recoger los datos. La primera fue desarrollar un agente que recogiera correctamente los datos mediante los protocolos necesarios de forma eficiente. Afortunadamente, existen interfaces que facilitaron este intercambio de información entre el dispositivo activo y el pasivo. La segunda dificultad fue estudiar

cuál era la distancia máxima a la que el dispositivo móvil detectaba el tag. Esto era importante a nivel funcional, pues el vigilante quiere la mayor comodidad posible para poder leer las etiquetas. Por ello, se investigó acerca de diferentes formas para aumentar la distancia de reconocimiento. Al final, la conclusión a la que llegamos fue que dependíamos mayoritariamente del hardware y del tipo de dispositivo activo que utilizáramos.

Para estudiar los resultados obtenidos con reconocimiento de la matrícula en una imagen, poniendo especial atención en el tipo de iluminación y características, y en el tipo de algoritmo o estructura implementada. Se han realizado pruebas con diversas imágenes de matrículas, y posteriormente se estudió cómo es el reconocimiento de los caracteres de la imagen.

El algoritmo del vecino más cercano se configuró para que considerara 50 vecinos a la hora de clasificar cada carácter. Por su parte, las redes neuronales fueron un poco más complejas de definir y entrenar. Se consideró sólo una red de perceptrón multicapa, con una capa oculta. El primer problema fue precisamente determinar la estructura más adecuada de la red. Desde un primer momento, consideramos una red con una entrada por píxel. Se determinó que cada imagen debía tener 12 píxeles de alto por 12 píxeles de ancho, lo cual nos dejaba con una red neuronal de 144 entradas. El número de salidas tenía que ser igual al total de las letras (28 neuronas de salida, una por cada letra). Tras varias pruebas, se fijó el número de neuronas de la capa oculta en 150, con una función de activación tangencial (las más utilizadas para reconocer caracteres (Holley, 2009)).

Presentamos a continuación una tabla ilustrativa (Tabla 1) con los resultados obtenidos al utilizar el algoritmo del vecino más cercano, comparándolos con las redes neuronales. También se presenta el porcentaje de acierto a la hora de reconocer la placa de la matrícula y cada uno de los dígitos. Los datos han sido calculados al probar 25 matrículas cualesquiera.

Tabla 1. Tabla comparativa de los resultados obtenidos en cada fase del reconocimiento

| Etapa | Clustering | Redes Neuronales |
|------------------------------|------------------|------------------|
| Acotamiento de la matrícula | 84 % (21/25) | |
| Acotamiento de los dígitos | 86'5 % (173/200) | |
| Reconocimiento de caracteres | 60 % (120/200) | 80'5 % (161/200) |

Tras un análisis más detallado, se reveló que algunos caracteres como la M o la W se reconocían en la mayor parte de los casos, mientras que otros eran fácilmente confundidos entre sí. Aun así, el nivel de error aumentaba si las matrículas probadas estaban muy inclinadas o con poca resolución. En concreto, observamos que la letra B era reconocida frecuentemente como el número 8, la G como el número 6. Esta confusión entre números y letras se pudo solventar mediante la introducción de una condición. Si el dígito reconocido era uno de los

cuatro primeros, se reconocería como una letra. En caso contrario, se reconocería como un número. De esta forma, si por ejemplo, se detectara un 8 en el segundo dígito, lo consideraríamos una B. Por supuesto, una mejora de esta solución sería posible si el reconocedor se mejorara con algunos patrones de lógica difusa o con más sets de entrenamiento.

Por otra parte, tampoco se reconocieron correctamente algunos caracteres que tenían patrones similares. Por ejemplo, la X, la K y la Y, o la B y la G. Para este último problema la única solución plausible era engordar el número de patrones de entrenamiento, y así mejorar el reconocedor.

Los tiempos de ejecución del reconocimiento de la matrícula, dependían de las diferentes características de la imagen. Aun así, podemos fijar el tiempo de ejecución medio alrededor de los 4 segundos. Es un tiempo aceptable, aunque un poco lento para el entorno móvil en el que se está desarrollando. La mejora de este tiempo y de los resultados mostrados queda pendiente como líneas futuras.

Otro punto a destacar, fue el éxito en el desarrollo de las organizaciones virtuales para la consecución de los objetivos de este trabajo. El proceso de identificación y organización de roles ayudó a la mejora de la gestión y por tanto a la mejora de la eficiencia. El uso de PANGEA como plataforma permitió una comunicación fluida entre agentes, que queda patente en el diseño de la aplicación.

5. Conclusiones

El trabajo de investigación presentado en esta memoria pretende aportar nuevas perspectivas en el ámbito del reconocimiento de imágenes, utilizando los sistemas multiagente. Cabe destacar también la investigación realizada en el campo de las organizaciones virtuales, imprescindible además para la gestión adecuada de la aplicación. En este sentido, se ha estudiado y utilizado una metodología orientada al desarrollo de organizaciones que ha permitido cubrir el análisis, diseño de la estructura organizativa y diseño de la dinámica de la organización de forma eficiente, demostrando que el sistema es viable para resolver nuestro problema de automatización del aparcamiento.

Hemos demostrado que existe una forma viable de reconocer dígitos mediante una cámara de un dispositivo móvil. Sin embargo, los resultados no han sido tan buenos como esperábamos en un principio. Como línea futura principal se propone la mejora de resultados no sólo en la precisión, sino en el tiempo de ejecución, fundamental dada la limitación de recursos del dispositivo móvil y el carácter del mismo (fácil de usar y rápido). Algunas propuestas para disminuir este tiempo podría ser reducir el número de neuronas en la red neuronal, o utilizar otro algoritmo de clustering, otra estructura de red neuronal o incluso cambiar el tipo de red si esta no satisface nuestras necesidades. Aun así, esto nunca debería ir en detrimento de la precisión de los resultados, que es aún más importante.

En lo referente a estos resultados, la idea fundamental y que prevalece, podría ser realizar un sistema que evolucione de forma dinámica. Este dinamismo

puede venir dado por una retroalimentación del set de entrenamiento. Una posibilidad sería que el sistema, después de que el usuario compruebe si ha sido bien reconocida la matrícula, guarde la imagen de cada dígito en algún sistema de almacenamiento dentro del servidor para engrosar así el set de entrenamiento. Cada cierto tiempo con estos nuevos datos se pueden re-entrenar los algoritmo para mejorar la tasa de aciertos.

Además de estas mejoras, esta aplicación puede ser analizada para futuras investigaciones en otros campos de investigación. Por ejemplo, se podría indicar al usuario qué plazas están libres, buscando un nuevo algoritmo de lógica difusa que decida en función del tráfico y de una serie de parámetros determinados, cuánta probabilidad tiene de encontrar una plaza libre en una calle determinada. Este algoritmo estaría dentro de un campo de investigación que en esta memoria y en este trabajo no se ha explorado aún.

Bibliografía

- [1] J. M. López Fernández, “Software para el reconocimiento automático de matrículas,” 2006.
- [2] N. Vázquez, M. Nakano, , and H. Pérez-Meana, “Automatic system for localization and recognition of vehicle plate numbers,” *Journal of Applied Research and Technology*, 2002.
- [3] S. Rodríguez, *Modelo Adaptativo para organizaciones virtuales de agentes*. PhD thesis, University of Salamanca, 2010.
- [4] J. Ferber, O. Gutknecht, and F. Michel, “From agents to organizations: an organizational view of multi-agent systems,” in *AAMAS03 - Agent-Oriented Software Engineering Workshop*, 2003.
- [5] F. Zambonelli, N. Jennings, and M. Wooldrige, *ACM Transactions on Software Engineering and Methodology*, ch. Developing Multiagent Systems: The Gaia Methodology. 2003.
- [6] E. Argente and I. V. J. Botti, *GORMAS: Guía para el desarrollo de sistemas multiagente abiertos basados en organizaciones*. PhD thesis, 2008.
- [7] C. Zato, G. Villarrubia, A. Sánchez, I. Barri, E. R. Soler, A. F. del Viso, and otros., *PANGEA - Platform for Automatic coNstruction of orGanizations of intElligent Agents*. Springer Verlag, 2012.
- [8] K. Patrick, G. Griswold, F. Raab, and S. S. Intille, “Health and the mobile phone,” tech. rep., 2008.
- [9] J. Morak, H. Kumpusch, D. Hayn, R. Modre-Osprian, and G. Schreier, “Design and evaluation of a telemonitoring concept based on nfc,” in *Enabled Mobile Phones and Sensor Devices. IEEE Transactions on Information Technology in Biomedicine*.
- [10] E. Haselsteiner and K. Breitfuss, “Security in near field communication. strengths and weakness,” tech. rep., 2012.
- [11] C. Shyang-Lih and C. Sei-Wan, “Automatic license plate recognition,” in *IEEE Trans. On Intelligent Transportation Systems*.
- [12] F. Martín and D. Borges, “Automatic car plate recognition using partial segmentation algorithm,” in *SPPRA*.
- [13] F. De la Prieta, S. Rodríguez, J. Bajo, and J. M. Corchado, *A Multiagent System for Resource Distribution into a Cloud Computing Environment*. Springer Verlag, 2013.
- [14] G. Bradski and A. Kaehler, “Computer vision with the opencv library,” tech. rep., 2008.

Estudio de Técnicas de Análisis Visual de Guiones Cinematográficos

Daniel Toimil Martín¹ y Roberto Therón Sánchez¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008 Salamanca, España
{toimild,theron}@usal.es

Resumen Los guiones cinematográficos contienen gran cantidad de información susceptible de ser representada de forma gráfica. En este trabajo se propone una nueva representación orientada a la información estructural de los guiones y a las interacciones de los personajes a lo largo de las escenas. La representación planteada presenta un abanico de técnicas visuales orientadas a mejorar la capacidad expresiva y analítica de los guiones y que tiene en cuenta las características inherentes a la forma de percibir de los seres humanos con el fin de mejorar la transmisión de conocimientos a partir de los datos.

Keywords: Visualización de la Información, Analítica Visual, Guiones Cinematográficos, Análisis Estructural

1. Introducción

Los guiones cinematográficos son documentos que contienen una gran cantidad de información. La lectura y análisis de los mismos es una tarea larga y compleja que requiere, en muchos casos, más de una lectura. Sin embargo, la lectura es un proceso que no favorece el análisis porque no mantiene una visión global de la historia y la estructura, si no que es necesario recordarla a medida que se avanza en la lectura.

Las técnicas de visualización de la información proporcionan una manera útil y eficaz de realizar análisis sobre los datos de los guiones, ofreciendo una visión global de los mismos y favoreciendo la búsqueda de patrones, estructuras y relaciones entre personajes al margen de la lectura del guion.

El presente trabajo propone una representación visual de guiones cuyo objetivo es proporcionar una manera eficaz de transmitir la información subyacente en los datos presentando a la vez un aspecto estético agradable. La representación propuesta pone especial énfasis en la información estructural, incluyendo las escenas, los personajes y sus relaciones, validándolos con la teoría narrativa desarrollada por analistas de guiones profesionales.

2. Visualización de la Información

La visualización de la información, cuyas primeras infografías [1, 2] datan del siglo XVII, ha evolucionado notablemente hasta la actualidad. La incorporación

de los ordenadores a los procesos de visualización ofrece una gran cantidad de nuevas posibilidades para las representaciones visuales e incrementa la cantidad de datos que se pueden capturar y examinar, permitiendo así la aplicación de técnicas personalizadas que cubren las necesidades de varios tipos de datos [3].

Las representaciones visuales nos ayudan a ilustrar conceptos que, si son expresados verbalmente, resulta complicado que el interlocutor pueda entenderlos fácilmente. Cuando se tienen datos con los que se quiere ilustrar conceptos, ideas o propiedades intrínsecas a esos datos, el uso de representaciones visuales ofrece una herramienta de comunicación muy útil.

Según Tufte [4]: “la excelencia en los gráficos estadísticos consiste en comunicar ideas complejas con claridad, precisión y eficiencia”. Consecuentemente, una representación debe buscar ofrecer tanta información como sea posible procesar de manera rápida, usando correctamente el espacio limitado disponible.

El objetivo principal de la visualización es la transmisión de la información subyacente en los datos, de tal manera que el observador la perciba fácilmente, minimizando el esfuerzo y el tiempo necesarios. Card et al. [5] definen la visualización de la información como “el uso de representaciones visuales de los datos, interactivas y asistidas por ordenador, para amplificar la cognición”.

Esta transmisión de conocimiento a partir de los datos se hace más efectiva cuando se aprovecha la capacidad innata del ser humano de procesar información a través del sentido de la vista. Larkin and Simon [6] demostraron, a través de un estudio, como el uso de representaciones visuales conlleva una mejora perceptual, aumentando la comprensión y la capacidad de análisis del observador.

3. Teoría Narrativa

Para comprender los resultados presentados en este trabajo y valorar la representación propuesta, resulta interesante plantear una serie de conceptos básicos sobre teoría narrativa y guiones cinematográficos.

Se puede definir el guion como un documento de producción en el que se expone el contenido de una obra cinematográfica con los detalles necesarios para su realización [7]. El guion contiene diferentes escenas, acciones y diálogos entre personajes, acontecimientos, descripciones del entorno y acotaciones breves para los actores sobre la emoción con la que se debe interpretar.

Con el paso del tiempo, el formato de los guiones se ha ido estandarizando de manera que la industria del cine espera que todos los guiones profesionales cumplan ciertos requisitos que van, desde la tipografía utilizada, hasta los márgenes o los cambios de escena: si la acción sucede en un interior o en el exterior, si es de día o de noche, etc.

No sólo el formato del documento es algo estándar, sino que las estructuras de las historias de los guiones siguen también ciertos estándares que alrededor del 95 % de las películas cumplen (hasta el 100 % cuando se habla de las películas comerciales de Hollywood) [8]. Estas estructuras, largo tiempo usadas, se han establecido en los espectadores hasta el punto de que la gente tiene un sentido

inherente de dichas estructuras y sienten que algo “no va bien” en una película si é no se ajusta, en mayor o menor medida, a ellas.

A grandes rasgos, Calvisi [8] define la forma del guion como una estructura en cuatro actos (en realidad tres actos con el segundo acto dividido en dos) que ocupan, aproximadamente, el 25 % de la película cada uno:

- **Acto uno:** Se plantea el tema central de la historia y ocurre una crisis sobre la que el protagonista debe tomar una decisión.
- **Acto dos-A:** El protagonista entra en acción. Crecen el conflicto y los problemas.
- **Acto dos-B:** Las cosas se desmoronan, el conflicto alcanza el punto máximo y el protagonista toca fondo.
- **Acto tres:** Todas las subtramas convergen y se produce el enfrentamiento final y la resolución de la historia.

Otro elemento a tener en cuenta son los personajes y su función en la historia descrita en el guion. En este trabajo se consideran los siguientes roles como los fundamentales propuestos en [9, 10]:

- **Protagonista:** El personaje central de la historia, que evoluciona a lo largo de la historia para alcanzar sus objetivos principales.
- **Atractor:** Es el personaje conectado con la parte emocional del protagonista y está ligado a su crecimiento emocional a lo largo de la historia.
- **Mentor:** Es quien guía al protagonista, está ligado a su crecimiento intelectual y a menudo, lo entrena o sirve como fuente de sabiduría.
- **Némesis:** Usualmente, el enemigo del protagonista y quien lo mantiene alejado de sus objetivos.
- **Trickster:** A veces amigo, a veces enemigo. Pone trabas al protagonista y pone a prueba la evolución personal del protagonista, que lo acerca a sus objetivos.

Estos arquetipos no son únicos y no están presentes en todas las películas, pero son elementos muy habituales que resultan interesantes para el análisis de guiones.

4. Representación de Películas: Trabajos Relacionados

Existen diferentes trabajos previos relacionados con representación de películas aunque desde un enfoque muy diferente al planteado en este trabajo. Por ejemplo, Brodbeck [11] propone una visualización en forma de anillo cuyo objetivo es crear una imagen global de la película que sirva como “huella dactilar” visual, resaltando las características de la película y permitiendo encontrar patrones diferentes según la “huella” de cada una. Sin embargo, este trabajo no resulta adecuado para un análisis más profundo de la estructura y las relaciones entre personajes presentes en el guion. Brodbeck afirma que “nadie ve una película como realmente es, si no siempre una vista parcial”, en el sentido de que

su representación permite una visualización global del aspecto de la película, en contraste con la secuencia de escenas no simultáneas que se ven al visionar la película. En el presente trabajo, a pesar del distinto enfoque, se proporciona también una visualización que permite tener un modelo mental del “aspecto” global de la película.

La representación propuesta por Munroe [12] en su viñeta XKCD muestra una imagen donde utiliza líneas para representar los personajes de las películas a lo largo de la historia. A partir de esta idea de representación, Ogievetsky [13] creó una herramienta de edición que permite construir manualmente dichas representaciones haciendo uso de un programa informático. Posteriormente, dentro de un proyecto de investigación del grupo VisUsal, Agudo [14] planteó una representación con muchas de las características de aquellos trabajos pero que generaba la representación de manera automática a partir de la información contenida en los guiones (ver Fig. 1). Este trabajo se enmarca dentro de ese proyecto y desarrolla la propuesta inicial de Agudo con el fin de conseguir una representación capaz de transmitir la información de manera más efectiva y con un nivel más alto de interactividad.

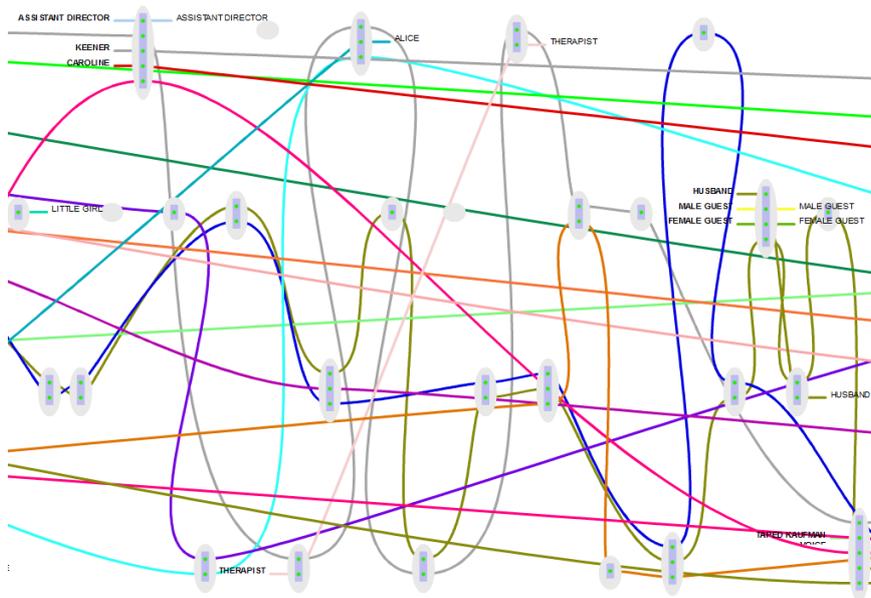


Figura 1. Representación visual de guiones realizada por Agudo.

Por último, la representación planteada puede verse como un tipo de coordenadas paralelas [15] en el sentido de que las escenas del guion pueden verse como ejes paralelos por donde los personajes se sitúan dependiendo de si participan

en la escena y por tanto, convergen en ella o, por el contrario, no participan y se alejan hacia la parte exterior de la representación. Existen otros trabajos previos que también usan variaciones y extensiones de las coordenadas paralelas para analizar la evolución de datos a través de coordenadas paralelas animadas [16] o para análisis de información diacrónica de diccionarios [17].

5. Decisiones de Diseño

Los guiones contienen gran cantidad de datos susceptibles de ser representados. Elegir correctamente la manera de representarlos es una tarea crítica para conseguir una representación eficaz.

En primer lugar, es necesario establecer la fuente de datos del guion que, en este caso, son ficheros XML generados automáticamente a partir de los guiones escritos y que contienen la mayor parte de la información de los documentos necesaria para la representación. Esta tarea es posible gracias a la estructuración más o menos estándar que presentan los guiones.

El resto de esta sección contiene las principales decisiones y principios de diseño que se han tenido en cuenta a la hora de elaborar la representación.

5.1. Mapeo de los Datos a Elementos Visuales

Los datos relevantes que se muestran en la representación deben estar asociados a elementos visuales que los representen. Los elementos visuales han sido elegidos atendiendo a la Tabla 1 propuesta por Mazza [18].

Tabla 1. Codificación de datos cualitativos, ordinales y categóricos usando diferentes atributos (✓: apropiado, -: limitado, ✗: no apropiado).

| Atributo | Cuantitativo | Ordinal | Categórico |
|-----------------------|--------------|---------|------------|
| Color | | | |
| Matiz | ✗ | ✗ | ✓ |
| Intensidad | - | ✓ | ✗ |
| Forma | | | |
| Orientación | - | - | ✗ |
| Longitud | ✓ | - | ✗ |
| Ancho | - | - | ✗ |
| Tamaño | - | - | ✗ |
| Colinealidad | ✗ | ✗ | ✗ |
| Curvatura | - | - | ✗ |
| Agrupamiento | ✗ | ✗ | ✗ |
| Marcas | ✗ | ✗ | ✓ |
| Forma | ✗ | ✗ | ✓ |
| Numerosidad | ✓ | ✓ | ✗ |
| Posición espacial | | | |
| Posición 2D | ✓ | ✓ | - |
| Profundidad | ✗ | ✗ | ✗ |
| Concavidad/convexidad | - | - | ✗ |
| Movimiento | | | |
| Parpadeo | ✗ | ✗ | - |
| Movimiento | - | - | ✗ |

La lista de los principales datos mapeados son:

- **Atributos de Personaje:**
 - **Personaje:** Cada personaje aparece como una línea que recorre horizontalmente la representación y cuyo color la distingue de las líneas de otros personajes.
 - **Nombre:** Representado mediante etiquetas textuales al inicio y al fin de la línea de personaje.
 - **Relevancia del personaje:** Se indica mediante el grosor de la línea de personaje, siendo más gruesa cuanto más importante es el personaje en función del número de escenas en que aparece.
- **Atributos de Escena:**
 - **Escena:** Representadas por elipses que se van sucediendo horizontalmente en el mismo orden en que aparecen en el guion.
 - **Número de escena:** Están indicadas mediante etiquetas numéricas en la parte superior de la representación, coincidiendo verticalmente con la escena asociada.
 - **Datos de cabecera:** Fundamentalmente los datos de localización. Se muestran en una vista de detalle de la escena mediante una etiqueta textual (lugar) y un icono (interior/exterior).
 - **Número de personajes:** Se muestran mediante puntos contenidos en la elipse de la escena. Esto afecta también a la altura de la escena, consiguiendo un efecto interesante y perceptible a simple vista donde las escenas más altas tienen más personajes.
 - **Duración:** Codificado en la vista de duración, descrita más adelante, a través de la anchura de la elipse de la escena.

Otra información importante se representa indirectamente como consecuencia de los mapeos elegidos anteriormente:

- **Evolución de los personajes:** Se representa como resultado del recorrido de la línea de personaje a lo largo de la representación y mientras va pasando por las escenas en las que aparece.
- **Importancia de la escena:** La relevancia de la escena está planteada como una relación entre el número de personajes que contiene y su duración. Ya que el número de personajes influye en el alto y la duración en el ancho, la relevancia está codificada mediante el área de la escena.

5.2. Algoritmo de Distribución de Elementos

Una de las decisiones críticas en el diseño de la representación es la elección de las posiciones de los elementos en la imagen final. Los personajes (líneas de personaje) y las escenas (elipses) son los elementos principales de la representación y por tanto su colocación es fundamental para que la representación transmita correctamente la información estructural del guion así como las relaciones y evolución de los personajes.

Ogievetsky [13] propone en su herramienta un algoritmo genético que realiza un reposicionamiento de las escenas minimizando los cruces y mejorando así la imagen final generada. Sin embargo, en su trabajo la representación se crea manualmente y no incluye todos los personajes y escenas del guion, sino aquellos considerados más relevantes por aquel que crea la representación. En este trabajo, se propone una generación automática y exhaustiva de la representación, que incluya todos los personajes y escenas, de tal manera que contenga toda la información necesaria para un correcto análisis del guion, permitiendo el filtrado a demanda de personajes poco relevantes cuando resulta necesario. Por esta razón, la cantidad de elementos a posicionar es mucho mayor y un algoritmo genético tiene un coste temporal mucho mayor que en el trabajo de Ogievetsky. Además, debido a la naturaleza aleatoria de los algoritmos genéticos, las soluciones que generan producen líneas de personaje con muchas oscilaciones que provocan ruido y empeoran la eficacia de la representación.

Como solución al problema se plantea un algoritmo voraz, que busca posicionar las escenas y personajes minimizando en cada paso el número de cruces en las líneas de los personajes, ya que estos cruces generan dificultades para obtener información a partir de la representación. Por otro lado, para conseguir una representación más limpia, se definen n posiciones posibles de los personajes en la representación, donde n representa el número de personajes total del guion. De esta manera, en cada paso horizontal de la representación se deberán posicionar tanto los personajes como la escena en alguna de esas n posiciones con la restricción de que una posición solo puede ser ocupada por un personaje. Debido a que los personajes deben converger en cada escena en la que aparecen, será necesario hacer modificaciones en las posiciones asignadas y dichas modificaciones dependerán exclusivamente de la posición de la escena. Por esta razón, la minimización del número de cruces en cada paso horizontal se basa en elegir la posición de la escena que menos cruces provoque.

Como muestra el Algoritmo 2, se parte de unas posiciones iniciales de los personajes que se calculan situando en las posiciones centrales a los personajes más relevantes. El objetivo es mantenerlos en la zona central de la representación. Las posiciones de las escenas se calculan a partir de las posiciones de los personajes en el paso anterior, minimizando el número de cruces generado. Las posiciones restantes de los personajes están determinadas por la posición de cada escena.

5.3. El Color

La elección de los colores de la representación no es una tarea trivial e influye directamente en la capacidad de transmitir información. Brewer et al. [19] y Bostock et al. [20] proponen una lista de colores adecuados para la representación de atributos categóricos y que minimizan las dificultades de identificación para las personas con problemas de percepción de colores como el daltonismo. La elección de los colores, tanto de las escenas como de las líneas de personaje, está basada en esos trabajos con el objetivo de mejorar la eficiencia de la representación, consiguiendo además un aspecto estético más agradable.

Algoritmo 2 Algoritmo de posicionamiento

```

posPersonajes ← generaPosicionesInicialesPersonajes()
añadir posPersonajes a solucion
for all escenas do
    mejorPosicion ← calcularMejorPosicionEscena(posPersonajes)
    posPersonajes ← calcularPosicionPersonajes(mejorPosicion)
    añadir posPersonajes a solucion
end for

return solucion

```

5.4. Uso de Transiciones

La representación propuesta en este trabajo incluye varias interacciones que permiten acceder a información no disponible en la representación inicial. Esto es debido a que representar a la vez toda la información disponible supone una sobrecarga importante y satura la imagen reduciendo la eficacia en la transmisión de información. Algunas de las interacciones conllevan la recolocación de los elementos representados, suponiendo un cambio de contexto.

En aquellas interacciones en las que el cambio en la representación es sustancial, se ha incluido una animación que sirve de transición entre la imagen inicial y la final. Esto permite que el usuario mantenga el contexto durante el cambio de vista, reduciendo así la carga cognitiva. Consiste, básicamente, en realizar una interpolación visual de los cambios en la representación [21].

5.5. Vista de Duración de Escenas

Una de las interacciones fundamentales de la representación consiste en una vista en la que la duración de las escenas está representada mediante su anchura. A través de esta vista, se pueden identificar fácilmente las escenas más importantes y relevantes del guion. La razón de que la duración no esté representada en la vista básica es que la gran diferencia de duración entre las escenas es suficientemente grande como para que las diferencias de tamaño afecten al análisis de otros aspectos del guion. La Fig. 2 muestra un ejemplo de esta vista.

6. Análisis de Guiones

En esta sección se muestran dos guiones de ejemplo usando la representación propuesta y se analizan algunas de las características que se pueden extraer haciendo uso de la representación. Los ejemplos aquí mostrados son estudios breves que pueden ser ampliados si se realiza un análisis más profundo y exhaustivo y solo pretenden ser una muestra de la capacidad de análisis que ofrece la representación.

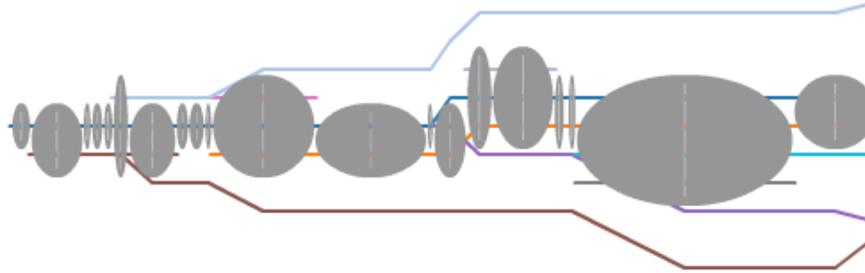


Figura 2. Vista de la duración de las escenas a través del ancho.

6.1. Babel

El primer guion representado (Fig. 3) corresponde a la película Babel dirigida por Alejandro González Iñárritu. Una vista global permite ver varias características interesantes:

- **Desarrollo de la historia:** Se puede observar una forma de cono en la representación que comienza con solo unas pocas líneas y va creciendo a medida que se desarrolla horizontalmente. Esto muestra la evolución de la película en general, que comienza de manera más simple (acto uno) y, a medida que la película avanza (acto dos-A y dos-B), se van presentando más historias y la trama se complica. La altura creciente de la representación codifica esta complicación de las tramas. Sin embargo, en un punto determinado de la representación (que coincide aproximadamente con el acto tres) la altura de ésta decrece, representando la convergencia de las historias típica del acto tres mencionada en la Sección 3, alcanzando la convergencia de todas las líneas en las escenas finales correspondientes al clímax de la película. La representación muestra, por tanto, parte de las características de cada acto planteadas por Calvisi [8].
- **Historias paralelas:** Babel es una película que sigue un formato de historias paralelas, donde se narran varias historias independientes que se van cruzando en algún momento de la película. Si se observa detenidamente, se pueden observar agrupaciones de líneas de personaje que evolucionan juntas y representan dichas historias independientes.
- **Interacciones entre personajes:** Otra característica que se puede observar son unas líneas que “saltan” de su historia principal a otra de las historias. Esto representa la interacción de los personajes con otras historias y queda representado de manera clara.
- **Convergencia de historias:** En la parte final del guion, las historias acaban juntándose dando lugar al final de la película. Esto se puede apreciar en los saltos de todas las líneas de personaje restantes hacia las escenas finales.

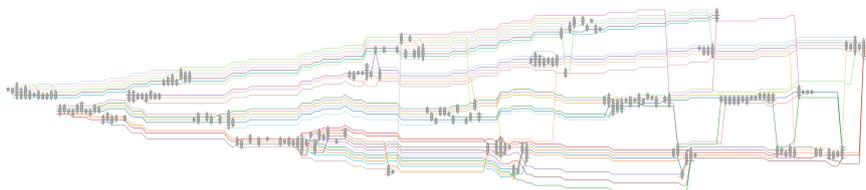


Figura 3. Representación de Babel (2006).

6.2. Rocky

La segunda representación (Fig. 4) muestra el guion de Rocky, dirigida por John G. Avildsen. En este caso, se pueden ver algunas otras características interesantes extraídas directamente de la representación:

- **Personajes principales:** La representación mostrada tiene los personajes filtrados, de tal manera que solo se muestran 5 personajes. El algoritmo empleado para este filtrado es capaz de elegir los personajes más importantes. En este caso, los personajes mostrados corresponden precisamente a los personajes más relevantes y que representan los roles comentados en la Sección 3 y analizados para esta película por Saylor [22]:
 - Protagonista: Rocky, línea azul. Personaje principal. Se puede ver como, siendo el protagonista, interviene en casi la totalidad de las escenas. Su línea es la más gruesa, ayudando a resaltar su importancia.
 - Atractor: Adrian, línea roja.
 - Mentor: Mickey, línea morada.
 - Némesis: Apollo, línea verde.
 - *Trickster*: Paulie, línea naranja.
- **Localizaciones:** En esta representación se pueden apreciar de manera bastante visual los cambios de localización del guion. La imagen muestra diferentes series de escenas juntas representadas a diferentes alturas. Estos saltos, que son seguidos por la línea azul del protagonista, se corresponden casi en su totalidad con cambios en el lugar donde sucede la acción de las escenas, ofreciendo esa información de manera intuitiva y eficaz.

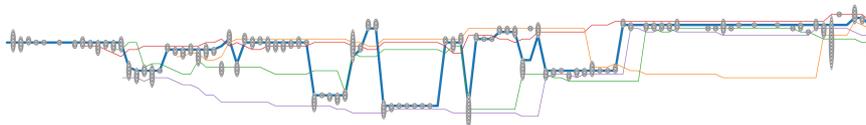


Figura 4. Representación de Rocky (1976)

7. Conclusiones y Trabajo Futuro

Se ha presentado una representación visual que, partiendo de trabajos anteriores [12, 13, 14], propone nuevas técnicas que mejoran la representación, permitiendo un análisis profundo de la estructura del guion y las relaciones entre los personajes. A través del uso de interacciones, apoyadas en algunos casos por transiciones, se proporcionan herramientas para acceder a toda la información disponible sin la sobrecarga que supondría mostrarla toda junta de forma estática.

Los ejemplos de análisis planteados ofrecen una muestra del tipo de análisis que se puede realizar con la representación, haciendo hincapié en la capacidad de transmitir las características estructurales del guion y la información relacionada con los personajes y sus interacciones. Además, las características analizadas se contrastan con parte de la teoría narrativa encontrada en la literatura [8, 9, 10], validando así los resultados obtenidos a través de la representación.

A continuación se muestran algunas líneas de trabajo futuras:

- **Representación temporal:** En la representación planteada se muestran las escenas en el mismo orden en que aparecen en el guion. Sin embargo, esto no siempre coincide con el orden temporal de las escenas en la historia (por ejemplo, en los *flashbacks* u otros saltos temporales). Una evolución interesante sería una representación temporal que se pudiese contrastar con la actual y que ayude a comparar y analizar las estructuras generadas en ambos casos.
- **Análisis de sentimientos:** A través de un análisis de sentimientos en los textos de las escenas, es posible incorporar en la representación información relativa a las emociones dominantes en cada escena, añadiendo información semántica a la información estructural ya existente.
- **Capacidad de edición:** Otra posibilidad interesante es añadir capacidad de edición a la representación, permitiendo modificar el guion a la vez que se realiza el análisis, incluyendo nuevas escenas, modificando las ya existentes, cambiando los personajes que intervienen en ellas, etc.

Bibliografía

- [1] W. Playfair, *The commercial and political atlas: representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure, and debts of England, during the whole of the eighteenth century*. T. Burton, 1768.
- [2] J. H. Lambert, *Mémoire sur quelques propriétés remarquables des quantités transcendentes et logarithmiques, Histoire de l' Académie royale des sciences et belles-lettres*. Histoire de l' Académie royale des sciences et belles-lettres, Berlín, 1761.
- [3] O. Sindiy, K. Litomisky, S. Davidoff, and F. Dekens, "Introduction to information visualization (infovis) techniques for model-based systems engineering," *Procedia Computer Science*, vol. 16, no. 0, pp. 49 – 58, 2013.
- [4] E. R. Tufte, "The visual display of quantitative information," *CT Graphics, Cheshire*, 1983.
- [5] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann Pub, 1999.
- [6] J. H. Larkin and H. A. Simon, "Why a diagram is (sometimes) worth ten thousand words," *Cognitive science*, vol. 11, no. 1, pp. 65–100, 1987.
- [7] Wikipedia, "Guión cinematográfico - wikipedia, la enciclopedia libre." http://es.wikipedia.org/wiki/Guion_cinematografico, 2013.
- [8] D. P. Calvisi, *STORY MAPS: How to Write a GREAT Screenplay*. Act Four Screenplays, 2012.
- [9] C. Vogler, *The Writer's journey*. Michael Wiese Productions, 1998.
- [10] S. Myers, "Character archetypes | go into the story." <http://gointothestory.blcklst.com/2008/08/character-archetypes.html>, 2008.
- [11] F. Brodbeck, "Cinematics." <http://cinematics.fredericbrodbeck.de/>, 2011.
- [12] R. Munroe, "Movie narrative charts."
- [13] V. Ogievetsky, "Plotweaver. cs488b data visualization final project," 2009.
- [14] J. A. P. Agudo, "Herramienta de visualización de la línea temporal en narraciones audiovisuales," 2011.
- [15] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 2, pp. 69–91, 1985.
- [16] R. Theron, "Visual analytics of paleoceanographic conditions," in *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, pp. 19–26, 2006.
- [17] R. Theron and L. Fontanillo, "Diachronic-information visualization in historical dictionaries," *Information Visualization En Prensa*, 2013.
- [18] R. Mazza, *Introduction to information visualization*. Springer, 2009.
- [19] C. Brewer, M. Harrower, A. Woodruff, and D. Heyman, "Colorbrewer 2.0: color advice for cartography," 2009.
- [20] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.

- [21] J. Heer and G. G. Robertson, “Animated transitions in statistical data graphics,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1240–1247, 2007.
- [22] J. Salyer, *Rocky Road: The Hero’s Journey of Rocky Balboa Through the Rocky Anthology*. ProQuest, UMI Dissertation Publishing, 2011.

Mobile Image Recognition with Convolutional Neural Networks

William Raveane and María Angélica González Arrieta

Department of Computer Science, University of Salamanca. Plaza de la Merced s/n.
37008, Salamanca, Spain
{raveane,angelica}@usal.es

Abstract. Convolutional neural networks provide a powerful solution for image recognition, but also present a trade-off due the computing power required to carry out the classification procedure. This is primarily due to the difficulty involved in calculating multiple convolution operations over various layers in the neural network. With the ever increasing performance of mobile devices, it is natural to explore the possibility of implementing such a system on board a device to perform interactive image recognition tasks in real time.

Keywords: Deep Learning, Computer Vision, Image Recognition, Convolutional Neural Networks, Machine Learning, Artificial Intelligence, Biologically Inspired Computing

1. Introduction

Biological visual systems are easily able to recognize objects. However, computer systems still have great difficult performing the same task, primarily due to viewpoint dependent differences that greatly impact the numerical representation of the visual stimulus perceived by these systems. For this reason, computer vision and image recognition continue to be an active research priority in machine learning.

Over time, multiple solutions have been proposed to this problem. One of the most successful of these has been biologically inspired deep networks [1], where input images are processed through several layers of feature extractors and are eventually classified as one out of several target labels. These works are based on the findings by Hubel & Wiesel [2] where two main types of cells were identified in the primary visual cortex of cats, the so called simple cells and complex cells, connected through a cascading hierarchical model. The first functional model to be inspired by these findings was the Neocognitron, introduced by Fukushima [3], wherein two types of cells similarly take on the roles of local feature extractors and deformation toleration.

Later work by Yann LeCun [4] introduced the convolutional neural network (CNN) as a more robust system for the identification of handwritten characters in scanned documents. Again, a similar dual cell type system is used, where a convolutional layer extracts features and a subsampling layer reduces the input

space to tolerate space-wise variations in these features. CNNs form the foundation for most modern image recognition systems based on hierarchical deep learning techniques as they have proven to be a powerful solution when applied to a large variety of vision tasks. Traditionally, these systems are deployed over multiple GPUs to leverage the power of parallelism provided by this kind of hardware[5].

The problem, however, becomes much more challenging when one attempts to fit the demanding algorithms required by this task into a mobile environment where computing capacity is limited. The main focus of this work is an attempt to alleviate some of these issues by deploying a full CNN within a mobile device for real time image recognition. Mobile Image Recognition (MIR) is thus defined as an umbrella term under which to group the various technological components involved in such a system.

The task involves the recognition of 2D planar images within a full 3D environment to provide a solid framework upon which to base the results observed and onto which apply the techniques detailed herein. Such a system would have considerable real world usage applications, and it is quite evident how a mobile device equipped with such capabilities can be exploited in a commercially successful fashion.

As is well known, there currently exist various other systems capable of doing similar tasks. Historically, the needs of these applications has been met by using either simplified targets – as with the 2D markers usually seen in augmented reality applications – or simply by offloading the required computations to a remote server when full image recognition is required, thereby introducing a certain latency in the results.

This work attempts to solve this problem by presenting a method where the recognition occurs in real time within the device itself, to provide immediate feedback to the user and to take advantage of the recognition on a live video feed to increase the responsiveness of the MIR system.

2. Convolutional Neural Networks

This chapter reviews the basic background information forming the basis of neural networks, making an emphasis on the biological concepts behind these mathematical models, and how they can be applied to supervised learning tasks in machine learning.

2.1. Biological Fundamentals of Neural Networks

Neurons in a biological brain are connected to each other through synapses, connections usually formed from the axon of an emitting cell to the dendrites in the receiving cell. The synapses transmit electrical and chemical signals from one neuron to another through a process called synaptic transmission [6].

The connections formed by a synapse between two cells can have either an excitatory or inhibitory effect on the receiving neuron's stimulation. An excitatory synapse will act as previously described increasing the neuron's potential

voltage. An inhibitory synapse, on the other hand, will act in the opposite manner by further lowering the neuron's potential – thereby making it more difficult for other incoming transmissions to overcome it and reach the required threshold value needed to trigger a spike. As there may be both types of synapses present on the receiving end of each cell, complex patterns of excitatory and inhibitory stimulations can result in different behaviors altering the signals this cell itself transmits to others.

This is the basic method by which cells communicate with one another, thus providing a mechanism through which information can be either forwarded or restrained throughout the connected network – depending on the unique behavior governing the actions of individual neurons.

Hubel & Wiesel Visual Cortex Model. The work by Hubel & Wiesel [2] played an important role in understanding the operation of the visual cortex – a specialized section of the brain responsible for interpreting visual stimuli. Of particular interest are the cells in control of orientation selectivity and edge detection of visual information in the V1 visual cortex. Two main types of cells were identified in this area, which have elongated receptive fields and therefore a better response to elongated stimuli – such as lines and edges. These cells received the names of *simple* and *complex* cells.

Simple cells have distinct excitatory and inhibitory regions, both forming elementary elongated patterns in one particular direction, position and size for each cell. If visual stimulation reaches the cell in the exact same orientation and position such that it aligns itself perfectly with the patterns created by the excitatory regions, and at the same time the inhibitory regions do not get stimulated – the neuron gets activated and responds accordingly.

Complex cells operate in a similar manner. Like simple cells, they have a particular orientation they are sensitive to. However, they lack position sensitivity. Therefore, a visual stimulation needs only arrive in the correct angle to activate this neuron, regardless of the exact position in the visual field it falls upon.

Another important fact about the cells in the visual cortex is the structure that they form. Along the cortex hierarchy, proceeding from the V1 region of the visual cortex and proceeding to the V2, V4, and IT regions, one finds that the complexity of the ideal stimuli for each cell increases in complexity. Simultaneously, the activations of neurons become less sensitive to position and size. This happens as cells activate and propagate their own stimuli to other cells they are connected to in this hierarchy, primarily due to alternating simple and complex cell layers.

2.2. Mathematical Model of Convolutional Neural Networks

Traditionally, the architecture of image recognizers has always been divided into two separate functions. The first of these being a fixed feature extractor that transforms input images into a low dimensional representation vector holding characteristics found in the input image. The resulting vector should not

only be easily comparable, but also relatively invariant to transformations and distortions in the input image dataset. Next, a classification module takes this low dimensional vector and interprets it by classifying the data into one of several possible target labels.

The difficulty with such an architecture is the feature extraction stage, which usually requires manual fine-tuning of the parameters to find the features needed which can then be successfully classified. The advantage of CNNs over previous systems, then, is the unification of both procedures into a single neural network system. Using backpropagation, the CNN is able to automatically learn the parameters that will produce the most optimal set of feature extractors, as well as learn the optimal way to classify those same features into a discrete target class set.

A CNN is composed of several specialized layers, the first few of which form the feature extraction stage, and the latter compose a classifier. The feature extraction stage is built with alternating convolutional and subsampling layers.

This alternating structure between neuron types is influenced by the Hubel & Wiesel model of the visual cortex. Convolutional neurons have similar properties to the biological simple cells, in that they are highly sensitive to the edges of an image, in particular to the orientation of edges and the position where they appear in the image. Subsampling layers, on the other hand, are modeled after complex cells, in that the sampling procedure effectively blurs away position sensitivity by bundling together the activation values of neighboring pixels. Finally, the structure in the biological model is mimicked by the CNN as well, in that further layers increase the complexity of the neuron responses while increasing the tolerance for positional invariance.

Convolutional Layer. In the feature extraction stage, the single value neurons forming a traditional multi layer perceptron are replaced by matrix processors which perform an action over the 2D image data passing through them, instead of a single scalar value. The output of each convolutional neurons is given by:

$$Y_j = g \left(b_j + \sum_i K_{ij} \otimes Y_i \right) \quad (1)$$

The output of the calculated neuron Y_j is a matrix holding image data, as are the outputs Y_i of each connected neuron in the preceding layer, all of which are convolved (\otimes operator) with the corresponding kernel filters K_{ij} of each connection. These kernels replace the scalar weights of traditional MLPs. The term for bias for the receiving neuron b_j , and the nonlinear activation function $g(x)$ are identical to the original MLP linear combination equation.

The convolution operator has the effect of filtering input image data with a previously trained kernel. This transforms the data in such a way that certain features, determined by the shape of the kernel, become more dominant by having a higher numerical value assigned to the pixels representing them. The kernels learned by a CNN will usually have intricate shapes and structures

to extract composite features which may have non-trivial purposes within the feature extraction stage of the neural network system.

Max-pooling. In principle, CNNs should have a small amount of tolerance over tiny perturbations in the input image data. For example, if two essentially identical images differ only by the translation of a few pixels, the network should be able to treat both identically, and not expect to learn an entirely new set of feature extractors to account for such a small divergence. This is achieved by reducing the sampling of the output from the convolutional neuron. By having a lower resolution, the same features which would otherwise differ only by a small difference in position, will now be assigned to the same subsampled output position.

Max-pooling effectively finds the maximum value over every consecutively sampled pixel window. As a result, the data size of the input is reduced by a factor equal to the size of the window over which this function operates.

Linear Classifier. The last stage in a CNN is that of a linear classifier, which is exactly identical to the latter stages of a typical MLP.

The input of this classifier is given by the output of the last step in the CNN feature extractor. By the time the data reaches this stage, the input images should have been distilled into a lower size representation of multiple extracted features, having been reduced several times over the course of several convolutional and max-pooling layers. This last stage should now be able to do a final non-linear classification over the feature vector to finally recognize the original input image. Therefore, the output layer of the classifier consists of one neuron for each of the target labels over which the data is being classified.

3. CNN Architecture

This chapter reviews the architecture of the CNN implemented in the MIR system and the reason behind these decisions – not only for practical reasons – but also for optimization and performance enhancement.

3.1. Input Size

The input image size is the first, and possibly most important parameter that must be set in the CNN architecture. It defines the data size over which the network will run, requiring all tested images to first be resized to match these dimensions.

A fine balance must be found when selecting the input data size, the reason being that this setting will affect both the performance and the accuracy of the model. If the size is too small, the model will be fast, but it won't have enough pixel data to make correct decisions on what exactly it is viewing, so classification accuracy will drop. If the size is too big, there will be a lot of data

to work with which will aid the model's accuracy rating but more calculations will be required and performance may be too slow for real time usage.

Table 1 shows a few different values for possible image input sizes. The classification accuracy on the test dataset can be seen in the last column. This provides sufficient evidence that a 32×32 input image size is an optimal value for this particular system by balancing both needs in an appropriate manner.

Table 1. Multiple input size configuration values and its effect on the DCNN performance

| Input Size | Parameters | | | Performance | |
|----------------|--------------|--------------|-------------------|------------------|------------|
| | Kernel Size | Pooling Size | No. of Operations | Computation Time | Mean Error |
| 16×16 | 3×3 | 2 | 110,954 | 3.2 ms | 12.54 % |
| 24×24 | 5×5 | 2 | 481,418 | 8.5 ms | 1.32 % |
| 32×32 | 5×5 | 2 | 1,121,418 | 17.1 ms | 0.44 % |
| 48×48 | 7×7 | 2 | 4,907,882 | 54.9 ms | 0.15 % |
| 64×64 | 7×7 | 4 | 10,038,378 | 147.3 ms | 0.18 % |

3.2. Color Space

Historically, the RGB color space is the preferred color model when dealing with digital imaging. The question arises if there is a better way to represent the image data such as to help with the neural network's learning process, making as efficient use as possible of the given data distribution. As usual, we look to the biological visual system for inspiration on how to better treat this information. It is well known that the human visual cortex is much more sensitive to changes in luminance than to changes in color [7]. Therefore, it would be beneficial to have a system where the intensity can be treated with the attention it demands, while still making use of chromatic data as supporting information.

There exist various color models which follow such a structure, such as the well known YUV color model, where the image intensity is represented in one channel (Y) and the color information is given in two supporting channels (U and V). The YUV method is chosen for the system presented in this work for one particular reason: the MIR device camera.

In accordance to digital video standards, most hardware cameras output image data in YUV format, and it therefore makes perfect sense to make use of this data directly, thus avoiding the computationally expensive step of continuously transforming a full HD video signal frame from one color space to another.

3.3. Contrast Normalization

Prior to processing the image data by the first layer of neurons, the luminosity channel of the image is normalized. The purpose of this is to make a preemptive attempt at levelling out the effects caused by illumination differences.

Traditionally, in image recognition, preprocessing involves one form or another of histogram equalization. For this work, however, contrast normalization is chosen. The main reason being that this system is biologically inspired as well [8, 9] usually outperforming other alternative methods. This process is primarily calculated through image convolution operations, making it somewhat more expensive computationally than other statistics-based algorithms – however, the MIR system can reuse for this preprocessing step the same optimized algorithms that the CNN itself already takes advantage of.

It is important to note that this normalization is applied only to the luminosity channel. The color channels are not modified in any way with prior preprocessing filters, as shifting any of their values would have the effect of remapping image color hues significantly, something which is not desirable when the chromatic value of specific regions have great significance.

The contrast normalization filter is achieved by applying two sets of convolutions on the input data, each with a 1D gaussian kernel, an operation which is considerably less computationally expensive than a full 2D kernel convolution. The contrast normalization filter is performed by the following set of operations:

$$\begin{aligned}
 C^S(I, K) &= I - (K^T \otimes (K \otimes I)) \\
 C^D(I, K) &= \frac{I}{\sqrt{K^T \otimes (K \otimes (I \circ I))}} \\
 C^N(I, K) &= C^D(C^S(I, K), K)
 \end{aligned}$$

Where I is the input image at each operation, usually modified with the 1D gaussian kernel K and its corresponding transposed kernel K^T to convolve in both the horizontal and vertical directions of the image.

The $C^S(I, K)$ function is a subtractive operation, where the dual convolutions have the effect of generating a localized mean representation of the input image, thus, subtracting this mean from the original input image yields a zero-centered normalized image.

The $C^D(I, K)$ function is a divisive operation, where the dual convolutions operate on the input image squared (element-wise matrix multiplication by itself). The square root of this amount represents a localized standard deviation of the input image, thus, dividing the input by its own standard deviation results in a normal distribution with $\sigma = 1$.

The final $C^N(I, K)$ function to implement the contrast normalization operation, then, is that of the previous two operations applied sequentially with the same kernel, resulting in a normalized image with values centered around 0, and a standard deviation of 1 – priming it for a machine learning algorithm to process. The localized convolutions applied throughout the process also have the

effect of flattening gradients over the image, while enhancing the ever important edge outlines describing the object contours, both of which are ideal for easier recognition.

Figure 1 shows the mean classification error over 10 training epochs of two learning runs, one using the contrast normalization preprocessing algorithm, and the other without. It is clear from these results that contrast normalization applied on the Y channel data achieves better recognition performance.

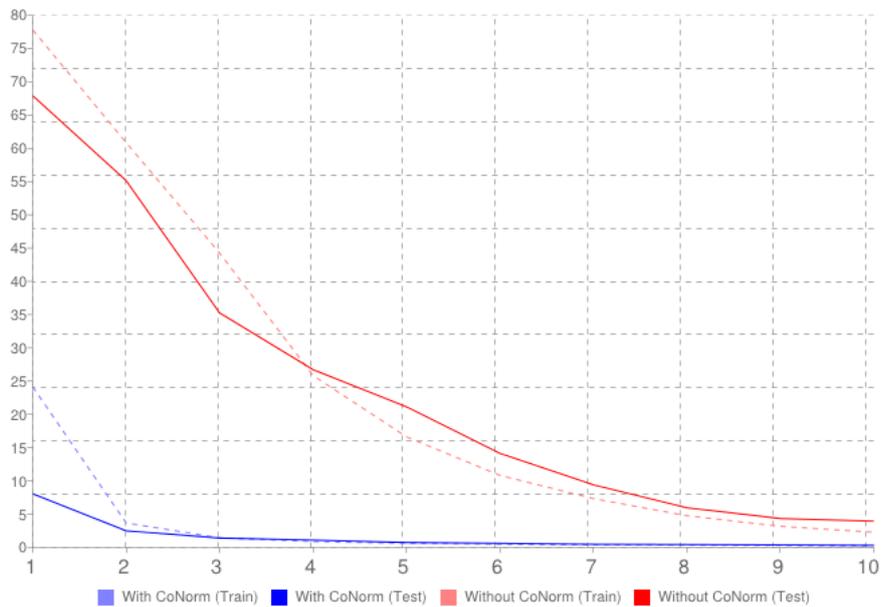


Figure 1. The effect of contrast normalization on the mean error learning curves.

3.4. Connection Mapping

The connectivity of the neural network is very important in that prior domain knowledge can be integrated into the system, and in this manner mold the algorithm to better suit a particular goal. Neurons are not simply fully connected in every possible way to each other, but rather, information bottlenecks can be introduced that force the influence of data to take specific routes within the network, thereby modifying the behavior of individual neurons in a particular manner.

As explained in section 3.2, more importance needs to be assigned to the luminosity channel of the input image than to the color data channels. Structure-wise, this influence is built into the system through a custom connection map

between the first two layers. By doing this, the prior knowledge of asymmetrical channel influence can be embedded into system. A connection map is thus formed between the input neurons and the first convolutional layer in such a way that the Y channel input neuron fully connects to eight convolutional layer neurons, whereas each of the U and V channel input neurons connect to only 2 other convolutional neurons each. Assigning more connections to one neuron greatly increases the effect its stimulus information has over the entire system state. It is important to note here that each channel will be processed separately on this first convolutional layer of the CNN, and it is not until the next stage where the information from all channels is finally mixed together.

The two convolutional stages serve as a form of feature extractor that reduce the input image to a lower dimensional representation. By repeated convolutions and max-pooling operations, the data is reduced from a tensor of size $32 \times 32 \times 3 = 3072$ data points, to one of size $5 \times 5 \times 32 = 800$. These 800 data points serve as the input for the final stage of the CNN, a simple MLP which is able to classify the 800 non-linear features extracted from the input image into the final number of target classes the recognizer is trained upon.

The output layer in the linear classifier has as many neurons as there are classes in the problem to solve, plus one. The extra output neuron corresponds to a catch-all background class.

3.5. Training

The training targets are ideal one-vs-all vectors, where the i -th element of the vector corresponding to the class index takes on a value of +1, while all other elements are assigned a -1 value instead. The (-1,+1) interval, being zero-centered, helps in the training phase by having evenly and symmetrically distributed target values.

The network is trained through Stochastic Gradient Descent, where the configuration parameters are updated after every single training image. This requires the image data set to be previously shuffled so as to avoid any biasing that may occur from training over the same class for too many sequential steps.

4. Implementation and Results

The decision was made to use Torch [10], which is a library that implements many features related to deep neural networks. The MIR system itself is built as a hybrid Java and C++ mobile application for the Android OS. The API provided by the Android Software Development Kit is used for the application's outer UI and the high level logic of the program, while the low level core computations dealing with the CNN implementation are all custom built in C++ and interfaced with the Java program via the Java Native Interface, and compiled with the Android Native Development Kit.

4.1. Artificial Fovea

Traditionally, image recognition systems scan an entire target image iteratively by stepping through all possible positions at different sized scales, thereby searching for a recognizable object in any of the viable extracted regions. Although functional, it is computationally expensive as the full CNN must effectively run at each of these regions. Assuming a search pattern of ten different scales with a conservative 50% window overlap over a complete 1280x720 image frame, a full image scanner would require approximately 900 executions of the CNN.

In a mobile device, where the user has full control over the direction that the camera points in, it is possible to greatly reduce the searchable image space over which the CNN runs by simply requiring the user to aim the camera directly towards the object, a perfectly reasonable demand given the nature of the recognition task. Under these conditions, the MIR system only needs to scan over a series of concentric square regions extracted from the center of the view at several different scales. Again, assuming ten unique scales over which the CNN scans for an object – but without the need of offsetting the scanning position – the device now only needs to run the network over the ten extracted regions, reducing the computational cost by as much as a factor of 90. Figure 2 shows the scanning steps followed by this procedure.



Figure 2. A screen capture of the application showing the artificial fovea overlay, wherein three of the foveal regions have detected one of the images the system has been trained to recognize

This method of operation can be compared to the way the fovea centralis works in the human eye. The fovea is the region of the eye where there is a greater concentration of photoreceptor cells [11]. It is located in the center of the retinal surface and it accounts for the greater visual acuity at the the middle of the eye's field of vision. As the human eye is free to rotate and look around, there only needs to be such a concentration of cone cells at its center, as additional detail can easily be obtained by looking towards a different direction. This mechanism then, allows allocating resources more efficiently where they are truly needed as opposed to having a dense concentration of photoreceptors throughout the eye.

The same idea applies in the artificial fovea proposed here, where the costly CNN computations are concentrated towards the center of the field of vision, as the freedom of movement given by the device diminishes the need for full frame scanning. By reducing the required computational time needed to analyze each image frame, the system can be released sooner and thus be able to scan the camera view with much greater frequency.

Table 2 shows the results of using an artificial fovea in this manner, where the input image fed to the network is classified through a simple winner-takes-all approach over multiple CNNs using the arrangement described above. This run was made over 32,000 image samples, each of which consisting of a target trained-for image, overlaid over a random background at a random scale – in such a way that different images will have different CNNs activated for each of the ten analyzed scales. In this case, the neuron with the highest output over all ten foveal views wins and the full image frame is classified with this output neuron's label.

Table 2. Confusion matrix and global mean error of the CNN, using a winner takes all strategy

| | Classes | | | | | | | | | | Error |
|---------------------------------------|---------|-----|----|----|----|----|-----|-----|----|----|----------------|
| | BG | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | |
| BG | - | 148 | 57 | 36 | 18 | 26 | 102 | 96 | 1 | 53 | 6.781% |
| C1 | 7 | - | 1 | 2 | 1 | 11 | 0 | 7 | 0 | 2 | 0.969% |
| C2 | 10 | 0 | - | 3 | 1 | 0 | 4 | 0 | 1 | 14 | 1.021% |
| C3 | 9 | 26 | 19 | - | 16 | 0 | 15 | 261 | 0 | 41 | 12.084% |
| C4 | 0 | 4 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0.130% |
| C5 | 8 | 6 | 2 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0.500% |
| C6 | 15 | 1 | 5 | 4 | 0 | 3 | - | 0 | 0 | 13 | 1.281% |
| C7 | 20 | 0 | 0 | 71 | 0 | 1 | 3 | - | 0 | 0 | 2.969% |
| C8 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | - | 0 | 0.156% |
| C9 | 0 | 4 | 23 | 0 | 0 | 0 | 3 | 0 | 2 | - | 1.000% |
| Global Misclassification Error | | | | | | | | | | | 3.691 % |

4.2. Committee Algorithm

The practice of using neural network committees starts from the Hypothesis Boosting Problem [12], which when simply stated asks if a set of *weak learners* can form a single *strong learner*, where the strength of a learner is an informal measure of its prediction or classification accuracy. Many algorithms have shown this to be the case and various boosting meta-algorithms have been developed based on this idea.

Implementing such a procedure naively into the MIR system would involve the execution of multiple CNNs for every single viewed input. Unfortunately, this is something that must be avoided as it would obviously increase computational demand linearly by a factor proportional to the number of concurrent networks being executed. The fact that concentric viewing squares are being extracted from the camera view, however, gives the opportunity of reusing the existing knowledge from each of these foveal view outputs as a weak learner itself, and thus it is possible to combine them all into a stronger learner with a higher recognition accuracy than what any of the individual recognizers would be able to achieve.

Unlike regular classifier ensembles, however, a straightforward average operation will not work properly as every network being sampled is looking at a different input scale (different foveal view). As can be seen in figure 2, various networks will recognize their corresponding input differently – and naturally all networks will never concurrently agree on the same output. Therefore, a new mechanism is introduced where the ensemble takes into account the probability that each of the CNNs is viewing a non-recognizable image (background class). This probability is used to weight that network’s influence over the averaging calculation.

This is implemented by first normalizing the output values of each CNN with a softmax activation function, which basically scales the output into a probability-like distribution with all values summing to 1. This activation is defined as follows:

$$\hat{y}_{jn} = \frac{e^{y_{jn}}}{\sum_{i=1}^M e^{y_{in}}}$$

Where y_{jn} is the output value of neuron j for network n , M is the total number of output neurons or classes in a network, and \hat{y}_{jn} is the softmax activated output of this neuron. Note that this function looks at the outputs of all other neurons in the current network to calculate what this neuron’s activated output should be.

Next, a weighted ensemble average over the outputs of all networks is applied. This is defined by:

$$\tilde{y}_j(\alpha) = \sum_{k=1}^N \alpha_{jk} \hat{y}_{jk}$$

Where \hat{y}_j is the averaged value of the output neuron for class j over all networks. The weight α_{jk} is the influence network k has over the average, and this is defined by:

$$\alpha_{jk} = \begin{cases} (\hat{y}_{1k} + \gamma)^{-1} & \text{if } j > 1 \\ 1 & \text{if } j = 1 \end{cases}$$

Here, γ is a small positive number that not only helps to avoid dividing by zero, but also plays a regulatory role by modulating the influence of small BG-class probabilities over the end results. Essentially, this definition of α_{jk} weights the output of class j in network k by the inverse of \hat{y}_{1k} . This \hat{y}_{1k} value is the probability of the output neuron with index 1, that is, the probability that network k has detected a background (or non-recognizable) image. An exception is made for the background class itself, where the averaging weight is left as 1 in order to avoid compounding the effect of this class on itself.

The basic idea behind this is to give greater influence to neural networks that have not defaulted to the background target output in the current foveal view region they are analyzing. Table 3 shows an updated confusion matrix, trained in the same manner as the previous run as in Table 2 above, and over the same data – with the only change being the use of the proposed algorithm for weighted ensemble classification. It is evident that there is a strong improvement by using this type of meta-algorithm to unify the outputs of multiple networks in a more intelligent manner.

Table 3. Confusion matrix and global mean error of the CNN, using the proposed algorithm for weighted ensemble classification

| | Classes | | | | | | | | | | Error |
|---------------------------------------|---------|----|----|----|----|----|----|----|----|----|---------|
| | BG | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | |
| BG | - | 14 | 6 | 16 | 0 | 8 | 15 | 20 | 0 | 3 | 2.562% |
| C1 | 6 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.969% |
| C2 | 2 | 0 | - | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0.125% |
| C3 | 3 | 0 | 0 | - | 0 | 0 | 0 | 19 | 0 | 0 | 0.687% |
| C4 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0.000% |
| C5 | 4 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0.125% |
| C6 | 8 | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0.250% |
| C7 | 21 | 0 | 0 | 6 | 0 | 0 | 0 | - | 0 | 0 | 0.844% |
| C8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 0 | 0.000% |
| C9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 0.000% |
| Global Misclassification Error | | | | | | | | | | | 0.478 % |

One additional benefit provided by this algorithm, in addition to boosting the strength of individual networks, is that the ensemble can work together to reduce the occurrence of false positives. A false positive occurs when there is nothing particular in view, but the network still gets activated due to a random

internal activation that just happens to be triggered by an incorrect stimulus. By taking into account the results of multiple networks looking over different scales, this effect is greatly reduced as the chances of the same random stimulus to happen over various image scales is close to zero.

The end result is a very robust self-correcting system that is capable of recognizing with great accuracy the planar images it has been trained with. A final mean error over tested data of only 0.478% represents a strong result in any machine learning task, let alone image recognition.

5. Conclusions

In this work, a mobile image recognition system based on convolutional neural networks was presented. Many of the steps involved in the development of this application were reviewed in detail, although there were many other implementation issues which for brevity were not mentioned here. A lot of background information was also seen in order to fully understand the operation of this type of neural network, including much of the neuroscience that provides the support upon which this branch of deep learning is built upon.

The effects of the chosen CNN architecture were reviewed, reviewing the choices on colorspace model and normalization algorithm. The meta-algorithm proposed in this work for multiple neural network ensemble averaging over different input scales proved to be quite valuable in strengthening the classification performance of the overall system. It was seen how it boosts the performance of individual networks, but perhaps the most important contribution is the reduction of false positives in the detection.

The final performance obtained by the system when deployed on a mobile device was quite surprising. Although a simple use of the system was reviewed here, the fast response of the system opens up the door to a large amount of other applications on which this system can be used, such as augmented reality and possibly even motion analysis applications.

Bibliography

- [1] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *12th International Conference on Computer Vision*, pp. 2146–2153, IEEE, October 2009.
- [2] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *The Journal of Physiology*, no. 148, pp. 574–591, 1959.
- [3] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 93–202, 1980.
- [4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, IEEE, November 1998.
- [5] D. C. Ciregan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, no. 32, pp. 333–338, 2012.
- [6] J. H. Byrne, "Neuroscience online, <http://neuroscience.uth.tmc.edu/>," 1997.
- [7] M. Livingstone, *Vision and Art: The Biology of Seeing*. Abrams, 2008.
- [8] N. Pinto, D. D. Cox, and J. J. Dicarlo, "Why is real-world visual object recognition hard?," *PLoS Computational Biology*, vol. 4, no. 1, 2008.
- [9] S. Lyu and E. Simoncelli, "Nonlinear image representation using divisive normalization," in *Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, June 2008.
- [10] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," *BigLearn, NIPS Workshop*, 2011.
- [11] H. Kolb, R. Nelson, E. Fernandez, and B. Jones, "Webvision: The organization of the retina and visual system," 2013.
- [12] M. Kearns, "Thoughts on hypothesis boosting." 1988.

Sistema Multiagente Basado en Organizaciones para el Control de Vídeo Multidispositivo

Juan Agustín Rodríguez Valle¹ y Juan Manuel Corchado¹

¹Departamento de Informática y Automática, Universidad de Salamanca. Plaza de la Merced s/n. 37008, Salamanca, España
{jrodriguez, corchado}@usal.es

Resumen La visión artificial está siendo actualmente objeto de un intenso estudio de investigación. El objetivo de los investigadores es principalmente conseguir un sistema funcional capaz de hacer un análisis de las imágenes obtenidas lo más preciso posible en tiempo real.

Este trabajo tiene dos metas principales: por un lado, el estudio de las técnicas para la adquisición y tratamiento, en tiempo de ejecución, de imágenes procedentes de cámaras web; y por otro, el desarrollo del sistema de análisis de dicha información con tecnología multiagente.

Como se expondrá a lo largo de este documento, los algoritmos de visión son muy costosos y toman tiempos de respuesta muy altos, lo que en nuestro caso es un gran inconveniente si consideramos que muchas de las aplicaciones pueden requerir operación en tiempo real. La arquitectura de agentes pretende automatizar el proceso de análisis de las imágenes obtenidas por las cámaras y optimizar su procesamiento, para reducir el tiempo de ejecución de los procesos costosos.

Keywords: Visión artificial, SMA, Multiagentes, Reconocimiento de personas

1. Introducción

La gran evolución de la tecnología hace que los sistemas que hay en el mercado cada vez sean mejores y más capaces de facilitar la vida de las personas. Con la alta velocidad de internet y el alto avance de las cámaras web se puede visualizar en tiempo real lo que está pasando al otro lado de la cámara.

Esto ha supuesto que empresas de seguridad o de marketing quieran utilizar esta tecnología para hacer que su trabajo sea mucho más sencillo y amplio que sin estos sistemas, para evitar, por ejemplo, accidentes como el que surgió en el Madrid Arena el pasado 1 de Noviembre.

El problema de la visión artificial, ha recibido una considerable atención durante muchos años desde el punto de vista psicofísico, y en los últimos años en la comunidad científica preocupada por el estudio de la visión artificial. Se ha producido un considerable aumento de interés en los temas de procesado de imagen, lo que ha conllevado la necesidad y aparición de nuevos métodos teóricos y desarrollos prácticos para el diseño de este tipo de sistemas. Las aplicaciones del

procesado de imagen son muy variadas, e incluyen aspectos como la medición remota, el análisis de imágenes biomédicas, la simulación de cirugía guiada remota, el reconocimiento de caracteres, aplicaciones de realidad virtual y realidad aumentada en sistemas colaborativos, entre otros.

Un análisis rápido de las publicaciones recientes revela que una gran cantidad de contribuciones tratan el problema del análisis de imágenes y escenas. El objetivo general del procesado de imagen es analizar imágenes de una determinada escena y reconocer su contenido, para poder tomar decisiones de acuerdo con el mismo. Muchos tipos de escenas son esencialmente bidimensionales, como el reconocimiento de caracteres, o el tratamiento de la mayoría de imágenes biomédicas, pero en otras situaciones, las escenas a describir o interpretar son fundamentalmente tridimensionales.

Se presenta en este trabajo un sistema diseñado con el objetivo de mejorar la eficiencia de la seguridad en los recintos cerrados. El sistema interactúa con los usuarios y el entorno a través de un conjunto de agentes inteligentes incorporando características propias de las técnicas utilizadas en visión artificial, para proporcionar una monitorización en tiempo real de los usuarios. El sistema facilita un control y supervisión preciso del número de personas que están presentes, ya que este trabaja en tiempo real.

2. Objetivos

El objetivo principal de este trabajo es diseñar un sistema multiagente que implemente algoritmos para la descomposición de las características de una imagen con el objetivo de detectar personas que entran y salen de una o varias habitaciones. Las principales tareas que se persiguen son:

- Creación de un sistema capaz de contar personas que entran y salen de una habitación.
- Obtención de imágenes a partir de una cámara web.
- Investigación de las diferentes técnicas de procesado de imagen, detección del movimiento y de seguimiento de personas y seleccionar la mejor entre ellas.
- Creación de un sistema de agentes capaces de comunicarse entre ellos y así compartir y distribuirse la información y las tareas para el análisis de imágenes capturadas por una cámara web.
- Análisis de los distintos tipos de sensores que se utilizan en estos sistemas viendo las ventajas y desventajas con la cámara web que se utilizará para la realización del sistema.
- El sistema de agentes será un sistema coordinado en el cual los agentes se comuniquen entre ellos, compartiendo determinada información. Estos agentes realizarán funciones como:
 - Sumar el conteo en distintas puertas.
 - Enviar un mensaje de alerta al móvil cuando se sobrepase un determinado umbral de personas.

- Permitir la visualización y la modificación de la información solo a las personas autorizadas.

El principal motivo de la elección de una cámara web para la utilización en el sistema ha sido su bajo coste, ya que los distintos tipos de aplicaciones que existen en el mercado utilizan cámaras estereoscópicas encareciendo la utilización del mismo.

Se ha decidido utilizar la arquitectura Pangea [1] [2] [3] [4], ya que este sistema posee un protocolo de comunicación con el cual se podrá utilizar cualquier tipo de dispositivo en un futuro si fuese necesario.

3. Arquitectura del sistema

En el presente sistema se desea implementar una arquitectura distribuida basada en agentes, los cuales permitan procesar la información visual contenida de una imagen que ha sido anteriormente obtenida por una cámara web.

Este sistema se implementará en la arquitectura Pangea. Todos los agentes creados para el sistema tendrán que adaptarse a la arquitectura, que más tarde conformarán el conjunto de agentes nombrado.

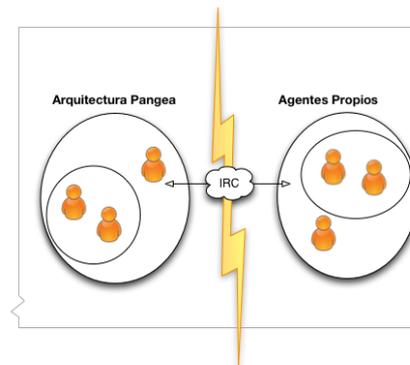


Figura 1. Integración con Pangea

La comunicación con la plataforma Pangea se realizará mediante el protocolo IRC [5]. Es un protocolo de tiempo real basado en texto plano.

La sencillez de este protocolo y la corta carga de datos que genera en las comunicaciones hacen de éste un protocolo rápido y eficaz. Como la comunicación es realizada mediante éste protocolo, el cual utiliza socket para la comunicación, se pueden desarrollar los agentes en cualquier lenguaje de programación, haciendo más fácil su implementación.

Como se aprecia en la figura 1, una de las características es que, al no estar restringidas las aplicaciones por un determinado lenguaje de programación, debido al protocolo de comunicación, se amplía el abanico de posibilidades para desarrollar interfaces sobre un mayor número de dispositivos y sistemas. La estructura de componentes es modular, siendo los agentes quienes gestionan todas las funcionalidades y comunicaciones, tanto de los sistemas como de la propia arquitectura.

El sistema está compuesto por un conjunto de agentes, con roles bien definidos que comparten información y servicios. El análisis de imágenes implica un proceso complejo donde cada agente ejecuta su tarea con la información disponible en cada momento. En la figura 2 se muestra un esquema de la arquitectura multiagente propuesta.

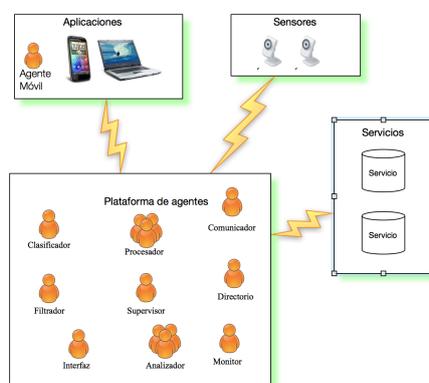


Figura 2. Esquema de la arquitectura

Como se muestra en la figura 2 la arquitectura presentada está compuesta por 4 capas:

- Las **aplicaciones** consisten en todos aquellos programas que pueden ser utilizados para acceder a las funcionalidades del sistema. Las aplicaciones son dinámicas y adaptables al contexto, reaccionando de forma distinta ante determinadas situaciones y el tipo de servicio solicitado. Las aplicaciones pueden ser ejecutadas localmente o de forma remota, incluso desde dispositivos móviles con limitada capacidad de procesamiento, debido a que las tareas de cómputo se delegan a los agentes y a los servicios.
- Los **servicios** representan el grueso de las funcionalidades del sistema a nivel de procesamiento, entrega y adquisición de información. Los servicios pueden definirse como conjuntos de actividades que pretenden responder a las necesidades de quien los invoca de forma local o incluso de forma remota. Se pretende que el sistema cuente con un directorio flexible de servicios, de

manera que puedan ser modificados, añadidos o eliminados dinámicamente y bajo demanda. En esta capa también se ha de introducir una base de datos para almacenar la información que después es utilizada por los distintos agentes. Un servicio puede ser prácticamente cualquier programa que ejecute una determinada tarea y que además comparta sus recursos con la plataforma. Estos programas proporcionan métodos para analizar datos, obtener información de dispositivos externos (sensores), publicar información, o incluso hacer uso de otros servicios para ejecutar una determinada tarea. Los desarrolladores tienen la libertad de utilizar cualquier lenguaje de programación, con el único requerimiento de seguir el protocolo de comunicación definido en la arquitectura.

- La capa de **agentes**. Es un servidor que contiene una organización de agentes, y es donde todos ellos se comunican para llevar a cabo las tareas o servicios.
- Capa de **sensores**: es una capa física de sensores instalados en un edificio que se comunican con la Capa de Agentes de Sensores por medio del protocolo IRC. En ella se encuentran todos los sensores de la aplicación, en este caso cámaras web, pero puede ser ampliable a otros dispositivos, ya que el protocolo utilizado permite la comunicación con múltiples dispositivos siempre y cuando se respete el protocolo.

El funcionamiento del sistema multiagente conlleva la coordinación de agentes que cooperan y comparten entre sí la información necesaria durante el proceso de análisis. Al descomponer el proceso de análisis descrito en los anteriores apartados, las tareas generadas se distribuyen entre cada uno de los roles del sistema. Se han obtenido los siguientes agentes:

- Clasificador: encargado de recibir las imágenes, comprobar su formato y cambiarlo en el caso de que sea necesario.
- Filtrador: encargado de las tareas de filtrado básico. Coordina los filtrados de las primeras fases ya sea desde las aplicaciones incrustadas en las cámaras o desde el propio sistema de análisis.
- Procesador: encargado de la extracción de características. Podrá hacer uso del servicio de extracción que crea conveniente (Canny, etc.)
- Analizador: Es el encargado de detectar las personas.
- Móvil: Es el encargado de enviar las notificaciones a las distintas aplicaciones en caso de necesitarlo.
- Sensor. Es el encargado de detectar si hay algún movimiento en la imagen, para así enviarla al sistema.
- Monitor: se encarga del control de los procesos. Asigna tareas a los demás agentes. Divide en bloques de las imágenes y asigna las tareas de análisis a los agentes analizadores. Los agentes analizadores le devolverán sus resultados y éste enviará la información concentrada al agente Reconstructor que se encargará de hacer que se represente tridimensionalmente.
- Comunicador: responsable de las comunicaciones entre las aplicaciones y la plataforma y los servicios y la plataforma. Maneja las peticiones de entrada desde las aplicaciones para ser procesadas por los servicios.

- Supervisor: analiza la estructura y sintaxis de todos los mensajes de entrada y salida del sistema y supervisa el correcto funcionamiento de los otros agentes en el sistema (verifica periódicamente el estado de todos los agentes registrados en la arquitectura enviando mensajes ping).
- Roles: Define los permisos que tienen los usuarios, así como identificar los agentes no válidos que se conecten a la arquitectura.
- Directorio: llevará un listado activo de todos los servicios que ofrece la arquitectura (locales o remotos).

Aplicando la metodología Gaia [6], se ha obtenido un modelo de roles de agentes como muestra la figura 3. Estos roles pueden ser:

- CLASIFICADOR. Encargado comprobar el formato de las imágenes y cambiarlo en el caso de que sea necesario.
- PROCESAIMAGEN. Se aplicará los distintos tipos de algoritmos para procesar la imagen.
- ANALIZADOR. Analizará los mensajes entre la plataforma, así como las imágenes que se obtienen.
- COMUNICADOR. Mantendrá una comunicación entre los agentes de la plataforma internos como externos.
- MONITOR. Se encarga del control de los procesos. Asigna tareas a los demás agentes.
- ALMACENADOR. Llevará el acceso a la base de datos.
- TRACKING. Detecta movimiento y lo envía al sistema.

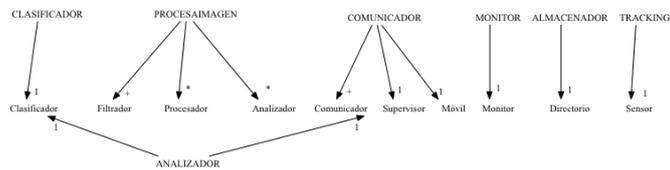


Figura 3. Modelo de roles

Otro modelo aplicando la metodología es el que muestra la figura 4. Este modelo representa la comunicación entre los agentes y permite visualizar el flujo de comunicación entre ellos, además de detectar posibles puntos de congestión antes de la implementación. Como resultado de la interacción entre los roles, se definen los protocolos necesarios, así como los roles que toman parte de dicha interacción.

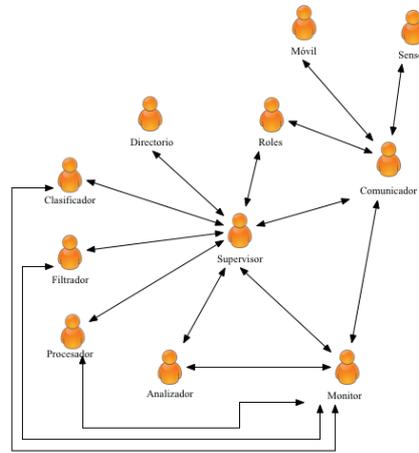


Figura 4. Modelo de interacción

4. Servicios ofrecidos

Dentro de los servicios, podemos clasificar estos en 2 categorías, los controlados por el usuario dentro de un dispositivo y los servicios brindados por comandos dentro de la plataforma de agentes, que incluye a los primeros.

En la Tabla 1 se pueden ver la lista de algunos de los servicios a los que pueden acceder las aplicaciones del sistema.

Tabla 1. Servicios desde las aplicaciones

| Nombre del servicio | Descripción |
|-----------------------------------|--|
| NúmeroPersonas([habitación]) | Número de personas que hay en una determinada habitación. |
| EntradoPersonas([habitación]) | Número de personas que han entrado en una determinada habitación. |
| SalidoPersonas([habitación]) | Número de personas que han salido de una determinada habitación. |
| VerCámara(habitación) | Permite visualizar la cámara de una determinada habitación. |
| ModificaUmbral(umbral,habitación) | Definir número de personas máximas de una habitación |
| NuevosAgentes() | Devuelve una lista de los agentes nuevos que hay en la cola de espera. |
| ConfirmarAgente(NombreAgente) | Aceptar agente en la plataforma. |

Todos los agentes realizan una tarea específica y esta puede ser finalizada por ellos mismos o con ayuda de otro agente. Algunos agentes necesitan de otros para finalizar su requerimiento, por lo que se creó una serie de comandos los cuales proveen comunicación entre ellos para lograr los objetivos previamente planteados por cada Agente.

Tabla 2. Servicios desde las aplicaciones

| | |
|--|--|
| ListarServicios() | Devuelve un listado de los servicios disponibles y su manera de uso. |
| ListarAgentes() | Devuelve un listado de los agentes disponibles en el sistema. |
| EstadoAgente(NombreAgente) | Devuelve el estado actual del agente especificado. |
| ActivarAgente(NombreAgente) | Activa el agente determinado y lo conecta a la plataforma de agentes. |
| EstadoSensor(NombreAgente) | Devuelve el estado actual del sensor asociado al agente especificado. |
| ActivarSensor(NombreAgente) | Enciende el sensor asociado al agente si este se encuentra apagado. |
| DesactivarSensor(NombreAgente) | Apaga el sensor asociado al agente si este se encuentra apagado. |
| AgregarTarea(NombreTarea, NombreAgente, [Fecha]) | Crea una tarea específica para un agente determinado a una hora. |
| EliminarTarea(NombreTarea) | Elimina una tarea en caso de que exista dentro de la cola. |
| ListarTareas([NombreTarea]) | Lista las tareas pendientes. |
| UsuariosPermitidos() | Devuelve una lista de usuarios a los cuales se les permite conectarse. |
| RolUsuario(NombreAgente,Rol) | Cambia el rol de un determinado usuario. |
| AddUsuario(NombreAgente,Rol) | Añade un usuario a la plataforma con un determinado rol. |

Tras el estudio de los diferentes tipos de algoritmos de análisis, procesamiento y tratamiento de imágenes, en este proyecto se han utilizado los filtros de suavizamiento y realce [7], el algoritmo Canny [8] para la extracción de las características de la imagen y el algoritmo de Hog para detectar personas y para la detección de movimiento se utilizan métodos recursivos.

El algoritmo de detección de movimiento y el de Hog [9] [10] [11] se ha desarrollado bajo openCV utilizando la librería Egmu para .NET, ya que implementa las características de openCV para poder utilizarlas en el lenguaje de programación C Sharp.

5. Resultados y conclusiones

El objetivo fundamental de este proyecto era crear una arquitectura de sistema que brindara servicios de control de video vigilancia que pudieran ser utilizados remotamente, esto con el fin de instalar en un edificio o casa particular, cámaras con el fin de poder saber el número de personas que existen dentro.

En la gestión y control de estos sensores se eligió realizarlo a través de una arquitectura distribuida de agentes que lograran procesar la información de ellos. Conforme a esto se crearon 3 capas (aparte de la capa física de sensores), una para la gestión de los sensores como tal, una para la gestión del sistema multiagentes y otra de presentación, desarrollada para dispositivos, cuya finalidad era poder ver los sensores externamente, así como las estadísticas ofrecidas por el sistema.

A continuación se muestran las pantallas que forman la aplicación en los dispositivos que muestran la información al usuario. Esta aplicación constará de tres pantallas y una última que mostrará determinadas alertas:

- La pantalla de login que sirve para que un agente se conecte a la plataforma. Si el agente no estuviese en la plataforma se notificaría al usuario de que hasta que no le den de alta en la plataforma no podrá acceder a la aplicación.

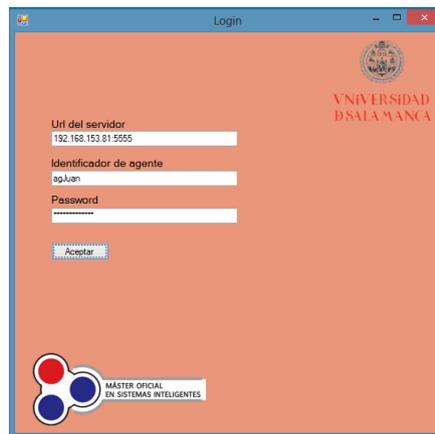


Figura 5. Pantalla de login

- Una vez que se entra en la aplicación el usuario verá otra pantalla en la cual podrá seleccionar alguna de las cámaras que disponga la casa. Tras seleccionarla le saldrá el video en vivo de la cámara. Si el agente dispone de los suficientes permisos aparecerá un botón para acceder a la pantalla de opciones.



Figura 6. Pantalla de principal

- La pantalla de opciones muestra la información por habitación del conteo de personas, tanto las que entran como las que salen como las que se encuentran dentro. Si el usuario tiene permisos suficientes aparecerá un control para establecer el umbral de personas en esa habitación.



Figura 7. Pantalla de opciones

- La pantalla notificación aparecerá cuando se sobre pase el umbral definido para una habitación notificando de ello y pudiendo acceder a la cámara o en su defecto ignorar la advertencia.

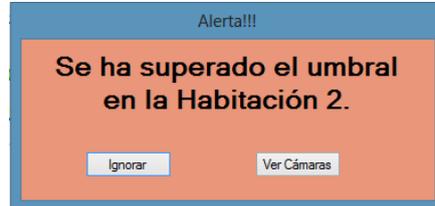


Figura 8. Pantalla de alerta

En la arquitectura multiagente se desarrollaron una serie de servicios ofrecidos por los distintos tipos de agentes. Pero la plataforma provee otra cantidad de servicios de gestión como obtener la imagen procesada o enviar notificación, que pueden ser utilizados internamente, o extendidos para ser utilizados externamente. Se podría ver la detección de la imagen devolviendo este servicio o una utilización del sistema de notificaciones para un posible sistema de avisos entre usuarios. Se utilizó un lenguaje procedural a través de una serie de comandos con los cuales se invocan a estos servicios.

Además, se definieron una serie de roles que la plataforma debía de tener, y conforme a esto se crearon distintos tipos de agentes. Esto con el fin de proveer a la plataforma con una mejor gestión y control del proceso.

Para darle un control más autónomo a la plataforma, se implementó un servicio especial de Planificación, en el que un usuario puede establecer horarios de ejecución de tareas, y posterior a esto, el control total de la plataforma es administrado entre los Agentes de Planificación, Coordinación y Sensores, sin intervención del usuario.

Al final, se obtuvo un sistema inteligente, capaz de utilizar los sensores instalados en él sin la supervisión directa de un usuario, los cuales están conectados a través de un agente a una plataforma central de agentes, y a la vez, están controlados automáticamente por un Agente.

Se pudo observar que la calidad del servicio obtenido por los agentes era de un 88 % de precisión. Esto era así por los errores que se cometían al interceptar cambios muy bruscos en la intensidad de la imagen en las cámaras.

5.1. Líneas de trabajo futuro

Después del desarrollo de la arquitectura de agentes los cuales son capaces de proporcionar unas estadísticas del conteo de personas en una habitación a partir de cámaras web, podemos diferenciar como líneas principales de trabajo futuro:

- Los agentes en los dispositivos puedan controlar las cámaras cambiando su configuración o su posición con respecto a la actual.
- Desarrollo de técnicas de paralelización automática de los algoritmos para mejorar la optimización del sistema.

- Utilizar otros dispositivos, como cámaras estereoscópicas, para mejorar la calidad del sistema.

Como apunte de futuro final decir que la información del color aporta una fuente de importante peso. Los resultados de todos los métodos mencionados en las secciones anteriores pueden mejorarse considerablemente añadiendo información del color. La calidad de los resultados del algoritmo basado en bordes puede mejorarse grandemente usando imágenes en color.

Bibliografía

- [1] L. Búrdalo, A. Terrasa, V. Julián, C. Zato, S. Rodríguez, J. Bajo, and J. M. Corchado, “Improving the tracing system in PANGAEA using the TRAMMAS model,” in *Advances in Artificial Intelligence-IBERAMIA 2012*, pp. 422–431, Springer, 2012.
- [2] C. Zato, A. Sánchez, G. Villarrubia, J. Bajo, S. Rodríguez, and J. F. De Paz, “Personalization of the workplace through a proximity detection system using user’s profiles,” in *7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing*, pp. 505–513, 2013.
- [3] C. Zato, A. Sanchez, G. Villarrubia, S. Rodriguez, J. M. Corchado, and J. Bajo, “Platform for building large-scale agent-based systems,” in *Evolving and Adaptive Intelligent Systems (EAIS), 2012 IEEE Conference on*, pp. 69–73, 2012.
- [4] J. M. Corchado, J. Bajo, Y. De Paz, and D. I. Tapia, “Intelligent environment for monitoring alzheimer patients, agent technology for health care,” *Decision Support Systems*, vol. 44, no. 2, pp. 382–396, 2008.
- [5] B. Jähne, *Computer Vision and Applications*. Academic Press.
- [6] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.
- [7] S. McHugh, “Tutorials – digital camera sensors,” 2007.
- [8] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679–698, Nov. 1986.
- [9] N. Dalal, *Finding People in Images and Videos*. PhD thesis, 2006.
- [10] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. vol. 1, pp. 886 – 893, 2005.
- [11] T. University of Iowa, “Digital image processing. image analysis and understanding.” <http://www.icaen.uiowa.edu/dip/LECTURE/contents.html>.

Intelligent Models and Simulation to Optimise Industrial Processes

Roberto Vega Ruiz¹, Héctor Quintián Pardo¹ and Emilio Corchado Rodríguez¹

¹Department of Computer Science, University of Salamanca. Plaza de los Caídos s/n.
37008, Salamanca, Spain

{rvegaru,hector.quintian,escorchado}@usal.es

Abstract. This study presents a novel Intelligent System based on the application of Soft Computing Models and Identification Systems, which makes possible to optimise industrial processes, such as a dental milling process. The main goal is the optimization of some parameters as time. This novel intelligent procedure is based on two steps. Firstly, statistical techniques and unsupervised learning are used to analyse the internal structure of the data set and the identification of the most relevant variables. Secondly, the most those variables are used to model the system using supervised learning such as artificial neural networks and support vector machines. The model has been successfully tested using an industrial real data set.

Keywords: Soft Computing, Identification Systems, Support Vector Machine, Artificial Neural Networks, Industrial Processes, Milling Dental Process

1. Introduction

Soft Computing [1, 2] has been successfully used for industrial process modelling. Recently, different paradigms of Soft Computing have been applied to different industrial problems [3]. System identification [1, 5] has made possible to model, simulate and predict the behavior of many industrial applications successfully and in different areas, such as control [6], robotics [8], energy processes [8], milling machine [9], high precision [10], etc. A novel and economically advantageous application is the optimization process in the area of Odontostomatology. Improved processing and optimization of parameters, such as time processing, for the development of dental pieces are the focus of rigorous studies today.

The optimization process of machine parameters, such as the time parameter and accuracy, permits significant saving money due to the high number of dental pieces produced daily by the same high precision dental milling machine centre.

One way to achieve this optimization is based in the use of emerging methodologies, such as neural networks [11], nature-inspired smart systems[9], fuzzy logic [10], support vector machine [11], data mining [15], visualization [16], genetic algorithms [17], case-based reasoning [18], among others.

This contribution is organised as follows. Section 2 describes the two steps intelligent model for analysing and modelling the dataset. Section 3 introduces the industrial case study. Section 4 describes the experiments and the results obtained. Finally, the conclusions are set out and some comments on future research lines are presented.

2. Two Steps Model

A Two Steps Model (Fig. 1) is presented in this contribution. It is based on the analysis of the Internal Structure of a data set and the identification of the most relevant variables followed by the application of System Modelling to optimize the process.

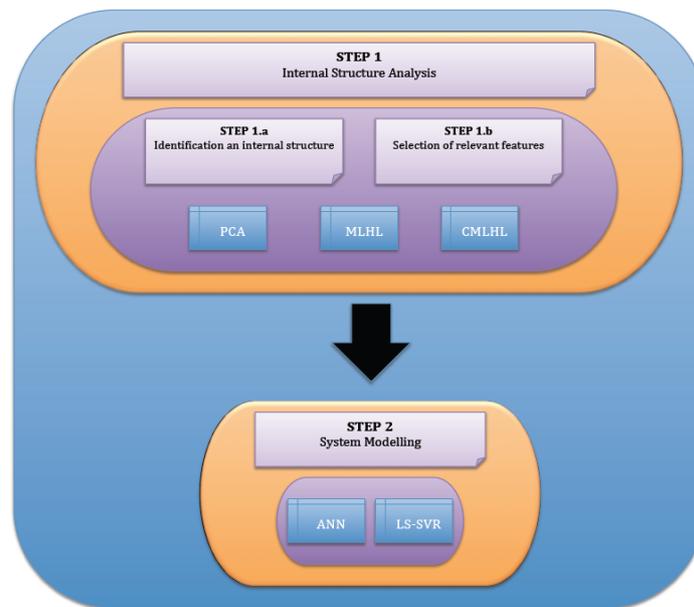


Figure 1. Two Intelligent Steps Model diagram

Firstly, the analysis of internal structure is done by using statistical methods as Principal Components Analysis (PCA) [19, 20] and unsupervised learning based models as Maximum Likelihood Hebbian Learning (MLHL) [5] and Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [22]. These models are also used to identify the most significant features.

Finally, Identification Systems is applied through Artificial Neural Networks (ANN) [23] and Support Vector Machine (SVM) [24] to create the best model of an industrial process.

This model has been built under a visual tool called DataEye. It has been developed in PHP, JavaScript and HTML5. This tool is accesible from any device (Fig. 2) such as Pc, Smartphone and Tablet, having internet connection.



Figure 2. DataEye is accesible from any device

2.1. Internal Structure Analysis

To begin with, PCA, MLHL and CMLHL are applied to analyze the internal structure of the data set from a case study. If it is possible to identify an internal structure, it means that the data set is informative and valid, otherwise it must be expanded with more data.

Principal Component Analysis (PCA)

PCA [19, 20] is a statistical model which describes the variation in a set of multivariate data in terms of a set of uncorrelated variables, each of which is a linear combination of the original variables. The main goal is to derive new variables in decreasing order of importance, with the less loss of information. Applying PCA, it is possible to find a smaller group of underlying variables that describe the data. PCA has been the most frequently reported linear operation involving unsupervised learning for data compression.

Maximum Likelihood Hebian Learning (MLHL)

MLHL [19] is a neuronal implementation of Exploratory Projection Pursuit

(EPP) [25] which is a recent statistical method aimed at solving the difficult problem for identifying structure in high dimensionality data. However, not all projections will reveal this structure well.

MLHL identifies interestingness by maximizing the probability of the residuals under specific probability density functions that are non-Gaussian.

Let have an N -dimensional input vector, x , and M -dimensional output vector, y , with W_{ij} being the weight linking the j^{th} output. The learning rate, η is a small value which will be annealed to zero over the course of training the network and p is a parameter related to the energy function. The activation passing from to output through the weights is described by Eq. 2.

MLHL can be expressed [5] as:

Weight change:

$$\Delta W_{ij} = \eta y_i \text{sign}(e_j) |e_j|^{p-1} \quad (1)$$

Feed-forward step:

$$Y_i = \sum_{j=1}^N W_{ij} x_j, \forall i \quad (2)$$

Feedback step:

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i, \forall j \quad (3)$$

Cooperative Maximum Likelihood Hebbian Learning (CMLHL)

CMLHL [22] is based on MLHL adding lateral connections.

CMLHL can be expressed [22] as:

Weight change:

$$\Delta W_{ij} = \eta y_i \text{sign}(e_j) |e_j|^{p-1} \quad (4)$$

Feed-forward step:

$$Y_i = \sum_{j=1}^N W_{ij} x_j, \forall i \quad (5)$$

Feedback step:

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i, \forall j \quad (6)$$

Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \quad (7)$$

Where: η is the learning rate, $[]^+$ is a rectification necessary to ensure that the y -values remain within the positive quadrant, τ is the "strength" of the lateral connections, b is the bias parameter, p is a parameter related to the energy function and A is the symmetric matrix used to modify the response to the data. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.2. System modelling using identification algorithms

System Identification [1, 5] procedure comprises several steps: the selection of the models and their structures, the learning methods, the identification and optimization criteria and the validation method.

Validation ensures that the selected model meets the necessary conditions for estimation and prediction.

A good model is one that makes good predictions and it produces small errors [1], for this reason Artificial Neural Networks (ANN) [11] as Multilayer Perceptron (MLP) [26] and Least Square Support Vector Regression (LS-SVR) [11] are applied in this study to get the best model.

Artificial Neural Networks (ANN)

ANN [11, 26] are one of the most interesting soft computing paradigms. It is used also under the frame of System Identification. When ANN is used, the purpose of an identification process is to determine the weight matrix based on the observations Z^t , so as to obtain the relationships between the network nodes.

Least Square Support Vector Regression (LS-SVR)

LS-SVR [11] is a modification of the algorithm of the Support Vector Machine (SVM) [27]. In SVR the main idea is to map the data into a high dimensional feature space F via a nonlinear mapping and to perform a linear regression in this space.

The application of LS-SVM for regression purposes is known as Least Square Support Vector Regression (LS-SVR).

In LS-SVR, only 2 parameters (γ, σ^2) are needed. Where σ^2 is the width of the used kernel and γ is a weight vector.

3. Optimising a Dental Milling Process

Advancements in dental technology can offer modern and innovative solutions to traditional dental problems. The driving force behind technological advances is a desire to provide with leading edge dental treatment that can be performed in a more efficient, effective, comfortable manner. The development of dental technology is advancing with the application of computer science in this field.

The aim of manufacturing a tooth prosthesis is creating an impression of the patient's mouth. To do this, the dentist jams silicone putty into the patient's mouth. The dentist removes the negative mold and fills it with plaster. When mold is dried, this provides an exact replica of the patient's dentition. Then, the 3-D model of the dentition is transferred to a CAM system, where an operator generates toolpaths for high-speed, multi-axis machining of the final workpiece.

The industrial case scenario is based on the real data gathered by means of a Machining Milling Center of HERMLE type-C 20 U (iTNC 530) (Fig. 3), with swiveling rotary (280 mm), with 5-axis, with a control system using high

precision drills and bits, by optimizing the time error detection for manufacturing dental metal.

The material (Cr-Co, Ti), the tools and the feed rates affect the conformation times. Since the processing involves polishing and milling the piece, the time of conformation determines its size.

The milling process in the preparation of dental pieces is currently the most modern processing prosthesis in existence. When it finished, the completed crown, bridge or veneer is cut away from the blank, hand-finished to remove coping and runner marks, polished and attached to the patient's damaged tooth using dental cement or affixed to a previously installed implant.

The advantages of 5-axis machines are numerous allowing complete multilateral machining in a single cycle, reducing non-productive time, eliminating the lack of precision arising from the multiple ties of the piece and allowing better access to restricted areas difficult to reach. The angle adjustment can be freely defined and it is possible to use shorter and more rigid tools, which results in improved surface finish.

The main parameter to estimate and optimize in this research is the time error for manufacturing. It is the difference between the estimated time by the machine itself and real work time. Negative values indicate that real time exceeds estimated time.



(a) Milling Center of HERMLE type-C 20 U (b) Metal pieces manufactured

Figure 3. HERMLE highprecision machine (a) and pieces manufactured process(b)

3.1. Data Set

The real case study is described by an initial data set of 190 samples obtained by the dental scanner in the manufacturing of dental pieces with different tool

types (plane, toric, spherical, and drill) characterized by 12 input variables (see Table 1). The input variables are the type of work, the thickness, the number of pieces, the radius of the tool, the tool, the revolutions per minute of the drill, the feed rate (X, Y and Z), the initial tool diameter, the initial temperature and the estimated duration of the work.

Table 1. Variables and range of values

| Variables | Values |
|--|---|
| Work | 1 to 7 |
| Disk thickness (mm) | 8 to 18 |
| Pieces | 1 to 4 |
| Radius (mm) | 0.25 to 1.5 |
| Type of tool | Plane (1), Toric (2), Spherical (3), Drill (4) |
| Revolutions per minute (RPM) | 7,500 to 38,000 |
| Feed X (mm) | 0 to 3000 |
| Feed Y (mm) | 0 to 3000 |
| Feed Z (mm) | 50 to 2000 |
| Initial Diameter (mm) | 91.061 to 125.57 |
| Initial Temperature (°C) | 21.6 to 31 |
| Theoretical Working Time (s) | 6 to 2,034 |
| Difference between Theoretical and Real Time (s) | -3 to 565 |
| Weathering (mm) | -8 to 22 |
| Temperature Variation (°C) | -5 to 6.7 |

4. Experiments and Results

This multidisciplinary research analyzed the data set to obtain the variables that are most important to optimize the time error in the manufacturing process.

In the first step, PCA and some unsupervised learning algorithms as MLHL and CMLHL were applied to identify the internal data sets structure and compare the results obtained by every algorithm.

In the case of PCA, the model is looking for those directions where variance is the biggest. An internal structure appears in the data set due to feed X, feed Y and feed Z increase when the X-axis values grow up and revolutions per minute

(RPM) increase when the Y-axis values decrease. In addition, five clusters are identified (Fig.4).

Once the internal structure of the data set is analysed it has been concluded that the most relevant variables are revolutions per minute (RPM), feed X, feed Y and feed Z variables.

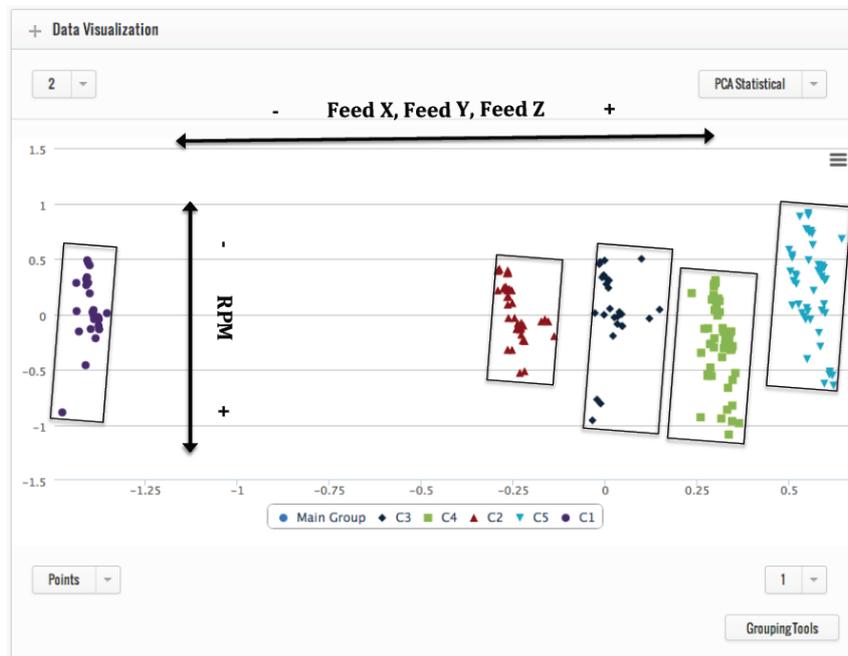


Figure 4. PCA projection

MLHL identifies five clusters as PCA projection but in this case the number of relevant variables is six instead of five as it can be seen in (Fig.5). Revolutions per minute (RPM), feed X, feed Y, feed Z, temperature, type of tool and difference between theoretical and real time are the most representative variables in this projection. It also has an internal structure in all clusters where revolutions per minute decrease when the Y-axis values increase and temperature values increase when the X-axis values increased.

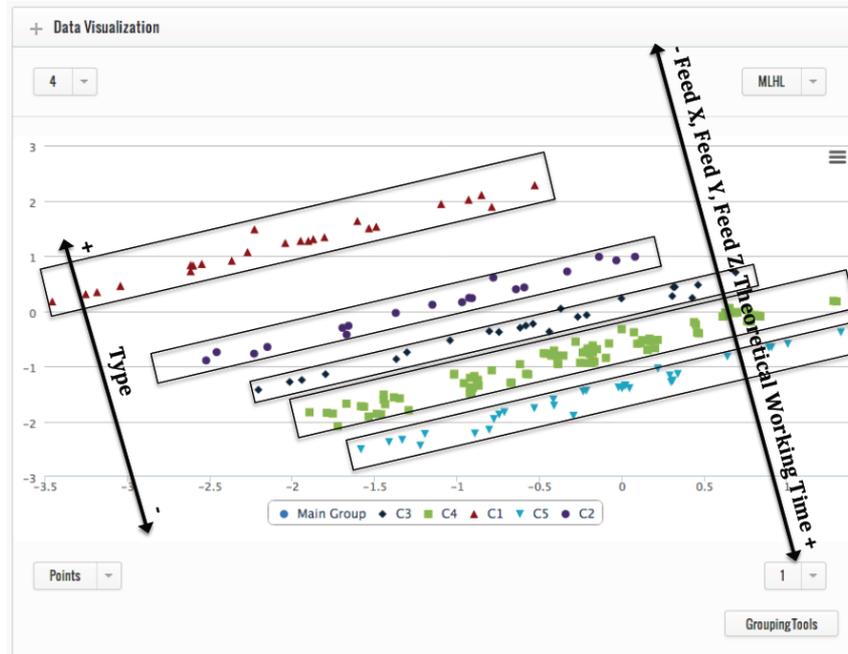


Figure 5. MLHL projections with $iters = 10,000$, $lrate = 0.05$, $m = 4$ and $p = 0.02$.

In the last case, CMLHL algorithm was applied to improve MLHL results and it got the best representation as it was showed in (Fig.6).

CMLHL projection is showed in (Fig.6). The results obtained were better than the others projections by PCA and MLHL. The clusters were well defined. Six clusters were identified and the most important variables were revolutions per minute (RPM), feed X, feed Y, feed Z, temperature, type of tool, difference between theoretical and real time and radius. It also has an internal structure in all clusters where revolutions per minute decrease when the Y-axis values increase and temperature values decrease when the X-axis values increased.

It has been concluded that the most relevant variables of the data set are eight: revolutions per minute (RPM), feed X, feed Y, feed Z, temperature, type of tool, theoretical working time and radius.

In the second step, these eight variables were selected to model the system. ANN and SVR are applied and compared.

The models are trained using an initial data set of 190 samples obtained by the dental scanner in the manufacturing of dental pieces. These samples are formed by 8 input variables (revolutions per minute (RPM), feed X, feed Y, feed Z, temperature, type of tool, theoretical working time and radius) and 1 output variable (difference between theoretical and real time).

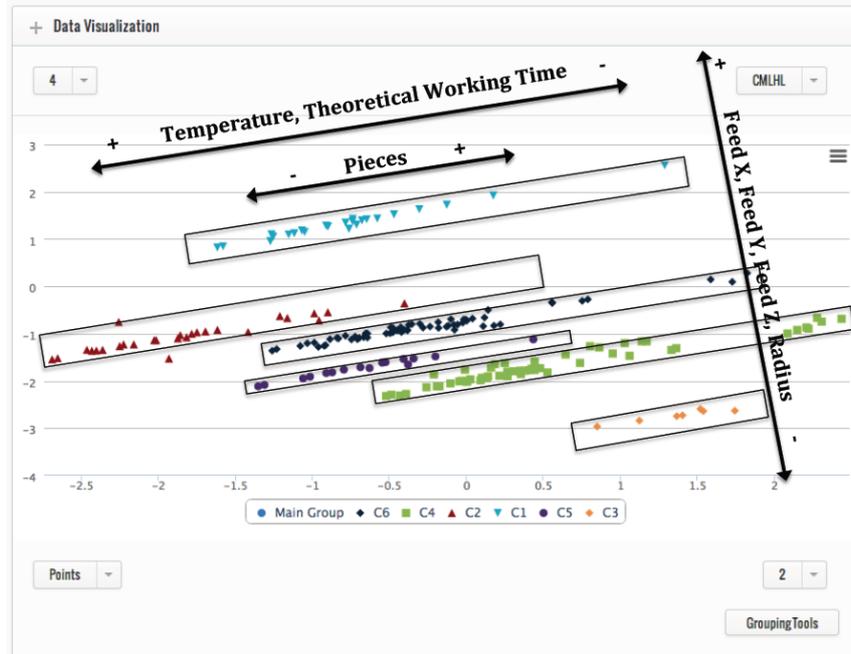


Figure 6. CMLHL projections with $iters = 100,000$, $lrate = 0.008$, $m = 4$, $p = 0.25$, $\tau = 0.1$.

The number of samples in the data set is small, for this reason a 12-fold cross-validation schema is used. This technique reduces variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. The final model is obtained using the full data set. Crossvalidation is applied to ANN and SVR.

MLP was tested with several difference structures, getting the best result using 4 hidden layers. The structure was made up by 12 neurons in the first layer, 3 neurons in the second layer, 10 neurons in the third layer and 6 neurons in the fourth layer. The selected activation function in the hidden layers was the tangent sigmoidal. The results applying MLP using all variables and the 8 main variables are shown in Table 2. It can be appreciated how the results using only the most relevant variables are better (67.5723%) than the ones which use all the variables (40.9975%).

LS-SVR was tested through Matlab toolbox. In this toolbox, the tuning of the parameters is conducted in two steps. First, a state-of-the-art global optimization technique, Coupled Simulated Annealing (CSA) [28], determines suitable parameters according to specific criterion. These parameters were then given to a second optimization procedure simplex to perform a fine-tuning step and

Radial Basis Function (RBF) kernel was selected. The optimal adjustment parameters are σ^2 and γ .

Table 2. Results for ANN applying all variables and the 8 main variables

| Variables | Model | Fit Test(%) | Fit Train(%) | Test Variance | Train Variance |
|-----------|-------|----------------|-----------------|------------------|-------------------|
| All | ANN | 40.9975 | 55.9809 | 22.2028 | 27.8434 |
| 8 Main | ANN | 67.5723 | 71.2486 | 8.2520 | 21.7662 |

Table 3. Results for SVM applying all variables and the 8 main variables

| Variables | Model | Fit Test(%) | Fit Train(%) | Test Variance | Train Variance | σ^2 | γ |
|-----------|-------|----------------|-----------------|------------------|-------------------|------------|----------|
| All | SVM | 75.1545 | 89.9686 | 1.3554 | 8.8492 | 71.5273 | 193.8523 |
| 8 Main | SVM | 75.0069 | 86.0872 | 2.1611 | 13.2415 | 3.94753 | 51.9611 |

The results applying SVM are shown in Table 3. It can be appreciated how the results are similar applying feature selection (75.0069%) and selecting all variables (75.1545%). These results justify that feature selection allows to discard the variables that are not relevant and reduce the computational cost.

Feature selection is necessary to identify the most relevant variables and reduce the computational cost. The results are also better applying SVM because the Fit Test parameter (75.1545%) is greater than ANN (67.5723%) and the variance is also more reduced for SVM (1.3554). For this reason, it has been concluded that SVM model is better than ANN model.

5. Conclusions

This contribution has been presented to prove the advantages of the combination of statistical and connectionist unsupervised models with identification systems, providing a two steps model characterized by indicating the relevance of the data. It avoids wasting time attempting to model a system with data that do not represent the system dynamics.

Feature selection has reduced the number of variables to be considered in the modeling process, with the least loss of information about the system. This reduces the complexity of the model and computational cost.

An intelligent two steps model has been presented which has allowed analyzing the real data set formed by 190 samples where every sample has 15 variables.

In the first step, some visualization techniques such as PCA, MLHL and CMLHL were applied. It has allowed to find the internal structure of the data set. At the first moment, an internal structure was found by applying PCA. MLHL and CMLHL are also applied and the most important variables were identified.

In the second step, ANN and SVM were applied to model the system. It has been concluded that SVM is better than ANN because SVM Fit Test is greater in a 7.5822% and SVM Variance Test is less than ANN model in a 6.8966.

Future research lines will be based on the analysis of some other features selection models as wrappers, filters and hybrid models. Also the application of other optimization models will be studied.

Finally the application of this intelligent model will be tested under the frame of some other industrial use cases.

Acknowledgement. This research has been funded through MIDAS project [ref:TIN2010-21272-C02-01] of the Spanish Ministry of Science and Innovation . The authors would also like to thank ESTUDIO PREVIO (Madrid-Spain) and TARAMI (Madrid-Spain) for their collaboration in this research.

Bibliography

- [1] T. T. Nguyen, S. Yang, and J. Branke, “Evolutionary dynamic optimization: A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, 2012.
- [2] E. Corchado, J. Sedano, L. Curiel, and J. R. Villar, “Optimizing the operating conditions in a high precision industrial process using soft computing techniques,” *Expert Systems*, vol. 29, no. 3, pp. 276–299, 2012.
- [3] E. Corchado, A. Abraham, and V. Snásel, “New trends on soft computing models in industrial and environmental applications,” *Neurocomputing*, vol. 109, pp. 1–2, 2013.
- [4] L. Ljung, ed., *System identification (2nd ed.): theory for the user*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- [5] Y. Irshad, M. Mossberg, and T. Söderström, “System identification in a networked environment using second order statistical properties,” *Automatica*, vol. 49, no. 2, pp. 652–659, 2013.
- [6] A. Smyk and M. Tudruj, “Genetic algorithms hierarchical execution control under a global application state monitoring infrastructure,” in *PDP*, pp. 16–23, 2013.
- [7] G. Palli, G. Borghesan, and C. Melchiorri, “Modeling, identification, and control of tendon-based actuation systems,” pp. 277–290, 2012.
- [8] P. Afshar, M. Brown, J. M. Maciejowski, and H. Wang, “Data-based robust multiobjective optimization of interconnected processes: Energy efficiency case study in papermaking,” pp. 2324–2338, 2011.
- [9] A. Ancuta, O. Company, and F. Pierrot, “Modeling and optimization of quadriglide, a schönflies motion generator module for 5-axis milling machine-tools,” in *ICRA*, pp. 2174–2179, 2009.
- [10] E. Corchado, J. Sedano, L. Curiel, and J. R. Villar, “Optimizing the operating conditions in a high precision industrial process using soft computing techniques,” *Expert Systems*, vol. 29, no. 3, pp. 276–299, 2012.
- [11] A. B. Nassif, D. Ho, and L. F. Capretz, “Towards an early software estimation using log-linear regression and a multilayer perceptron model,” *Journal of Systems and Software*, vol. 86, no. 1, pp. 144–160, 2013.
- [12] Y. Marinakis, M. Marinaki, G. Dounias, J. Jantzen, and B. Bjerregaard, “Intelligent and nature inspired optimization methods in medicine: the pap smear cell classification problem,” *Expert Systems*, vol. 26, no. 5, pp. 433–457, 2009.
- [13] T. K. Pratihar and D. K. Pratihar, “Design of cluster-wise optimal fuzzy logic controllers to model input-output relationships of some manufacturing processes,” *IJDMMM*, vol. 1, no. 2, pp. 178–205, 2009.
- [14] X. C. Guo, C. Wu, M. Marchese, and Y. Liang, “Ls-svr-based solving volterra integral equations,” *Applied Mathematics and Computation*, vol. 218, no. 23, pp. 11404–11409, 2012.

- [15] M. Deypir, M. H. Sadreddini, and S. Hashemi, "Towards a variable size sliding window model for frequent itemset mining over data streams," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 161–172, 2012.
- [16] E. Corchado and B. Baruque, "WeVoS-visom: An ensemble summarization algorithm for enhanced data visualization," *Neurocomputing*, vol. 75, no. 1, pp. 171–184, 2012.
- [17] R. Karakis, M. Tez, Y. A. Kiliç, B. Kuru, and I. Güler, "A genetic algorithm model based on artificial neural network for prediction of the axillary lymph node status in breast cancer," *Eng. Appl. of AI*, vol. 26, no. 3, pp. 945–950, 2013.
- [18] B. S. Villanueva and M. Sánchez-Marrè, "Case-based reasoning applied to textile industry processes," in *ICCBR*, pp. 428–442, 2012.
- [19] D. Shen, H. Shen, and J. S. Marron, "Consistency of sparse pca in high dimension, low sample size contexts," *J. Multivariate Analysis*, vol. 115, pp. 317–333, 2013.
- [20] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.
- [21] P. Krömer, E. Corchado, V. Snásel, J. Platos, and L. García-Hernandez, "Neural pca and maximum likelihood hebbian learning on the gpu," in *ICANN (2)*, pp. 132–139, 2012.
- [22] V. Vera, E. Corchado, R. Redondo, J. Sedano, and A. E. Garcia, "Applying soft computing techniques to optimise a dental milling process," *Neurocomputing*, vol. 109, pp. 94–104, 2013.
- [23] M. Ghiassi and S. Nangoy, "A dynamic artificial neural network model for forecasting nonlinear processes," *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 287–297, 2009.
- [24] A. Rosales-Pérez, H. J. Escalante, J. A. Gonzalez, C. A. R. García, and C. A. C. Coello, "Bias and variance multi-objective optimization for support vector machines model selection," in *IbPRIA*, pp. 108–116, 2013.
- [25] A. Berro, S. L. Marie-Sainte, and A. Ruiz-Gazen, "Genetic algorithms and particle swarm optimization for exploratory projection pursuit," *Ann. Math. Artif. Intell.*, vol. 60, no. 1-2, pp. 153–178, 2010.
- [26] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *Journal of Systems and Software*, vol. 86, no. 1, pp. 144–160, 2013.
- [27] A. Rosales-Pérez, H. J. Escalante, J. A. Gonzalez, C. A. R. García, and C. A. C. Coello, "Bias and variance multi-objective optimization for support vector machines model selection," in *IbPRIA*, pp. 108–116, 2013.
- [28] S. X. de Souza, J. A. K. Suykens, J. Vandewalle, and D. Bollé, "Coupled simulated annealing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 2, pp. 320–335, 2010.

Análisis Cibernético y Visual de Twitter

Zhangxian Zheng¹, José Luis Alonso Berrocal¹ y Luis Carlos García de Figuerola Paniagua¹

¹Departamento de Informática y Automática - Facultad de Traducción y Documentación, Universidad de Salamanca. Calle de Francisco Vitoria 6-16. 37008 Salamanca, España
{r.carlos, berrocal, figue}@usal.es

Resumen Este trabajo aborda el tema de análisis cibernético y visual de *Twitter*. En él se describe como resolver la necesidad de recoger el perfil, los seguidores y seguidos de un usuario de *Twitter* vía su propia API y desarrollar una aplicación de crawler en *Python* utilizando la librería *Python-Twitter*, con el objetivo de realizar un análisis y visualización de la red que construyen los usuarios de Twitter mediante una herramienta desarrollada.

Keywords: Twitter, Cibermetría, Crawler, Analítica Visual

1. Introducción

En la vida social actual, todo el mundo tiene sus propias relaciones personales; este tipo de relaciones existen entre las personas que constituyen la red social[1] de la vida real. En los últimos años, con la evolución de Internet, han comenzado a aparecer sitios web promocionando las redes de círculos de amigos en línea. El término se empleaba para describir las relaciones en las comunidades virtuales y se hizo popular en 2003 con la llegada de sitios tal como *MySpace*, y continúan avanzando en Internet a pasos agigantados, especialmente dentro de lo que se ha denominado Web 2.0.

Twitter [2] es uno de los sitios más relevantes hoy en día; en la actualidad, tiene más de 200 millones de usuarios activos y se publican más de 400 millones de mensajes de texto cada día [3]. Las informaciones que contiene en su sitio web son muy enriquecidas, significativas e importantes, por lo tanto nos interesa recoger datos desde este sitio web y hacer análisis y visualización para descubrir nuevos conocimientos.

2. Estado del Arte

2.1. Métricas de Análisis de Redes Sociales

Las redes sociales se pueden representar mediante grafos, y se utilizan técnicas de la teoría de grafos para estudiar la estructura de éstas. Sin embargo, existen diferencias en la forma en que se aplican los distintos conceptos de la

teoría de grafos sobre el Análisis de Redes Sociales (ARS). Las propiedades que presentan las redes sociales se pueden representar en base a su forma, distribución y similitud entre los conjuntos de nodos y relaciones que existen en la red.

- **Cohesión:** es una fuerza de atracción entre los individuos, que puede ser medida por el número de intercambios entre dos actores, el promedio de las distancias geodésicas, el tamaño de la red, etc.
- **Densidad:** la densidad de la red nos muestra la alta y baja conectividad de ésta, expresada en porcentaje como el cociente entre el número de relaciones activas entre las posibles. Se mide la proporción de enlaces que existen entre las relaciones posibles de una red en concreto.
- **Diámetro de la red:** el diámetro es la distancia de grafo más larga entre dos nodos cualesquiera en la red, el mayor valor de los geodésicos de la red. Representa el tamaño del grafo y permite saber lo grande que es.
- **Centralidad de Grado:** se define al grado de un nodo como el número de enlaces que posee, es decir, el número de relaciones que tiene el nodo con los otros nodos. Para los grafos dirigidos el grado del nodo suele ser dividido en dos formas: el grado de entrada y el grado de salida.
- **Centralidad de Intermediación:** este tipo de centralidad basada en la frecuencia en la que un nodo aparece en el geodésico entre dos nodos, es la medida de las veces en que se presenta entre un nodo con trayectoria mínima.
- **Centralidad de Cercanía:** la centralidad de cercanía es la distancia media desde un nodo inicial a todos los demás nodos de la red, mide los pasos requeridos para acceder a cada vértice desde un vértice dado.[4]

2.2. *Twitter* y *Twitter API*

El funcionamiento de *Twitter* es muy simple: el usuario envía mensajes de texto plano de corta longitud, con un máximo de 140 caracteres, llamados “tweets”, que se muestran en la página principal del usuario y que pueden ser leídos por cualquiera que tenga acceso a su página. Los usuarios pueden suscribirse a los “tweets” de otros usuarios, a esto se le llama “seguir” y a los usuarios abonados se les llama “seguidores”. A los usuarios a los que se sigue se les llama “seguidos”. [5]

Uno de los mayores éxitos de *Twitter* ha sido proporcionar una API abierta y funcional desde sus inicios, lo que permite que otras personas extiendan las características que *Twitter* ofrece y desarrollen aplicaciones para enviar y compartir nuestros tweets sin estar obligados a utilizar el sitio oficial. Resulta interesante como han surgido las posibilidades de permitir apreciar y analizar los datos del servicio en diferentes formas. Para recoger los perfiles de usuario y obtener los seguidos y los seguidores para crear un grafo de relaciones, son útiles los siguientes métodos del API[6]:

- **GET users/show:** Devuelve una información amplia de un solo usuario determinado, especificado por el ID o el nombre de la pantalla (*screen names*) según el parámetro id requerido.
- **GET followers/ids:** Devuelve un conjunto de IDs de cada usuario que está siguiendo al usuario específico, es decir, todos IDs de los seguidores/*followers* de un usuario específico.
- **GET friends/ids:** Este método es similar al anterior, en lugar de devolver los id de los seguidores/*followers*, devuelve los de los seguidos/*friends*.
- **GET users/lookup:** Este método es casi igual que GET users/show, la única diferencia es que este método devuelve información de hasta 100 usuarios por una petición.

2.3. Web Crawler

Un *Web Crawler* es un sistema capaz de recorrer, de forma planificada, un conjunto de recursos. Uno de los usos más frecuentes de un *Web crawler* es un programa que inspecciona las páginas del *World Wide Web* de forma metódica y automatizada, creando una copia de todas las páginas web visitadas para su procesamiento posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido.[7]

Hay varias formas de hacer un recorrido de *Web Crawler* en la web, aunque básicamente existen tres[8]: recorrido en anchura, recorrido en profundidad y el mejor posible. Las dos primeras son bien conocidas y dependen de que dicha lista se organice como una pila o como una cola. En el primer caso, el entorno inmediato y más cercano a una página es explorado primero, resulta útil cuando se buscan exploraciones exhaustivas de un dominio, aun cuando éstas se limiten a una determinada profundidad solamente.

2.4. Herramientas de Análisis y Visualización

La visualización de las redes sociales sirve como método para descubrir propiedades en las redes, aunque tiene menos peso teórico en el análisis, pero tiene la ventaja de alimentar rápidamente la intuición del investigador [9]. En la actualidad existen diversas herramientas desarrolladas para el análisis y visualización de redes, tales como *Pajek*, *NodeXL* o *UCINET*. *Gephi* [10] es una de las más relevantes.

Gephi es una herramienta de código abierto multiplataforma bajo la licencia GNU GPL 3 y gratuita. Es una de las herramientas más completas para manejar todo tipo de datos y gráficos y con un desempeño bastante considerable, incluso al manejar y procesar datos sumamente grandes, ofreciendo los cálculos de ARS tales como Grado, Diámetro, Centralidad y Modularidad. También está disponible una variedad de disposiciones para la visualización de grafos.

3. Desarrollo de la Aplicación

El primer paso para el desarrollo del trabajo es la realización de un *crawler* de información para *Twitter*. Este *crawler* debe ser capaz de obtener y almacenar información sobre *Twitter* de manera estructurada, continua y además, debe ser tolerante a fallos. El *crawler* de *Twitter* está íntegramente escrito en *Python* [11] y se apoya en la librería *Python Twitter Tools* [12] para realizar la autenticación y las llamadas a *Twitter API* [13].

3.1. Análisis de los Datos a Extraer

Los principales datos que extraeremos desde el *Twitter* de la Usal (Universidad de Salamanca) son los perfiles de todos seguidores de él y un grafo de la red social. Desde la nueva versión 1.1 de *Twitter API*, los resultados devueltos por el servidor de *Twitter* se encuentran únicamente en formato JSON. La estructura de datos es sencilla, cada clave tiene su nombre entre comillas y con su correspondiente valor después del caracter “:”, y los diferentes pares “clave”:valor se separan por comas, como la estructura de datos de diccionario de *Python*, facilitando el tratamiento previo y almacenamiento de los datos recogidos.

Los perfiles de usuarios de *Twitter* contienen las siguientes informaciones que nos pueden interesar extraer para hacer algún análisis estadístico: localización, huso horario, idiomas, la cantidad de tweets, seguidores, seguidos y fecha de creación de cuenta.

3.2. Arquitectura, Diseño y Funciones

Hay tres componentes importantes dentro del *crawler*: Autenticación, *Crawler* y Tratamiento previo. La Autenticación es el componente imprescindible del *crawler*, debido a que desde el retiro del antiguo *Twitter API v1*, cualquier petición enviada al servidor de *Twitter* requiere la autenticación del usuario o la aplicación. El *Crawler* es el componente principal, se encarga de las funciones principales, como mandar peticiones, recuperarse del límite de llamadas al API o error de red, extraer informaciones de interés desde los perfiles, etc. El Tratamiento previo es el componente encargado del pre-proceso y almacenamiento de datos en ficheros para su uso posterior, como importarlo en *Gephi* para hacer análisis y visualización.

El *Twitter Crawler* se compone de varias funciones y una clase de interfaz gráfica. A continuación se explica en detalle cada una de las funciones:

- **Autenticación:** para obtener el acceso completo a toda la *Twitter API* es necesario obtener la autenticación vía *OAuth*. Para ello, necesitamos crear una aplicación en *Twitter Apps* (<https://dev.twitter.com/apps>), y obtenemos el nombre de la aplicación (*App Name*), *Consumer Key* y *Consumer Secret* para el proceso de autenticación.

- **Mandar peticiones robustas [14]:** cuando mandamos peticiones al servidor de Twitter, los resultados que nos devuelve no siempre son los que esperamos, muchas veces el servidor de Twitter nos devuelve código error de HTTP para expresar si ha ocurrido algo. El *Twitter crawler* debe saber como responder a los códigos de error devuelto, por eso utilizamos la función *make_twitter_request* para manejar los errores.
- **Crawl por seguidores o seguidos:** para crear un grafo con los seguidores o seguidos de uno o varios usuarios de *Twitter*, necesitamos hacer al *crawler* caminar desde los primeros usuarios, recoger y seguir por sus seguidores o seguidos, hasta un nivel de profundidad deseado. Es necesario establecer alguna regla al *crawler*. En el *Twitter crawler v1* implementamos el nivel de profundidad de *crawl* y el número mínimo y máximo de seguidores de un usuario para una mejor recogida de datos.
- **Guardar resultados:** es necesario de guardar los resultados obtenemos mediante el *crawl* para usos posteriores. En lugar de utilizar una base de datos, *Python* proporciona el módulo *pickle* para almacenar sus objetos en ficheros, con la ventaja de funcionar bastante rápido y de recuperar la información sin problemas.
- **Exportación en CSV;** para importar datos a una herramienta de análisis o visualización, hay que proporcionar el dato en el formato adecuado. Por ejemplo en el siguiente apartado planteamos utilizar la herramienta *Gephi*, y necesitamos el dato en formato CSV: se escriben dos ids de usuarios separados por una coma, que significa que el usuario anterior está siguiendo al posterior.
- **Recoger perfiles:** aparte de los seguidores y seguidos, también nos interesa recoger el perfil de los usuarios, incluyendo su descripción, idiomas, localización, fecha de creación en *Twitter*, etc. Podemos utilizar el método *GET users/lookup* para lograr hasta 100 perfiles en una petición.
- **Obtener nombres:** El resultado de *crawl* proporciona los seguidores y seguidos mediante los ids de los usuarios. Debido a que necesitamos crear el grafo con las etiquetas en nombres, hacemos la transformación a través del método para recoger perfiles.

4. Análisis Cíbermétrico y Visualización

4.1. Descripción de Datos

- **Perfiles de todos seguidores de @usal:** El twitter de la Universidad de Salamanca tiene 14943 seguidores. Hemos recogido los perfiles de cada uno de ellos, estos datos están almacenado en formato Python pickle.

- **Red social en Twitter a partir de @usal:** Es una red de 3 niveles de profundidad desde el usuario la Universidad de Salamanca. Cada usuario que forma parte de esta red debe tener al menos 5 seguidores o seguidos, y pueden tener hasta 20 seguidores o seguidos en esta red, en resumen, esta red esta constituida por 37684 usuario y 47404 relaciones entre ellos.
- **Perfiles de todos usuarios de la red @usal:** En este dato se incluyen los perfiles de todos usuarios que aparecen en el dato de Red social en *Twitter* a partir de @usal.

4.2. Análisis Estadístico

El twitter de la Universidad de Salamanca tiene aproximadamente 15000 seguidores, que provinientes de diferentes zonas del mundo y tienen diversas características. También ocurre en el grupo de usuarios que aparecen en la red de @usal. Nos interesa a hacer un análisis estadístico sobre ellos para ver como están relacionados los dos grupos de usuarios, qué diferencia y similitud tienen. A continuación detallamos el resultado que obtenemos atendiendo a diferentes aspectos

Idiomas. En el perfil de usuario de *Twitter*, cada usuario tiene su idioma codificado con dos letras de abreviación, por ejemplo “es” significa español, “en” es inglés, etc. Como la Universidad de Salamanca es una universidad española, el resultado es como esperábamos: la mayoría hablan en español, y el inglés ocupa el segundo lugar.

| Idiomas | Seguidores @usal | Red @usal |
|---------------------|------------------|-----------|
| Español | 90,24 % | 79,00 % |
| Inglés | 6,37 % | 16,10 % |
| Dialectos en España | 0,38 % | 0,97 % |
| Idiomas europeos | 2,79 % | 3,03 % |
| Idiomas orientales | 0,20 % | 0,34 % |

Localización: Para este dato de perfiles es un poco difícil de hacer un análisis estadístico, porque el contenido de este campo no está normalizado, es libre de editar por los usuarios. Lo que podemos hacer es sacar los términos que aparecen con más frecuencia en las localizaciones de perfiles.

| Términos | Frecuencia en Seguidores @usal | Términos | Frecuencia en Seguidores @usal |
|------------|--------------------------------|-----------|--------------------------------|
| España | 1921 | Sevilla | 75 |
| Salamanca | 3104 | Valencia | 83 |
| Valladolid | 206 | Cáceres | 100 |
| Ávila | 150 | Argentina | 116 |
| León | 164 | México | 108 |
| Madrid | 582 | Venezuela | 103 |
| Barcelona | 159 | Colombia | 102 |

Huso horario. A diferencia de la localización, el huso horario en cada perfil de usuario está normalizado. Podemos obtener un resultado sobre esto para generalizar el estado geográfico del grupo de usuarios. Sin embargo, como no todos los usuarios rellenan esta parte del perfil, solo obtenemos las dos terceras partes de ellos.

| Huso horario | Seguidores @usal | Red @usal | Huso horario | Seguidores @usal | Red @usal |
|-------------------|------------------|-----------|--------------|------------------|-----------|
| Amsterdam | 8,26 % | 5,81 % | Hawaii | 1,55 % | 1,46 % |
| Athens | 8,71 % | 7,13 % | London | 1,22 % | 2,01 % |
| Caracas | 0,86 % | 1,21 % | Madrid | 24,99 % | 18,59 % |
| Central Time (US) | 1,30 % | 3,38 % | Mexico City | 0,82 % | 1,83 % |
| Eastern Time (US) | 0,95 % | 3,96 % | Otros | 8,50 % | 13,90 % |
| Greenland | 9,26 % | 4,59 % | Unknown | 33,58 % | 36,13 % |

Número de Tweets: En el perfil de usuario también viene la información sobre cuántos tweets ha publicado el usuario, en el que podemos observar la actividad de éste.

| Número | Seguidores @usal | Red @usal |
|----------------|------------------|-----------|
| 0 | 3,14 % | 2,42 % |
| 1-10 | 7,66 % | 6,72 % |
| 11-100 | 18,29 % | 17,26 % |
| 101-1.000 | 36,22 % | 32,40 % |
| 1.001-10.000 | 31,62 % | 32,71 % |
| 10.001-100.000 | 3,04 % | 8,29 % |
| 100.000+ | 0,02 % | 0,21 % |

Número de seguidores y seguidos: Los números de seguidores o seguidos que tiene un usuario también pueden reflejar su actividad de uso en *Twitter*.

| Número de seguidores | Seguidores @usal | Red @usal | Número de seguidores | Seguidores @usal | Red @usal |
|----------------------|------------------|-----------|----------------------|------------------|-----------|
| 0 | 0,63 % | 0,54 % | 10.001-100.000 | 0,62 % | 7,49 % |
| 1-10 | 6,72 % | 5,70 % | 100.001-1.000.000 | 0,07 % | 2,82 % |
| 11-100 | 37,43 % | 25,75 % | 1.000.000+ | 0,01 % | 0,66 % |
| 101-1.000 | 50,08 % | 42,13 % | 10.000.000+ | - | 0,06 % |
| 1.001-10.000 | 4,44 % | 14,85 % | - | - | - |

| Número de seguidos | Seguidores @usal | Red @usal | Número de seguidos | Seguidores @usal | Red @usal |
|--------------------|------------------|-----------|--------------------|------------------|-----------|
| 0 | 0 | 0,41 % | 1.001-10.000 | 12,07 % | 17,68 % |
| 1-10 | 1,98 % | 2,15 % | 10.001-100.000 | 0,18 % | 2,38 % |
| 11-100 | 20,20 % | 18,18 % | 100.000+ | 0,01 % | 0,33 % |
| 101-1.000 | 65,56 % | 58,86 % | 1.000.000+ | - | 1 |

Fecha de creación: Por último, el resultado sobre la fecha de creación de los usuarios en Twitter muestra que la mayoría seguidores de @usal se han registrado entre el año 2010 y 2012, y los de la red @usal se registran más uniformemente.

| Año | Seguidores @usal | Red @usal | Año | Seguidores @usal | Red @usal |
|------|------------------|-----------|------|------------------|-----------|
| 2006 | - | 0,04 % | 2010 | 17,23 % | 15,94 % |
| 2007 | 0,23 % | 0,80 % | 2011 | 38,49 % | 25,11 % |
| 2008 | 0,87 % | 2,00 % | 2012 | 26,92 % | 24,26 % |
| 2009 | 7,26 % | 10,38 % | 2013 | 9,00 % | 21,47 % |

4.3. Análisis de Redes Sociales

Para encontrar algunos nodos o patrones más significativos en la red @usal, necesitamos calcular y conseguir las métricas del Análisis de Redes Sociales de esta red. En la herramienta *Gephi* nos facilitan diversos cálculos de este aspecto. A continuación presentamos los resultados obtenidos.

Grado. El grado medio de la red @usal es 1,258. En la siguiente tabla se presentan los 12 primeros nodos que tienen el grado más alto en esta red y sus descripciones. Los usuarios de esta tabla son muy diversos, @usal ocupa el duodécimo puesto en la tabla. Además, se presentan los usuarios que tienen el grado de entrada y salida más alto en esta red.

Tabla 1. Usuarios con el grado más alto

| | Nombre | Descripción | Grado |
|-----------|---------------------|--|-----------|
| 1. | leonactualidad | Diario digital de la localidad de Leon. | 67 |
| 2. | SalamancArlequinada | Por un baloncesto de calidad en Salamanca | 66 |
| 3. | Educación INTEF | Instituto Nacional de Tecnologías Educativas | 62 |
| 4. | Farmacia EL PUENTE | Un lema 365 días de salud Evolucionando | 61 |
| 5. | Zamora Perfecta | Zamora. Desconocida. Perfecta. | 60 |
| 6. | | | .. |
| .. | | | .. |
| 12 | Usal | La Universidad de Salamanca es.. | 54 |

Tabla 2. Usuarios con grado de entrada más alto

| | Nombre | Descripción | Grado |
|-----------|---------------------|--|-----------|
| 1. | National Geographic | Desde 1888, hemos estado viajando por Tierra | 36 |
| 2. | Usal | La Universidad de Salamanca es una.. | 34 |
| 3. | Policía Nacional | Cuerpo Nacional de Policía de España | 33 |
| 4. | ElinformerdelaUSAL | Envíale tu mensaje al Informer y Él se encargará | 32 |
| 5. | EL PAIS | Las noticias globales más relevantes y la última | 30 |

Tabla 3. Usuarios con grado de salida más alto

| | Nombre | Descripción | Grado |
|----|---------------------|--|-------|
| 1. | leonactualidad | Diario digital de la localidad de Leon. | 44 |
| 2. | Farmacia EL PUENTE | Un lema 365 dias de salud Evolucionando | 41 |
| 3. | SalamancArlequinada | Por un baloncesto de calidad en Salamanca | 38 |
| 4. | BIOTECHNEWS | Si eres un científico español tus tweets merecen | 35 |
| 5. | RECIMOVIL | Compra-Venta telefonos de segunda mano: | 35 |

Diámetro y Centralidad: El diámetro de esta red es 18. La distancia media de grafo es 7,84. El usuario que tiene la intermediación más alta en esta red es *@usal*, que posee un valor de intermediación siete veces más alto que el segundo. Se presentan los cinco primeros usuarios con Intermediación más alta en esta red. Se puede observar un punto interesante: como hemos visto, hasta el momento, todos los nodos relevantes en esta red son de organizaciones; sin embargo en esta lista aparecen tres usuarios como individuos, que son dos estudiantes y un delegado.

Tabla 4. Usuarios con Intermediación más alto

| | Nombre | Descripción | Intermedia. |
|----|---------------------|--|---------------------|
| 1. | Usal | La Universidad de Salamanca es | 144 millones |
| 2. | Zamora Perfecta | Zamora. Desconocida. Perfecta. | 24 millones |
| 3. | Silvia Ramos Castro | Tecnica Superior en Educación Infantil | 23 millones |
| 4. | Susana Junquera Rey | Estudiante de Educación Social y.. | 22 millones |
| 5. | Rafael Daniel | Delegado de El Economista en.. | 19 millones |

4.4. Visualización de Twitter

En *Gephi* se ofrecen diversas disposiciones para visualización de datos. En nuestro trabajo, elegimos la disposición *ForceAtlas 2* [15] y *OpenOrd* [16] para visualizar este gran tamaño red, que son algoritmos que tienen la capacidad de tratar gran cantidad de nodos de un modo bastante rápido (ver Fig. 1).

Podemos ajustar las visualizaciones para descubrir patrones u obtener una imagen más intuitiva. A continuación aplicamos los resultados de cálculo que obtenemos anteriormente a los diferentes atributos de los nodos (color y tamaño) para poder presentar más informaciones.

En la disposición de *ForceAtlas 2*, aplicamos el grado al tamaño de nodo y una gama de colores rojo-blanco-azul al color de nodos. El nodo que tiene el color más rojo significa que tiene menos grado, y al contrario, el que tiene color más azul, tiene el grado más alto. En la disposición de *OpenOrd*, asignamos al tamaño del nodo su valor de Intermediación y su color con una gama de color de blanco a rojo según su grado de entrada. El nodo con color más rojo tiene el grado de entrada más alto. Como el nodo *@usal* tiene un valor de Intermediación

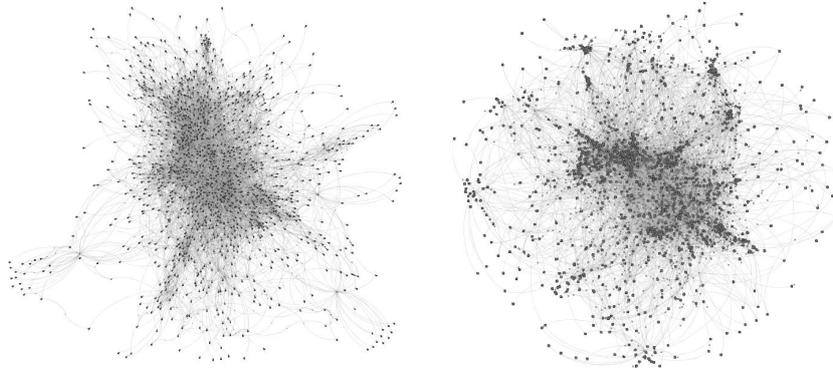


Figura 1. Visualización con ForceAtlas 2 y OpenOrd

mucho más alta que los demás, y su grado de entrada también es alto en la red, es fácil de encontrar este nodo en la figura de visualización (ver Fig. 2).

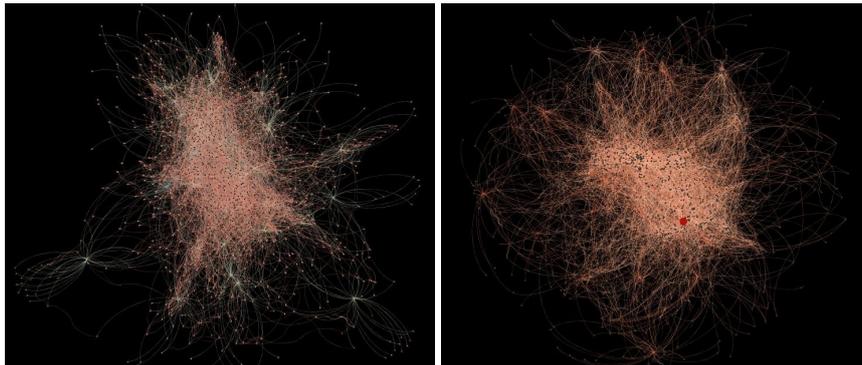


Figura 2. Visualización con grado y centralidad

Por otro lado, podemos utilizar los resultados de Modularidad, el algoritmo de detección de comunidades, tinturar los nodos y aristas con diferente color según su grupo de comunidad, para obtener una nueva representación de visualización (ver Fig. 3).

5. Conclusiones

La realización de este trabajo ha servido para aprender cómo manejar el *Twitter API* y emplearlo en un *crawler*, controlando el comportamiento del

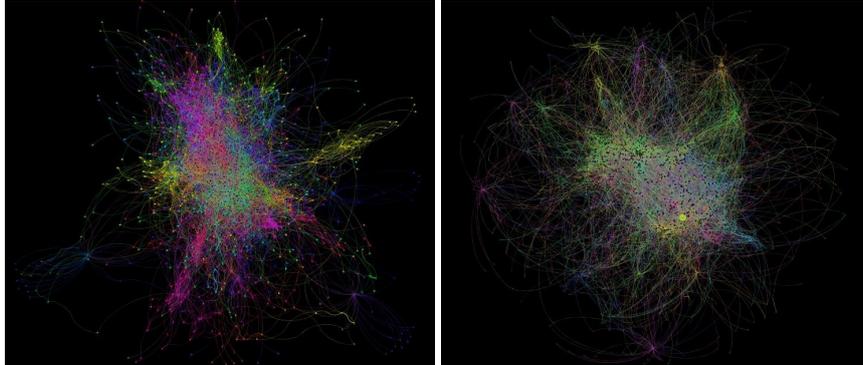


Figura 3. Visualización con Modularidad

crawl para conseguir una mejor recogida de datos. Hemos realizado un análisis y visualización de los datos obtenidos con las herramientas de análisis, hemos hecho una prueba con el Twitter de la Universidad de Salamanca y conseguimos un resultado muy interesante, que significa conseguimos el método para resolver el problema y la necesidad que planteamos al principio del trabajo. Desde este punto de vista, podemos decir hemos alcanzado los objetivos del trabajo.

Sin embargo, no hay que olvidar las deficiencias que existen en este trabajo, hay muchos aspectos que serán susceptibles de mejora, por ejemplo el límite impuesto por *Twitter API* es el mayor problema que nos acompaña durante toda la realización de trabajo, especialmente en el *Twitter API v1.1*, donde que se ha reducido considerablemente el límite. Deberíamos encontrar una manera para resolver esta problemas evitando gastar mucho más tiempo en la espera de recogida de datos que en el proceso más importante de análisis de los datos.

Para el trabajo futuro, podemos desarrollar un *crawler* de tipo distribuido para solucionar el problema de límite de API, podemos mejorar el presente trabajo en diferentes aspecto como una interfaz gráfica más estable, un algoritmo de *crawl* más eficiente o añadir más funciones en el crawler. Además, aparte de *Twitter* existen muchos sitios de redes sociales en Internet, pudiendo aplicar el método que aprendimos en este trabajo a otros sitios dependiendo sus APIs disponibles.

Bibliografía

- [1] Wikipedia, “Red social — wikipedia, la enciclopedia libre,” 2013. [Internet; descargado 1-julio-2013].
- [2] Wikipedia, “Twitter — wikipedia, la enciclopedia libre,” 2013. [Internet; descargado 1-julio-2013].
- [3] TwitterBlog, “Celebrating #twitter7 | twitter blog,” 2013. [Internet; descargado 1-julio-2013].
- [4] Wikipedia, “Centralidad — wikipedia, la enciclopedia libre,” 2013. [Internet; descargado 15-julio-2013].
- [5] J. L. Alonso Berrocal, “Twitter,” 2013. [Internet; descargado 1-julio-2013].
- [6] Dev.twitter.com, “Twitter developers,” 2013. [Internet; descargado 1-julio-2013].
- [7] J. L. Alonso Berrocal, C. García Figuerola Paniagua, and Á. F. Zazo Rodríguez, “Sacarino (sonda automática para la recuperación de información en la web): un robot para recorrer y procesar la web.” *Scire: representación y organización del conocimiento*, vol. 12, no. 1, pp. 211–224, 2006.
- [8] J. L. Alonso Berrocal, C. García Figuerola Paniagua, and Á. F. Zazo Rodríguez, “Diseño de spiders,” tech. rep., Departamento de Informática y Automática, Universidad de Salamanca (España), 2006.
- [9] M. M. Moreno, “Análisis de redes sociales: un tutorial,” 2008.
- [10] Gephi.org, “Gephi, an open source graph visualization and manipulation software,” 2012. [Internet; descargado 1-julio-2013].
- [11] Python.org, “Python programming language – official website,” 2013. [Internet; descargado 1-julio-2013].
- [12] Mike.verdone.ca, “Python twitter tools (command-line client and irc bot),” 2013. [Internet; descargado 1-julio-2013].
- [13] M. A. Russell, *Mining the social web: Analyzing data from Facebook, Twitter, LinkedIn, and other social media sites*. O’Reilly, 2011.
- [14] M. Russell *et al.*, *21 Recipes for Mining Twitter*. O’Reilly, 2011.
- [15] M. Jacomy, S. Heymann, T. Venturini, and M. Bastian, “Forceatlas2, a graph layout algorithm for handy network visualization,” *Paris http://www.medialab.sciences-po.fr/fr/publications-fr*, 2011.
- [16] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack, “Openord: an open-source toolbox for large graph layout,” in *IS&T/SPIE Electronic Imaging*, pp. 786806–786806, International Society for Optics and Photonics, 2011.

Autores



José Luis Alonso Berrocal es Doctor en Informática por la Universidad de Salamanca. Actualmente desarrolla su actividad académica como Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca (Área de Lenguajes y Sistemas Informáticos). En el apartado de investigación cuenta con dos tramos de investigación y pertenece al grupo de investigación REINA (REcuperación de INformación Avanzada). Es autor de un importante número de artículos en revistas y congresos internacionales y ha participado en diferentes proyectos de investigación de carácter regional, nacional. Sus líneas de investigación son: Recuperación de Información, Extracción Automática de Conocimiento, Cibermetría, Redes Sociales, Recuperación de Información en el Web. Imparte docencia en estudios de grado y en 3 másteres. Es miembro del Instituto de Estudios de la Ciencia y la Tecnología.

Juan José Álvarez Navarrete nació en Donostia, San Sebastián, donde terminó sus estudios preuniversitarios. Cursó Ingeniería Informática en la Universidad Pontificia de Salamanca; periodo durante el que además participó tanto en las becas de colaboración con departamentos ofertadas por el País Vasco, como en las becas de la Fundación Caja Duero. Tras terminar el Máster en Sistemas Inteligentes impartido por la Universidad de Salamanca, se encuentra trabajando como programador en Madrid.



Astrid Paola Bogram cursó sus estudios de Ingeniería en Computación en la Universidad Tecnológica de Honduras. Desde 2007 hasta 2011 formó parte de la Dirección Ejecutiva de Ingresos (DEI) localizada en Honduras, una Institución Gubernamental enfocada a recolectar los impuestos tributarios y aduaneros de todo el país, donde ocupó el puesto de Gestor de Calidad en el Departamento de Tecnologías de Información, trabajando con un gran abanico de técnicas y métricas desarrolladas para pruebas de caja blanca y caja negra. Cuenta con una gran habilidad para conseguir que el código a entregar sea de alta calidad y limpio de forma rápida y correcta, consiguiendo un código estructurado, trabajando con acceso a bases de datos, programación OO y adecuándose al flujo de trabajo y a las reglas del negocio. Aquí mismo presentó su Tesis, “El Control de Calidad en el Software”. Posteriormente obtuvo una

beca para cursar sus estudios de Maestría obteniendo el título, “Máster en Sistemas Inteligentes” en la Universidad de Salamanca, España. Actualmente reside en la ciudad de New Orleans, Estados Unidos de Norte América, continuando sus estudios como complemento al área de Ingeniería en el Máster “Engineering management” de la Universidad de New Orleans.

Emilio S. Corchado es profesor Titular acreditado como Catedrático de Universidad por la ANECA (Agencia Nacional de Evaluación de la Calidad y Acreditación) desde febrero de 2012. Ha sido elegido presidente del IEEE España durante el periodo 2014-2015. Es coautor de diversas patentes y propiedades intelectuales. Cuenta con una dilatada experiencia en la revisión y evaluación de proyectos de la Comisión Europea desde 2007. Es coautor de más de 60 publicaciones en revistas científicas con alto índice de impacto y también actúa como miembro del Consejo de Redacción de varias revistas de reconocido prestigio internacional. Ha sido invitado como editor en varias “Special Issues” de alto renombre. Es el organizador de diversos congresos científicos como (HAIS, IDEAL, SOCO, CISIS, NABIC, CASON, NWESP, WICT). Ha participado en varios proyectos Europeos (IT4Innovation, APP for AAL, ENGINE, SOFTCOMP, EMaCS, IntelliCIS, NISIS). Además ha sido Investigador Principal de diversos proyectos a nivel nacional y regional, también ha estado trabajando en estrecha colaboración con empresas como Grupo Antolín, Inmota, Infranor, etc. Sus líneas de investigación se centran en el estudio de las redes neuronales, con especial énfasis en la Búsqueda Exploratoria de Proyecciones, Maximum Likelihood Hebbian Learning, Mapas Auto-Organizados, Redes SOM, Sistemas de Clasificación Múltiple y Sistemas Híbridos. Además de la optimización de procesos industriales y energéticos.



Juan M. Corchado es catedrático de Ciencia de la Computación e Inteligencia Artificial. Es actualmente Decano de la Facultad de Ciencias de la Universidad de Salamanca y Director del Grupo en Biotecnología, Sistemas Inteligentes y Tecnología Educativa de esta misma universidad (<http://bisite.usal.es>). Doctor en Informática por la Universidad de Salamanca, España (1998) y doctor en Inteligencia Artificial por la University of the West of Scotland, Reino Unido (2000). Es Coordinador Académico del Instituto de Arte y Tecnología de la Animación. Anteriormente ha sido Sub-director de la Escuela Superior de Ingeniería Informática en la Universidad de Vigo (España, 1999-00) e investigador en la Universidad de Paisley (1995-98). Colabora, como investigador, con el Laboratorio Oceanográfico de Plymouth (Reino Unido) desde 1993. Ha dirigido más de 70 proyectos de Inteligencia Artificial con financiación tanto

internacional, como nacional o autonómica. Ha dirigido 16 tesis doctorales y es co-autor de más de 300 libros, capítulos de libros, artículos en revistas científicas, etc. la mayoría de ellos presentan aspectos tanto prácticos como teóricos relacionados con los Sistemas Híbridos de Inteligencia Artificial, la Biomedicina, la Inteligencia Ambiental, los Sistemas Inalámbricos y la Oceanografía. Es autor de más de una veintena de propiedades intelectuales y patentes.

Juan Cruz Benito es Graduado en Ingeniería Informática por la Universidad de Salamanca y máster en Sistemas Inteligentes por la misma universidad. Actualmente es uno de los miembros más jóvenes del GRupo de Investigación en InterAcción y eLearning (GRIAL) de la Universidad de Salamanca, donde desempeña labores técnicas relacionadas con los ecosistemas de aprendizaje. Posee gran experiencia en la definición, desarrollo y gestión de Mundos Virtuales de Aprendizaje adquirida a través del proyecto USALSIM (<http://sim.usal.es>) coordinado desde el Servicio de Inserción Profesional, Prácticas y Empleo (SIPPE) de la Universidad de Salamanca y el proyecto de innovación docente UsalPharma (<http://campus.usal.es/farmacia/node/87>) coordinado por el Departamento de Farmacia y Tecnología Farmacéutica de la Facultad de Farmacia de la misma universidad.



José Rafael García-Bermejo Giner posee un doctorado en Físicas (1989, Universidad de Salamanca, España) y la Certificación de Apple ACTC T3 (Snow 100, Snow 101, Snow 201). Actualmente desarrolla su actividad académica como Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca en el área de Lenguajes y Sistemas Informáticos. Ha realizado estancias como docente e investigador en diferentes universidades de Alemania y Finlandia. Es coordinador Erasmus de los estudiantes de Ingeniería Informática. Sus principales líneas de investigación incluyen Programación Estructurada, Programación Orientada a Objetos, Interfaces Hombre-Máquina, Interfaces de Usuario, Dispositivos Móviles y Administración de Sistemas. Es autor y traductor técnico de un gran número de libros.

Carlos García de Figuerola es profesor en el Grado en Documentación de la Universidad de Salamanca, así como en el Máster y Doctorado en Sistemas Inteligentes y en el de Estudios Sociales sobre Ciencia y Tecnología. Su docencia se centra en las técnicas informáticas base de las Ciencias de la Información y Documentación, tanto en asignaturas de planes de estudio de primer, segundo y tercer ciclo, posgrado (máster y doctorado), títulos propios, cursos extraordinarios, etc. Además es miembro del grupo de investigación en REcuperación de INformación Avanzada (REINA), grupo reconocido de la Universidad de Salamanca, cuyos temas de interés son la recuperación de información y la cibermetría: Implementación de Conocimiento Lingüístico en Sistemas de Recuperación, Procesamiento de Lenguaje Natural, Recuperación de Información Multilingüe, Clasificación Automática, Recuperación Robusta, Recuperación Interactiva, Recuperación de Información en el Web, Cibermetría, etc.



Alicia García Holgado es Ingeniera Informática (2011, Universidad de Salamanca, España) y máster en Sistemas Inteligentes (2013, Universidad de Salamanca, España). Actualmente es uno de los miembros más jóvenes del GRupo de Investigación en InterAcción y eLearning (GRIAL), donde está especializada en la definición e implantación de soluciones *software* basadas en ecosistemas tecnológicos. Coordina y desarrolla la plataforma *eLearning* y las herramientas *software* que dan soporte a la gestión del conocimiento generado en GRIAL. Es experta en el uso de redes y herramientas sociales con el fin de definir y gestionar las estrategias de diseminación de la información. Ha realizado varios desarrollos *Open Source* para extender la funcionalidad de algunos de los CMS y LMS más utilizados actualmente. Ha participado en varios proyectos europeos como soporte tecnológico (MIH, TRAILER, EHISTO). Ha sido colaborada externa en la empresa Aqua Development Network (ADN) del grupo Agbar definiendo el ecosistema de formación.

Francisco José García Peñalvo es Ingeniero Informático por la Universidad de Valladolid y Doctor en Informática por la Universidad de Salamanca. Actualmente desarrolla su actividad académica como Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca. Desde el punto de vista de investigación dirige el GRupo de investigación en InterAcción y eLearning (GRIAL) de esta Universidad, contando entre sus principales líneas de investigación la Interacción Persona-Ordenador, el



eLearning, la Ingeniería Web, la Arquitectura del *Software*, la Informática Educativa, y la Teoría de la Comunicación. Es autor de más de un centenar de artículos en revistas y congresos internacionales y ha participado en más de cincuenta proyectos de investigación de carácter regional, nacional, europeo e internacional en los últimos diez años. Por último, en relación con la actividad de gestión, fue Vicedecano de Innovación y Nuevas Tecnologías de la Facultad de Ciencias de enero de 2004 a marzo de 2007 y Vicerrector de Innovación Tecnológica de la Universidad de Salamanca de marzo de 2007 a diciembre de 2009. Fue promotor y presidente del Comité de Dirección del Observatorio de la actividad, la innovación y las tendencias en la Formación en Red-SCOPEO desde su creación hasta diciembre de 2009.



Sergio García Sánchez se graduó en Licenciatura en Física e Ingeniería Técnica en Informática de Sistemas por la Universidad de Salamanca en 2011. En noviembre del 2011 comienza a ser investigador FPI-MICINN del proyecto “Diodos y Transistores avanzados para generación, detección y procesado de señales milimétricas y submilimétricas” en el Grupo de Investigación en Dispositivos Semiconductores de la Universidad de Salamanca (<http://www.usal.es/gelec>), donde está realizando su tesis doctoral, centrándose en el estudio de las oscilaciones Gunn en dispositivos semiconductores III-V. Además, es colaborador del proyecto europeo RROOTHz. En febrero de 2013 presentó el Trabajo de Grado de Salamanca titulado “Análisis Monte Carlo de la frecuencia de las oscilaciones Gunn y la eficiencia de conversión en diodos verticales y planares de GaAs, InP y GaN”. En junio de 2013 finalizó el Máster en Sistemas Inteligentes.

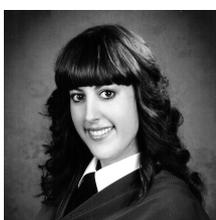
Angélica González Arrieta es doctora en Informática por la Universidad de Salamanca. Cuenta con una amplia experiencia investigadora en el campo de la Computación Neuroborrosa y las Redes Neuronales. Es Profesora Titular del Departamento de Informática y Automática de dicha Universidad. Actualmente compatibiliza su labor docente e investigadora con la dirección de diversas actividades formativas sobre seguridad informática, colaborando activamente con la Academia de la Policía Nacional de Ávila.





Daniel Hernández Alfageme nació en Zamora, España en 1989. Estudió Ingeniería Técnica en Informática de Gestión e Ingeniería Informática en la Universidad de Salamanca. Posteriormente realizó el Máster en Sistemas Inteligentes de la misma universidad. Actualmente trabaja como investigador y desarrollador en el Grupo de Investigación BISITE (Biomedicina, Sistemas Inteligentes y Tecnología Educativa, <http://bisite.usal.es>) en el que ha centrado su trabajo en las áreas de *Cloud Computing*, Agentes Inteligentes y Sistemas Multiagente.

Ignacio Iñiguez-de-la-Torre nació en Valladolid, España, en 1981. Se graduó en Física en la Universidad de Salamanca en 2004, donde recibió el título de Doctor en Física en 2008. Realizó una estancia de tres meses en el Institut d'Electronique, de Microélectronique et de Nanotechnologies (IEMN), en Francia en 2007 y otra de también de tres meses en la School of Electrical and Electronic Engineering en la Universidad de Manchester, Reino Unido en 2008. Trabajó durante un año, 2009-2010, en el Department of Electrical and Computer Engineering de la Universidad de Massachusetts en Lowell como Postdoc. Después, se unió al grupo de Electrónica del Departamento de Física Aplicada de la Universidad de Salamanca, primero durante 2010-2011 con un contrato Postdoc del proyecto europeo ROOTHZ y actualmente como Ayudante Doctor. Su investigación se concentra en el desarrollo de nuevos dispositivos y conceptos en la banda de los THz para el procesamiento de datos, detección y emisión de señales usando semiconductores III-V de gap ancho y estrecho.



María Navarro Cáceres nació en Salamanca el 8 de Marzo de 1990. Estudió Grado de Ingeniería Informática y un máster en Inteligencia Artificial por la Universidad de Salamanca, y de forma paralela, obtuvo el Grado Profesional en el Conservatorio de Salamanca. Actualmente, forma parte del grupo de investigación BISITE (Biomedicina, Sistemas Inteligentes y Tecnología Educativa), también en la Universidad de Salamanca. Sus líneas de investigación se centran en el estudio de plataformas multiagente, *e-learning* y temas relacionados con la creatividad artística.

Emiliana Pizarro Lucas es licenciada en psicología por la Universidad de Salamanca. Actualmente ocupa el cargo de Directora del Servicio de Inserción Profesional, Prácticas y Empleo (SIPE, <http://empleo.usal.es>) de la misma Universidad. Es especialista en competencias profesionales, recursos humanos y relaciones laborales, campos en los que cuenta con multitud de publicaciones y diversos proyectos de innovación dirigidos.



Héctor Quintián Pardo es becario predoctoral y está desarrollando su tesis doctoral en la Universidad de Salamanca. En 2010 obtuvo el título de Ingeniero Industrial por la Universidad de León y un año más tarde realizó el Máster en Computación por la Universidad de La Coruña. Ha participado en 1 proyecto proyectos de I+D+i financiados en convocatorias competitivas del Ministerio de Economía y Competitividad (MIDAS, TIN 2010-21272-C02-01). Es miembro de una COST Action financiada con fondos europeos (AAP for AAL: Algorithms, Architectures and Platforms for Ambient Assisted Living). Es autor de 1 propiedad intelectual denominada GAP Gestor de Aulas de Profesorado (Número de solicitud: C-277-2012). Tiene varias publicaciones en revistas de impacto. Es autor/coautor de 18 publicaciones en congresos internacionales, organizador de 10 congresos internacionales, editor de 6 congresos internacionales y 1 simposio. Es Miembro del IEEE desde 2011, y del IEEE capítulo español; Secretario del IEEE capítulo español desde Marzo de 2013. Además es miembro del IBSAL y del Grupo de Investigación BISITE. Sus líneas de investigación se centran en el estudio de las Redes Neuronales, con especial énfasis en la Búsqueda Exploratoria de Proyecciones, Maximum Likelihood Hebbian Learning, Mapas Auto-Organizados, Redes SOM, Sistemas de Clasificación Múltiple y Sistemas Híbridos. Además de la optimización de procesos industriales y energéticos.

William Raveane obtuvo una Licenciatura en Física por Southern Oregon University en Estados Unidos, y un Máster en Sistemas Inteligentes por la Universidad de Salamanca en España. Trabaja como programador principal en una compañía en Tokio, Japón, desarrollando sistemas *cloud* para aplicaciones web y procesamiento de imágenes. Sus principales líneas de investigación son las Redes Neuronales Profundas, en especial su uso en la visión artificial.

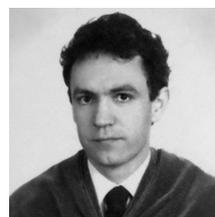




Juan Agustín Rodríguez Valle es Graduado en Informática por la Universidad Pontificia de Salamanca en 2012 y máster en Sistemas Inteligentes por la Universidad de Salamanca en 2013. Ha formado parte del Club de Innovación de la Universidad Pontificia de Salamanca participando como co-autor, diseñador y desarrollador en el proyecto Apolo destinado al control domótico para personas invidentes. También ha participado en el proyecto

PixKid financiado por la Junta de Castilla y León, un juego para niños diseñado para la plataforma iPad. Sus principales líneas de investigación giran en torno a la computación ubicua.

Ángel Luis Sánchez Lázaro es Doctor en Informática por la Universidad de Valladolid. Actualmente desarrolla su actividad académica como Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca (Lenguajes y Sistemas Informáticos). Desde el año 2010 es director del Departamento de Informática y Automática. Es miembro del Grupo de Investigación en Minería de Datos (MiDa), grupo reconocido de la Universidad de Salamanca. Sus principales líneas de investigación son las Aplicaciones de Minería de Datos, la Minería Web Semántica y los Modelos de estimación *software*.



Roberto Therón cursó sus estudios de Informática en la Universidad de Salamanca (Diplomatura) y la Universidad de la Coruña (Licenciatura). Tras entrar a formar parte del Grupo de Investigación en Robótica de la Universidad de Salamanca, presentó su trabajo de Tesis recibiendo el Premio Extraordinario de Doctorado. Posteriormente ha obtenido los títulos de Licenciado en Comunicación Audiovisual (Universidad de Salamanca) y Licenciado en

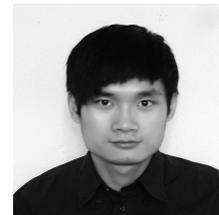
Humanidades (Universidad de Salamanca). En la misma Universidad de Salamanca continúa realizando su trabajo de investigador, como encargado del grupo VisUsal (dentro del Grupo de Investigación Reconocido MiDa) que se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París) o el Centro de Analítica Visual Avanzada de la ONU (Suiza). Es autor de más de 70 artículos en revistas y congresos internacionales.

Daniel Toimil Martín nació en 1987 en Avilés, España. Es Ingeniero Técnico Informático por la Universidad de Oviedo, 2009, donde posteriormente obtuvo el título de Ingeniero Informático en 2012. Actualmente es Máster en Sistemas Inteligentes por la Universidad de Salamanca. Sus líneas de investigación son el Análisis Estadístico y la Minería de Datos, en las cuales trabaja como investigador en la Universidad de Oviedo.



Roberto Vega Ruiz. Nacido en Valladolid el 1 de diciembre de 1990. Graduado en Ingeniería Informática con Mención en Ingeniería del *Software* por la Universidad Pontificia de Salamanca en el año 2012. Un año más tarde, obtuvo el Máster en Sistemas Inteligentes por la Universidad de Salamanca, trabaja en esta entidad como investigador dentro del grupo BISITE en el proyecto MIDAS financiado por el Ministerio de Economía y Competitividad. Además es miembro del IBSAL. Ha gestionado, coordinado y desarrollado diversos proyectos de investigación. Ha participado en la organización de diversos congresos científicos (HAIS, SOCO, ICEUTE y CISIS). En febrero de 2013 recibió el Premio Innovadores Proyecto Joven 2012, otorgado por el diario “El Mundo” en Valladolid. Las líneas de investigación que trata son la Inteligencia Artificial y las Tecnologías Móviles.

Zhangxian Zheng es licenciado en Ingeniería Informática por la Universidad de Shenyang en China, máster en Sistemas de Información Digital por la Universidad de Salamanca en 2012 y máster en Sistemas Inteligentes por la misma Universidad en 2013. Sus principales líneas de investigación se centran en los Sistemas de Información y en el Análisis de la Información generada en los mismos, con especial hincapié en las redes sociales.



El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. Como colofón a las diferentes tareas de investigación que se llevan a cabo durante el Máster, el Departamento de Informática y Automática organiza un *workshop* en el que se da la oportunidad a los estudiantes de presentar los principales resultados de sus Trabajos de Fin de Máster y obtener una realimentación del interés de los mismos.

Editores:

Juan Cruz Benito
Alicia García Holgado
Sergio García Sánchez
Daniel Hernández Alfageme
María Navarro Cáceres
Roberto Vega Ruiz

ISBN - 10 84-695-8670-X
ISBN - 13 978-84-695-8670-9

Departamento de Informática y Automática.
Facultad de Ciencias, Universidad de Salamanca
<http://informatica.usal.es>