

Moodle 2.0 y las nuevas plataformas de aprendizaje orientadas a servicios¹

Miguel A. Conde

Departamento de Informática y Automática. Instituto Universitario de Ciencias de la Educación (IUCE). Grupo de investigación GRIAL. Universidad de Salamanca
mconde@usal.es

Alberto del Pozo

Departamento de Informática y Automática. Instituto Universitario de Ciencias de la Educación (IUCE). Grupo de investigación GRIAL. Universidad de Salamanca
delpozo1988@usal.es

Francisco J. García

Departamento de Informática y Automática. Instituto Universitario de Ciencias de la Educación (IUCE). Grupo de investigación GRIAL. Universidad de Salamanca
fgarcia@usal.es

Resumen

El imparable avance de las nuevas tecnologías ha puesto de manifiesto la necesidad de actualización de las plataformas de aprendizaje. Esa actualización se basa en la incorporación de nuevas funcionalidades para satisfacer las nuevas necesidades de los usuarios. Una de las formas en que poder llevarlo a cabo pasa por la evolución de los entornos de aprendizaje hacia las Arquitecturas Orientadas a Servicios (SOA, *Service Oriented Architecture*). Estas arquitecturas, implementadas principalmente en forma de servicios web, permitirán la creación tanto de clientes como de herramientas externas que puedan interoperar con la plataforma, ampliando sus posibilidades y proporcionando una libertad de movimiento a los usuarios que antes no tenían. Moodle 2.0 será un ejemplo de esa evolución y en el presente artículo se verán alguna de las nuevas posibles aplicaciones a incorporar.

1. Motivación

El gran avance tanto científico como tecnológico que se está produciendo en la sociedad durante los últimos años ha provocado que se deban de buscar nuevas formas de hacer llegar la enseñanza a todos los usuarios [6, 7]. El eLearning supone un planteamiento diferente de los procesos de aprendizaje y por tanto introduce nuevas necesidades. Estas van a verse respaldadas por las plataformas de aprendizaje. Mientras que hace tan solo unos años el poder acceder al aprendizaje a

través de un ordenador era considerado un gran avance tecnológico, ahora el poder acceder a él a través de estos dispositivos se queda corto, creando la necesidad de tener toda la información que se desee cuando se desee, es decir, tener a mano en todo momento el conocimiento. Sin embargo no siempre las plataformas serán capaces de satisfacer estos requisitos.

Otro problema de estas plataformas es que son demasiado genéricas, poco adaptadas a circunstancias especializadas y son pocos escalables modularmente [5] dificultando de esta manera la posibilidad de añadirle funcionalidades propias a la plataforma, limita el desarrollo y la sostenibilidad de los LMS (Learning Management Systems, plataformas de aprendizaje). Añadido a todo esto debe considerarse que, en la mayor parte de los casos, la potencia de los LMS no se aprovecha totalmente, es decir, muchas de las funcionalidades de las plataformas de aprendizaje son ignoradas convirtiéndose en muchos casos en meros contenedores de recursos [4][12][20].

Por eso surgen una serie de proyectos que pretenden dotar de cierta independencia a estas plataformas de eLearning (por ejemplo Moodle) para que de esta manera dejen de ser plataformas monolíticas y cerradas, es decir, de difícil evolución y cuya interacción con otros sistemas sea mínima. Se busca concretamente aportar capacidad de evolución y crecimiento independiente de la tecnología. Esta independencia se consigue mediante la implementación de unos servicios web que permitan la conexión con la plataforma de una manera externa, aunque antes de llegar a los

Tabla 1.

¹ Este trabajo está subvencionado por el Ministerio de Industria Turismo y Comercio (proyecto TSI-020302-2009-35 y proyecto TSI-020302-2009-35) y por la Junta de Castilla y León a través del proyecto de excelencia GR47.

servicios web, se debe comprender como se estructuran y se fundamenta su uso, esto es, las arquitecturas SOA

Este artículo por lo tanto está organizado de la siguiente forma. El punto 2 habla acerca de SOA y sus posibilidades, entrando también en la relación entre SOA y los servicios web. El punto 3 analiza la inclusión de los servicios web dentro de la nueva versión de Moodle, la 2.0, que estará disponible próximamente trayendo consigo bastantes novedades. Además explicando de manera detallada un servicio y una herramienta de backoffice creada para su prueba, explicando también algunos trabajos relacionados con los servicios web.

2. SOA

De cara a la comprensión del presente artículo es preciso definir qué es la Arquitectura Orientada a Servicios. Se trata un paradigma o concepto de arquitectura de software que se basa en la creación de un conjunto de servicios, de diferente granularidad, entre los procesos de negocio y las aplicaciones [18]. Esta arquitectura tiene como objetivos principales:

- Modelar la lógica de negocio como servicios para poder expresar la capa de negocio mediante la facilidad que ofrece la orquestación de los mismos.
- Crear una capa de servicios que ofrezca la funcionalidad de la capa de aplicación de forma independiente de la tecnología que la soporta.
- Minimizar las dependencias entre la capa de negocio y la de aplicación para desacoplar el negocio de la tecnología, y de este modo permitir los cambios en cualquiera de ellas. El objetivo sería favorecer la agilidad para el negocio.

Para comprender claramente el concepto de Arquitectura Orientada a Servicios en primer lugar debe considerarse qué significa el concepto de Arquitectura en el ámbito del software. Según la IEEE-STD-1471-2000 (“Práctica Recomendada Para La Descripción Arquitectónica De Sistemas De Software Intensivos”) la arquitectura es “la organización fundamental de un sistema integrado en sus componentes, la relación entre ellos y el medio, y los principios que establecen su desarrollo y evolución” [11].

También puede entenderse una arquitectura como “la estructura global del software y las formas en que esa estructura proporciona integridad conceptual a un sistema [19] o bien como “la estructura lógica y física de un sistema, forjada por todas las decisiones de diseño estratégicas y tácticas aplicadas durante el desarrollo”[3]. En cualquiera de las definiciones aportadas se considera la arquitectura como la organización estructural de los componentes de un sistema, este tipo de organizaciones van a evolucionar vinculadas al planteamiento del modelo de negocio que a su vez se verá condicionado por la evolución de la tecnología. Como ejemplo de esas evoluciones puede observarse como las estructuras de negocio cambian de un modelo de jerarquización vertical, a uno horizontal y multidisciplinar, para posteriormente pasar un modelo de tipo ecosistema en el que las áreas implicadas no se comunican solo entre las áreas más próximas.

Una vez explicado el término Arquitectura, se tiene que conocer también el de servicio. Existen diferentes planteamientos acerca de lo que son los servicios. Pueden considerarse desde un punto de vista de negocio como “Una funcionalidad construida como un componente reusable para ser utilizado en un proceso de negocio” [9] o desde un punto de vista técnico como “elementos autodescritos e independientes de plataforma que soportan la composición rápida, barata y distribuida de aplicaciones” [14].

En cualquiera de los casos los servicios serán provistos por aplicaciones o proveedores de servicios de cara a proporcionar una funcionalidad sin desvelar su implementación. Esto supone la definición de una interfaz para su publicación. Teniendo en cuenta todo esto, las características deseables para un servicio serían:

- Tecnológicamente neutral. Los Servicios no solamente deben ser independientes de cualquier tipo de implementación sino que deben de poder ser invocados de una forma lo más estándar posible.
- Bajo acoplamiento. No debe requerirse ningún conocimiento de la estructura interna o externa del servicio en ningún contexto del mismo (ni en el lado del cliente ni en el del servidor).
- Descubrimiento transparente. La funcionalidad expresada por los servicios

debe estar disponible en diferentes repositorios para su localización y difusión.

- Calidad de servicios. Los servicios deben ser capaces de ofrecer la funcionalidad expresada con un nivel cualitativo adecuado de cara a que su reutilización sea posible en diferentes contextos.
- Debe existir la posibilidad de trabajar con un servicio de forma individual o formando parte de una composición de estos en busca de una funcionalidad más completa.

La idea principal, por tanto, de esta arquitectura es que conviene ordenar la forma en que se comunican las distintas partes de un sistema. Para conseguir este objetivo se define una capa de servicios con el que los sistemas interactúan, evitando así el trabajar de manera directa entre ellos, favoreciendo la interoperabilidad y la escalabilidad. De esta manera, si dos sistemas desean comunicarse, no necesitan conocer el funcionamiento del otro, sino que utilizarán esta capa de servicios como intermediaria, la cual sí conoce como funcionan estos sistemas. Si en algún momento se desea sustituir o realizar algún cambio en alguno de los dos sistemas, la acción sería independiente del otro, ya que se han desarrollado de manera que no dependan del otro sistema, sino que sólo dependen de los datos que devuelven [1]. Esta forma de relacionar componentes aporta las siguientes ventajas: i) Permite sustituir componentes individuales sin que eso afecte a otros componentes ii) Todos los sistemas se conectan al bus de la misma forma, con lo que se gana en homogeneidad iii) Facilidad en la operación y mantenimiento iv) Arquitectura sencilla, robusta y escalable.

2.1. SOA y los servicios Web

Es importante conocer que SOA no es un sinónimo de servicios web. Mientras que SOA es un paradigma de desarrollo (y estrategia de TI), los servicios web son una de las posibles tecnologías que se pueden utilizar para implementar SOA, aunque hay que destacar que la implantación de SOA está siendo mucho más rápida gracias a los servicios web y que éstos se están convirtiendo en el estándar de facto para la implementación de estas arquitecturas.

Una vez vista la diferencia entre SOA y los servicios web, se va a tratar de describir este último. Cabe destacar que existen varias posibles definiciones acerca de lo que son los servicios web, lo que muestra su complejidad a la hora de dar una explicación adecuada de todo lo que son e implican. Una posible sería hablar de ellos como un conjunto de aplicaciones o de tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a esos procedimientos a través de la web [21]. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. El uso de los servicios web proporciona una serie de ventajas (muchas de las cuales derivan de las ventajas de implementar una Arquitectura Orientada a Servicios) como son:

- Promueven la interoperabilidad ya que la interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y del lenguaje.
- Fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Reducen la complejidad por medio del encapsulamiento, ya que los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementando el servicio por parte del proveedor, y éste, a su vez, no sabe cómo utiliza el cliente el servicio.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos ya que las especificaciones son gestionadas por una organización abierta, la W3C.

2.2. Arquitectura SOA en eLearning

De cara a incorporar interoperabilidad en las plataformas de aprendizaje y hacer a estas flexibles y escalables es necesaria la definición de una nueva generación de plataformas de aprendizaje. Este tipo de plataformas se van a asentar en las Arquitecturas Orientadas a Servicios. Este tipo de solución proporcionará una separación entre la interfaz de un servicio y su implementación subyacente. No será relevante que una aplicación que se quiera conectar a una plataforma esté implementada en una tecnología diferente del core del LMS. SOA aporta independencia en la evolución del software permitiendo la incorporación de nuevas funcionalidades independientemente de la versión del LMS subyacente.

Dentro de las posibles aplicaciones de las arquitecturas SOA deberían considerarse diferentes enfoques:

- Uso de las arquitecturas SOA para proporcionar información a contextos externos. Como podría ser el uso de arquitecturas orientadas a servicios para búsquedas semánticas de información sobre la plataforma de aprendizaje, como el proyecto LUISA [10].
- Adaptación ligera de plataformas de aprendizaje a otras aplicaciones, como puede ser con servicios de autenticación o herramientas de comunicación administrativa y *backoffice* [15]. Generalmente son adaptaciones parciales y no necesariamente tienen que integrarse transparentemente.
- Vinculaciones completas entre las plataformas y las aplicaciones en las que la integración es totalmente transparente para los usuarios, permitiendo una comunicación bidireccional y aportando un modo de presentación totalmente adaptado al LMS. Para esto se proponen varias especificaciones entre las que destacan IMS LTI (*Learning Tools for Interoperability*), para integración transparente de aplicaciones en las plataformas o los OSIDs (*Open Service Interface Definitions*) del proyecto OKI (*Open Knowledge Initiative*) [13], que describen medios de comunicación entre la plataforma y otras herramientas.

- Adaptaciones mixtas, en las que las aplicaciones requieren de comunicación con el LMS para extraer información e incorporar información al mismo, pero sin que tengan que estar incorporados en ellos, como podrían ser clientes móviles para las plataformas de aprendizaje. En este sentido cabe destacar Moodbile [2], como ejemplo de integración.
- En cualquiera de los casos lo que se pretende es proporcionar nuevas funcionalidades a las plataformas de aprendizaje haciéndolas evolucionar de un modelo monolítico destinado a la extinción a un modelo evolutivo, escalable y flexible.

3. Servicios Web en Moodle 2.0

A lo largo del 2010 está previsto el lanzamiento de la nueva versión de Moodle. Esta nueva versión, la 2.0, que sustituirá a la actual 1.9, se ha visto como una oportunidad de realizar las cosas de manera distinta, de darle un cambio radical a la plataforma y adaptarla a las a las tecnologías que están inundando el mercado de las telecomunicaciones. Muchos de estos cambios son el soporte para repositorios externos (Picasa, Youtube, Flickr, Wikimedia, etc.), nuevos módulos y bloques, cambios en el código del core de Moodle (aunque se podrá actualizar sin problemas de la versión 1.9 a la 2.0), etc., pero el cambio más importante que aquí interesa es el de soporte para servicios web. Estos servicios web (en cuyo proyecto participa uno de los autores de este artículo) permitirán ampliar enormemente las posibilidades de Moodle, pasando de ser una plataforma monolítica a una aplicación escalable y con la cual se puede interoperar. De esta manera se podrían solucionar algunos problemas u obstáculos que habían ido apareciendo con el tiempo, como son:

- La aparición de nuevos dispositivos móviles con conexión a Internet, con diferentes interfaces y compatibilidades distintas. Muchos de estos navegadores permiten navegar por la aplicación, pero debido a sus limitaciones físicas ésta puede llegar a ser realmente compleja. Por lo tanto, sabiendo que el número de estos dispositivos está creciendo, es conveniente que Moodle

facilite la creación de interfaces alternativas adaptadas a estas plataformas.

- El gran número de organizaciones que confían en Moodle como su plataforma de eLearning ha aumentado en los últimos años, lo que provoca cambios en los requerimientos del sistema (escalabilidad) y diversificación de los mismos, ya que continuamente surgen nuevas necesidades a satisfacer, como son nuevas funcionalidades (siempre y cuando no se corrompan los principios pedagógicos de Moodle). Aún así, hay que adaptar Moodle a los sistemas de información de las organizaciones donde se implementa.
- Integración con *Backoffice*. Se ofrece la posibilidad de, por ejemplo, crear una aplicación que pueda interactuar con varios sistemas a la vez, ejecutando una misma acción en varios lugares, haciendo que todo funcione de forma conjunta, evitando tener que realizar las operaciones varias veces, como se ve en la Figura 1.

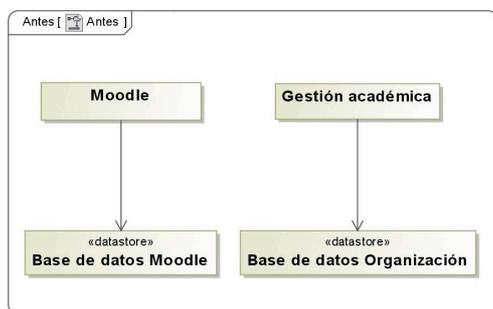


Figura 1. Gestión de Moodle y una base de datos de una organización antes de la utilización de los servicios web.

Utilizando los servicios web se pasaría a realizarlas tan sólo una vez, dentro de una herramienta de *backoffice*, la cual se encargará de realizar las operaciones pertinentes tanto en la base de datos de la organización, como en Moodle mediante los servicios web, sin necesidad de modificar o acceder al código, como se ve en la Figura 2.

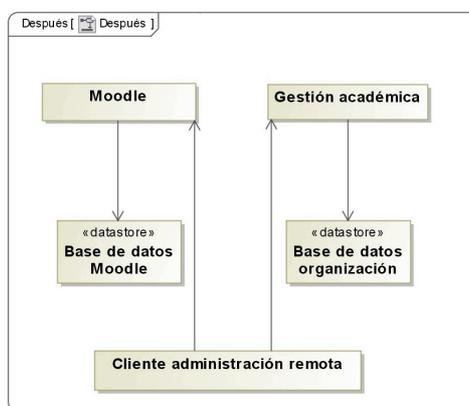


Figura 2. Gestión de Moodle y una base de datos de una organización después de la implementación de los servicios web, utilizando una herramienta de backoffice.

3.1. Planteamiento Arquitectónico

El primer paso dentro del desarrollo de los servicios web fue el de definir una arquitectura que permitiera garantizar la interoperabilidad que se les presupone. Por ello, dichos servicios web debían de cumplir una serie de requisitos establecidos por el equipo de desarrollo de Moodle [17]:

- Los servicios web deben ser accesibles desde cualquier sistema de conexión, tanto actual como futura, y deben de poder ser invocados independientemente del lenguaje utilizado para ello (interoperabilidad).
- La estructura de los servicios web debe de desarrollarse de tal manera que aunque se realicen cambios dentro del *core* de Moodle, sea necesario realizar pocas o ninguna modificación en la API.
- Las funciones que conforman la API deben ser ampliables para favorecer las contribuciones.
- El servicio web debe adaptarse al sistema de privilegios de Moodle (*capabilities*) para garantizar la seguridad.
- Atendiendo a esta serie de requisitos, los servicios web de Moodle 2.0 están divididos en tres capas fundamentales [16] que pueden observarse en la Figura 3 y se describen posteriormente.

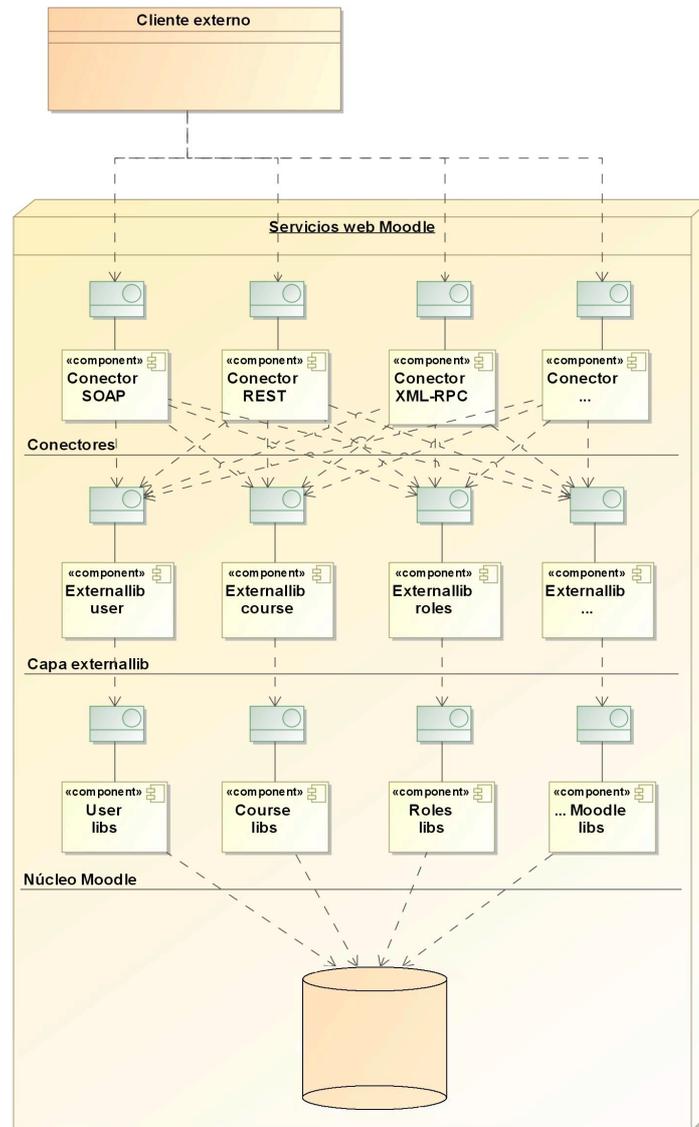


Figura 3. Capas de los servicios web de Moodle 2.0

1. Conectores. Hasta el momento se puede conectar con la plataforma a través de 4 protocolos, estos son: REST, SOAP, XML-RPC y AMF (Flash). Además, se está trabajando en un quinto conector, JSON, el cuál es bastante más rápido que los cuatro ya nombrados en lo que al intercambio de datos se refiere. Cada uno

de estos protocolos posee su propio conector, los cuales se encargarán de recibir la petición desde el exterior, comprobar si la función a la que se desea acceder existe, comprobar los permisos del usuario que la invoca (y en caso de que sea necesario, generar o devolverle el token que le va a identificar durante la sesión, aunque

también se puede conectar con la plataforma enviando el nombre de usuario y la contraseña en cada invocación). Una vez hecho esto, los conectores parsean los datos y llamarán a la función correspondiente, pasando de esta manera a la capa externa. Los conectores admiten plugins así que se podrán añadir fácilmente nuevos conectores para que los sistemas externos puedan conectar con Moodle utilizando protocolos distintos a los que la plataforma trae de serie.

2. **Externallib.** Esta capa está formada por un conjunto de ficheros denominados `externallib.php` los cuales se encuentran expandidos por todo el árbol de directorios de Moodle. Dichos ficheros son llamados desde los conectores y en ellos se encuentran todas las funciones que se ofrecen en la API de los servicios web. Es decir, resumen todas las funcionalidades de Moodle para ofrecerlas al exterior, intentando utilizar para ellos el mayor código posible existente dentro de la plataforma para no crear código innecesario o duplicado. De esta manera, el `externallib.php` de `users` se encuentra dentro de la carpeta `user` de Moodle, y contiene una serie de funciones que permiten gestionar los aspectos relacionados con los usuarios de la plataforma (como ya se explicará más adelante). Como es lógico, antes de comenzar a realizar cualquiera de las acciones, se chequean los permisos del usuario con relación a la acción a realizar, y se comprueban además los parámetros recibidos y que se van a devolver. Esto último se consigue gracias a que en estos ficheros se encuentran además una serie de funciones que indican los parámetros que acepta y que debe devolver cada función de la API. Por último, indicar un problema con el que se han encontrado los desarrolladores: el tratamiento de errores. La solución actual consiste en el lanzamiento de excepciones cada vez que haya un error, cambiando de esa manera la idea inicial de devolver cadenas de texto. Este tratamiento de errores ha sido posible finalmente gracias a los requisitos mínimos de Moodle 2.0, que necesita de PHP 5 para funcionar (las excepciones entraron en PHP en esta versión), por lo que si se desean utilizar los servicios web en versiones anteriores (hay un proyecto que está trabajando en el backporting de los servicios web) será necesario

actualizar la versión de PHP si se tiene versiones anteriores a la 5.

3. **Núcleo de Moodle.** La capa de núcleo de Moodle está formada por todas las librerías que contienen funciones que puedan interesar dentro de la capa de los `externallib`, es decir, funciones relacionadas con los usuarios, los cursos, los grupos, etc. Esta capa ha sido mejorada en Moodle 2.0 porque muchas de estas funciones imprimían mensajes de error en pantalla cuando había algún problema, por lo que se han reescrito parte de estas funciones del núcleo para que en caso de error devuelvan excepciones (hasta ahora Moodle no poseía una API y gracias a estos cambios se está generando una).

3.2. Ejemplo de descripción y uso de un servicio web

Dentro de este apartado se va a explicar la estructura interna de un servicio perteneciente a los servicios web de Moodle, más concretamente la parte correspondiente a la capa externa del servicio *logging*, encargado de la gestión de la actividad (*logs*) dentro de la plataforma. Los diagramas utilizados para la explicación del servicio son diagramas creados mediante SoaML, un proyecto de especificación *open source* que describe un perfil y metamodelado UML para el diseño y modelado de las arquitecturas orientadas a servicios.

3.2.1. Arquitectura y funciones del servicio

En primer lugar se representa la arquitectura del servicio, que indica la relación entre el consumidor del servicio y su proveedor, en este caso el paquete *logging* con su correspondiente `externallib.php`, estando de mediador el contrato del servicio. El diagrama utilizado para la representación de esta información es el llamado diagrama de arquitectura del servicio (*Service Architecture Diagram*) y se tiene un ejemplo en la Figura 4. Mientras que ese diagrama no ofrece una información demasiado valiosa dentro de la documentación de los servicios web, el diagrama de servicio (*service diagram*, Figura 5) representa de una manera clara las funciones que la API de *logging* va a ofrecer como servicio web

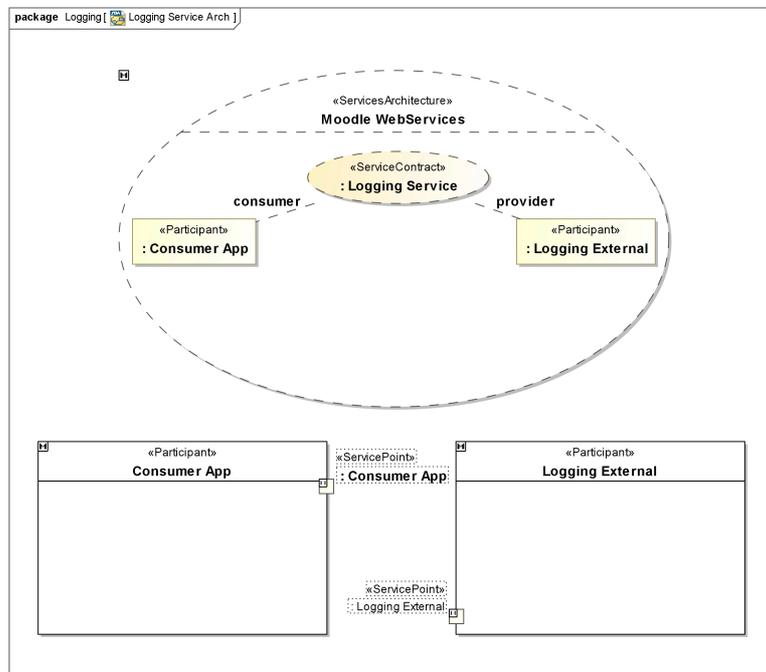


Figura 4. Diagrama de arquitectura de servicio (logging).

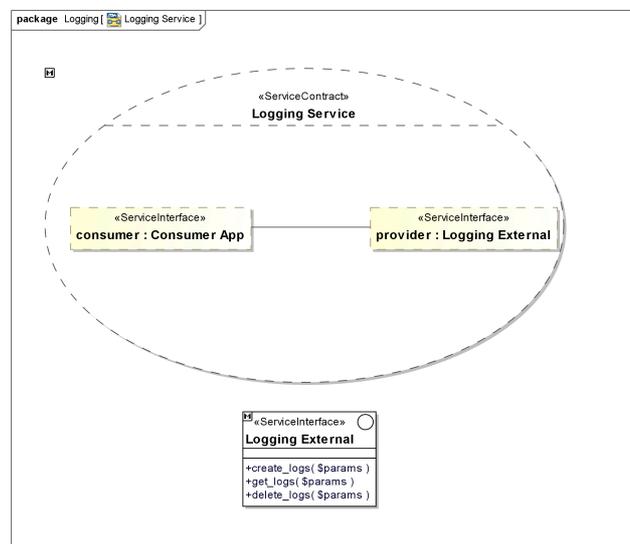


Figura 5. Diagrama del servicio (logging).

En el diagrama de la Figura 5 cabe destacar la representación de las 3 funciones que la API de los servicios web va a ofrecer en relación a la parte de logging, estas son: i) `create_logs`, que permitirá crear logs dentro de la base de datos de la plataforma, y por lo tanto ofrecer la posibilidad de registrar acciones importantes que se realicen desde fuera de la plataforma (realmente lo que haría es ofrecer una información más concreta del trabajo realizado, ya que los servicio web ya registran ellos mismos si han sido utilizados) ii) `get_logs`, función que permitirá recuperar una serie de logs de la plataforma. Esta función permite recuperar la actividad de Moodle en función del tiempo, del usuario, del curso o del id dentro de la base de datos. En algunas plataformas demasiado grandes, el número de logs es demasiado elevado, por lo que se está trabajando en una función que cuente los logs existentes para que en caso de que sea un número muy elevado éstos se puedan devolver por bloques, evitando así problemas de memoria iii) `delete_logs`, ésta es una función que no debería ser muy utilizada ya que sirve para borrar cierta actividad dentro de la plataforma.

3.2.2 Parámetros de las funciones

El tercer diagrama que documenta un servicio web consiste en uno llamado diagrama de mensajes (message diagram) que complementa la información ofrecida en el diagrama anterior, indicando los parámetros que cada una de las funciones del servicio acepta y devuelve en cada ejecución. Un ejemplo de este diagrama se puede ver en la Figura 6. En ella que se observan los mensajes que los consumidores van a intercambiar con el servicio. En la mayor parte de los casos, las funciones deben recibir un array de arrays, donde cada uno de estos segundos arrays contiene la información necesaria para una ejecución de la función. De esta manera, se hace posible realizar varias acciones, como dar de alta cien usuarios por ejemplo, utilizando sólo una invocación a la función `create_users`, pero enviándole un array con cien arrays. Dicho esto, en el diagrama se puede observar que la función `get_logs` debe recibir un array de arrays con los parámetros `criteria` (curso, usuario, id, fecha), `data1` y `data2`, mientras que devolverá toda la información existente acerca de los logs dentro de la base de datos. Cabe destacar que para conocer el uso de

los parámetros será necesario leer la documentación de Moodle donde aparece una pequeña descripción de cada uno de ellos con sus posibles valores.

3.3. Aplicación de los servicios web de Moodle, la herramienta de Backoffice

Durante el desarrollo de la capa externa de los servicios web, se decidió realizar una herramienta de prueba que permitiera probar el correcto funcionamiento de las funciones de la API que se iban creando. A la vez, esta herramienta podría permitir observar (desde el punto de vista de los futuros desarrolladores de clientes para Moodle) las posibles necesidades que éstos pudieran tener, para de esta manera poder realizar una API lo más sencilla y usable posible. Teniendo en cuenta que se quería abarcar el mayor número de funciones de la API posibles, se decidió crear una herramienta de backoffice (a pequeña escala) que permitiera a un administrador realizar las gestiones básicas de una plataforma Moodle. Dichas gestiones básicas serían:

- **Administración.** La opción más interesante dentro de la administración de la herramienta es la de poder cambiar el protocolo mediante el cual se quiere conectar con la plataforma Moodle, ya que el cliente creado permite la conexión con tres de los cuatro protocolos que soporta Moodle, estos son: REST, SOAP y XML-RPC. Además, se está trabajando para adaptarla al protocolo JSON, que como se ha dicho anteriormente se encuentra actualmente en desarrollo. El resto de opciones son el envío masivo de mensajes a los usuarios inactivos (se le indica el número de días mínimo que un usuario debe llevar inactivo y automática se manda un mensaje a todos esos usuarios) y la opción de probar directamente las funciones de la API de los servicios web, tan solo indicándole la función que se desea probar y los parámetros que se le van a enviar.
- **Usuarios.** Permite el acceso a todos los usuarios de la plataforma, además de permitir gestiones tales como la creación de nuevos usuarios y la actualización y eliminación de los ya existentes.
- **Cursos.** De la misma forma que con los usuarios, permite la gestión de la

configuración de los cursos, así como de alguna de sus actividades. Es decir se pueden crear, eliminar, modificar y acceder a cursos, así como a las categorías de cursos. Ya dentro de ellos, se puede acceder a sus foros, calificar a los alumnos y gestionar los grupos.

- **Roles.** Una parte importante dentro de Moodle es el sistema de roles, por lo que la herramienta permite la gestión de los roles en los dos contextos más utilizados: sistema y cursos.
- **Logs.** La parte más interesante llega con la gestión de los logs. Además de poder acceder a los logs de la plataforma ya sea mediante usuario, curso o fecha, con motivo de probar la interoperabilidad de los servicios

web, la herramienta incorpora un applet de un herramienta de visualización de logs creada por Diego Alonso Gómez [8] que crea visualizaciones de la actividad de la plataforma accediendo a los logs de la misma. Con anterioridad a los servicios web esta herramienta necesitaba el acceso directo a la base de datos para recoger la información, por lo que se necesitaba un contacto con la plataforma además de tener que acceder al código de la misma. De esta manera, gracias a los servicios web, la herramienta es independiente de la plataforma con la que trabaje, otorgándole una independencia que antes no tenía.

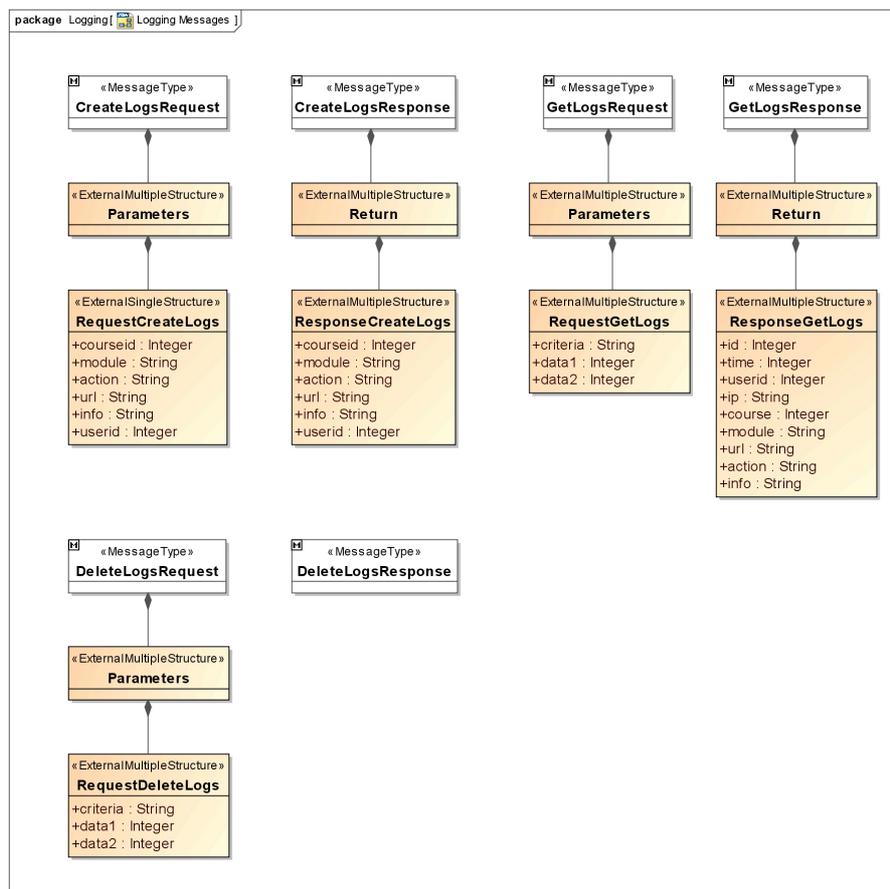


Figura 6. Diagrama de mensajes del servicio (logging).

4. Conclusiones

Los procesos de aprendizaje están cambiando, ya no solo debido a la evolución de las nuevas tecnologías, sino también a cambios sociológicos como pueden ser la irrupción de las tendencias sociales favorecidas por las herramientas 2.0; las nuevas necesidades formativas cada vez más orientadas hacia el alumno; los nuevos contextos y situaciones en los que los alumnos aprenden; el hecho de que los receptores de la información han nacido en la era digital y por tanto utilizan la tecnología a un nivel antes no contemplado; los nuevos paradigmas formativos derivados de propuestas como la normativa de Bolonia que aboga por el reconocimiento de aprendizaje tanto formal como no formal; etc.

Ante esta situación las plataformas de aprendizaje, que han sido las herramientas más extendidas y utilizadas de los últimos años están en peligro de extinción. Cuando estas herramientas han alcanzado un mayor grado de madurez, y aún a pesar de que su potencial no se aproveche, surgen nuevas necesidades y éstas suponen la evolución. Esa evolución conduce a nuevos contextos, paradigmas y orientaciones en lo que se refiere al aprendizaje, y para ello son necesarias las arquitecturas SOA.

Las Arquitecturas Orientadas a Servicios serán una puerta que permite incrementar la funcionalidad de las plataformas de aprendizaje, proporcionando por tanto un medio para evitar su estancamiento, permitiendo abrirse camino hacia las nuevas tendencias como pueden ser los entornos de aprendizaje personalizados (PLEs).

Moodle 2.0 y las herramientas que se han definido para su testeo son un claro ejemplo de cómo una plataforma puede ir más allá y puede realmente ser escalable y flexible gracias a la base que aportan los servicios web. Utilizando las Arquitecturas Orientadas a Servicios sobre este conocido LMS se ha conseguido proporcionar un medio para enriquecer las funcionalidades, existían ya trabajos en este sentido pero en cualquiera de los casos suponía tener que acceder directamente al código de Moodle y se generarían dependencias tecnológicas y de versión. De este modo, como demuestran los ejemplos descritos, se abre un nuevo mundo para este LMS. Lo que se consigue con este tipo de aplicaciones será por tanto la evolución de la “especie”, que no es otra que los medios que el alumno utiliza para

aprender cómo son las plataformas de aprendizaje y los contextos a qué estas pueden aplicarse.

Referencias

- [1] Alba, J. ¿Qué es SOA - Arquitectura Orientada al Servicio. bit, 167, 2008.
- [2] Alier, M. and Casany, M. J. Moodle: Extending Moodle to the Mobile on/offline Scenario. In Proceedings of the IADIS International Conference Mobile Learning. Algarve, 2008.
- [3] Booch, G. Object Oriented Analysis and Design with Applications. The Benjamin/Cummings Publishing Company, 1994.
- [4] Cuban, L. Oversold and underused: Computers in the classroom. Harvard University Press, Cambridge, MA, 2001.
- [5] Fontenla, J., Caeiro, M. and Llamas, M. Una Arquitectura SOA para sistemas de e-Learning a través de la integración de Web Services. V Congreso Iberoamericano de Telemática. CITA 2009. Gijón/Xixón. Mayo, 2009.
- [6] García Peñalvo, F. J. Estado Actual de los Sistemas E-Learning. Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, 6, 2. Octubre, 2005.
- [7] García Peñalvo, F. J. Preface of Advances in E-Learning: Experiences and Methodologies. Information Science Reference, Hershey, PA, USA, 2008.
- [8] Gómez, D. A., Sánchez, R. T. and García, F. J. Semantic Spiral Timeline como apoyo al e-learning. Journal of Universal Computer Science (j-jucs), 2008.
- [9] Keen M. et al. Implementing Technology to Support SOA Governance and Management. IBM RedBooks. 2007.
- [10] LUISA Learning Content Management System Using Innovative Semantic Web Services Architecture. 2009. <http://luisa.atosorigin.es>. Última vez consultado 12/07/2010
- [11] Maier, M. W., Emery, D. and Hilliard, R. IEEE 1471 and systems engineering. Syst. Eng., 7, 3 2004), 257-270.
- [12] Milligan, C. The Road to the Personal Learning Environment? CETIS, 2006. <http://zope.cetis.ac.uk/members/ple/resources/colimilligan.pdf>, Última vez consultado 12/07/2010
- [13] OKI. OKI Project. <http://www.okiproject.org/>. Última vez consultado 12/07/2010.

- [14] Papazoglou, M. P. Service -Oriented Computing: Concepts, Characteristics and Directions. In Proceedings of the Proceedings of the Fourth International Conference on Web Information Systems Engineering. IEEE Computer Society. 2003.
- [15] Pätzold, S., Rathmayer, S. and Graf, S. Proposal for the Design and Implementation of a Modern System Architecture and integration infrastructure in context of e-learning and exchange of relevant data, 2008.
- [16] Piguillem, J. Moodle 2.0 Web Services architecture, 2009. <http://blogs.dfwikilabs.org/pigui/2009/04/27/moodle-20-web-services-architecture/>. Última vez visitado 12/07/2010.
- [17] Recio, F. Arquitectura de los webservices de Moodle, 2008. http://blogs.dfwikilabs.org/moodle_ws/2008/04/10/arquitectura-de-los-webservices-de-moodle/. Última vez visitado 12/07/2010
- [18] Serverlabs ¿Pero, qué es realmente SOA? , <http://www.theserverlabs.com/folletos/Folleto%20SOA.pdf>. Última vez visitado 12/07/2010.
- [19] Shaw, M. and Garlan, D. Software architecture: perspectives on an emerging discipline. Prentice-Hall, Inc., 1996.
- [20] Universidad de Carolina del Norte. Sakai Pilot Evaluation Final Report. 2009 <http://www.unc.edu/sakaipilot/evaluation/FinalReport-Oct15-09-sm.pdf>. Última vez consultado 12/07/2010.
- [21] W3C Guía Breve de Servicios Web, 2010. <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>. Última vez consultado 12/07/2010