

ClayNet: Adaptación de contenidos en m-learning

Alberto Velasco¹, Jorge Carabias¹, Miguel Ángel Conde¹, Francisco José García²

¹ Departamento de I+D+i CLAY Formación Internacional,
C/Hoces del Duratón nº 57, 37008, Salamanca, España
{a.velasco, jorge, miguelangel}@clayformacion.com
<http://www.clayformacion.com/>

² Dpto. Informática y Automática – Universidad de Salamanca
Colaborador de Clay Formación Internacional
Plaza de los Caídos S/N 37008, Salamanca, España
fgarcia@usal.es

Resumen

El m-learning surge como la evolución natural del e-learning hacia el empleo de dispositivos móviles. Por ello, ClayNet, como plataforma de e-learning, busca dar este paso a través del desarrollo de un servicio web que permita el acceso a los recursos de la plataforma a los usuarios de estos dispositivos. Los recursos se adaptarán a las características técnicas del terminal utilizado para así minimizar la cantidad de datos transmitidos y optimizar su representación en pantalla.

1. Introducción

Se denomina m-learning, o aprendizaje electrónico móvil, a una metodología de enseñanza y aprendizaje que se vale del uso de pequeños dispositivos móviles, tales como teléfonos móviles o agendas electrónicas, en definitiva, todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica.

Las soluciones de m-learning ofrecen la libertad de capturar pensamientos e ideas de manera espontánea, justo cuando la inspiración llega, y permiten acceder a las tecnologías de la información cuando y donde el usuario lo necesite, facilitando la posibilidad de implementar innovadores modos de dar clase y aprender [3].

El rápido desarrollo de la tecnología de los dispositivos móviles y de los servicios que la telefonía móvil ofrece facilita el desarrollo de nuevas aplicaciones más sofisticadas, con la consiguiente demanda por parte de los consumidores.

Tal es la importancia que están teniendo estos dispositivos en la sociedad actual, que las organizaciones proveedoras de formación se han visto en la necesidad de producir contenidos específicamente dirigidos a los dispositivos móviles puesto que se trata de un mercado con millones de usuarios y en constante crecimiento.

2. ¿Qué es ClayNet?

ClayNet es una plataforma de aprendizaje que cubre el ámbito desde un apoyo a la educación presencial hasta el desarrollo de un aprendizaje totalmente en línea y adaptado y personalizado para cada alumno.

ClayNet se estructura como una aplicación web basada en la idea de portal haciendo uso de la tecnología Java Portlets. Los portlets son mini-aplicaciones web independientes que pueden agruparse e interaccionar para formar un portal. A pesar de estar integrada en un portal, ClayNet es independiente del mismo, lo que supone que pueda funcionar de forma aislada e integrada sobre otras plataformas [2]. Los portlets que componen ClayNet proporcionan las funcionalidades propias de cualquier plataforma de e-learning, y permiten su exportación y adaptación a otros entornos o a las necesidades de los usuarios. En la figura 1 puede observarse el aspecto del portal de ClayNet con varios portlets añadidos.

El uso de la tecnología portlets necesita de una base que soporte su almacenamiento y gestión. Entre las herramientas que implementan contenedores de portlets que cumplan con la especificación JSR 168 [1] se ha elegido Liferay

(<http://www.liferay.com>) que es una herramienta de código abierto que permite la construcción de portales. Los portlets de ClayNet también se apoyan en una base de datos externa para realizar la persistencia de datos. El sistema gestor de base de datos utilizado es MySQL (<http://www.mysql.com>).

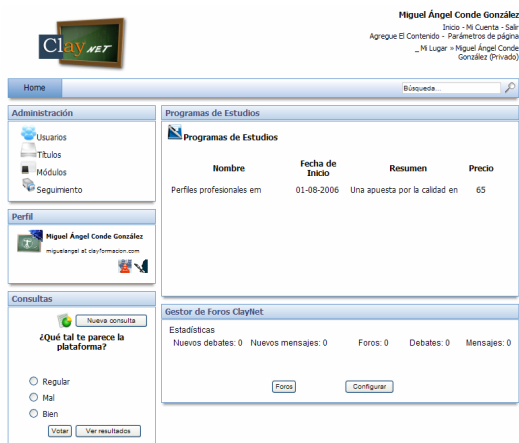


Figura 1. Vista de ClayNet

3. ClayNet y el m-learning

El objetivo principal del proyecto de m-learning en ClayNet es permitir a los usuarios el acceso y la interacción con los recursos de la plataforma de aprendizaje a través de un dispositivo móvil.

Se está desarrollando un servicio web sobre J2EE (*Java 2 Enterprise Edition*) y un cliente de dicho servicio para dispositivo móvil sobre J2ME (*Java 2 Micro Edition*) conforme con la especificación JSR-172 (*J2ME Web Services*). La comunicación entre ambos se llevará a cabo utilizando SOAP (*Simple Object Access Protocol*) sobre HTTP (*HyperText Transfer Protocol*).

El servicio web implementará los métodos necesarios para permitir y controlar la conexión y autenticación de los usuarios desde dispositivos móviles, así como el acceso a los recursos disponibles, adaptándolos a las características particulares del dispositivo utilizado.

El cliente móvil se encargará de invocar los métodos remotos disponibles en el servicio web para realizar distintas tareas: autenticación del usuario en la plataforma, acceso a los recursos

disponibles, sincronización de los cambios realizados en los mismos, etc. Asimismo, se encargará de mostrar los contenidos obtenidos de la plataforma y permitir la interacción del usuario con los mismos.

Debido a que los dispositivos móviles pueden no tener disponible de forma continua una conexión a Internet (zonas sin cobertura) y para minimizar la cantidad de datos transmitidos (actualmente el coste de la conexión a Internet vía móvil es bastante alto) se buscará que el usuario se conecte para acceder a los recursos deseados, los descargue en su terminal, se desconecte, interactúe con ellos y, una vez haya terminado, se conecte de nuevo para sincronizar los cambios con el servidor (ver figura 2).

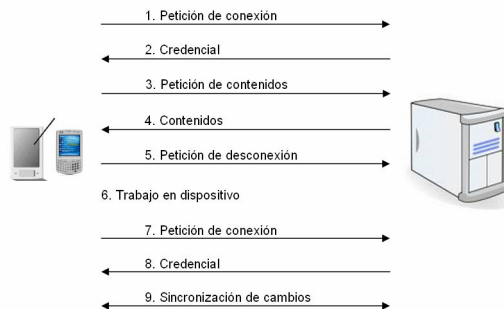


Figura 2. Diagrama de comunicación entre cliente y servidor.

La adaptación de contenidos a las capacidades de cada terminal concreto es otro objetivo importante relacionado con el ahorro de tiempo de conexión y cantidad de datos transmitidos. El cliente enviará al servidor un listado de las características técnicas del dispositivo del usuario tales como: dimensiones y número de colores de la pantalla, tipos de archivos soportados (audio, vídeo, PDF, HTML...), etc. Dicha información se obtiene en tiempo de ejecución consultando diversas funciones de las API de J2ME. Teniendo en cuenta esta información, el servidor realizará una serie acciones para llevar a cabo la adaptación, que dependerá del tipo de recurso.

4. M-learning en ClayNet ahora

Actualmente el proyecto de m-learning en ClayNet está en fase de desarrollo, con el objetivo de obtener una primera versión estable en julio de

2007. Sin embargo, la mayoría de los métodos del servicio web y la funcionalidad básica de la aplicación cliente ya han sido implementados.

4.1. Servicio web

En la parte del servidor, se ha desarrollado un servicio de autenticación y mantenimiento de sesión de usuarios móviles. Éste recibe desde la aplicación del cliente una petición de acceso al sistema que incluye el nombre de usuario y la contraseña indicados por el usuario y un objeto con las características técnicas del dispositivo, de modo que pueda ser asociado con la sesión y así se evita tener que enviarlo en cada petición de contenido. En caso de que los datos sean correctos, el servidor crea un objeto de sesión con un identificador único para ese usuario que actuará a modo de credencial y se lo devuelve a la aplicación cliente. A partir de ese momento, la aplicación cliente deberá enviar al servidor dicho identificador en cada petición que realice para poder así identificar al usuario (ver figura 3).

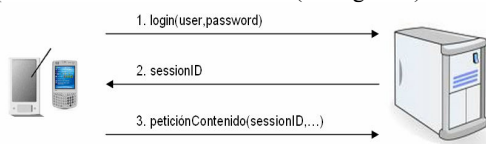


Figura 3. Diagrama de autenticación.

Por su parte, el servicio de contenido está formado por métodos para acceder a los recursos y otros que permiten navegar por la estructura de los cursos. Éstos últimos recuperan la lista de los elementos que están englobados dentro de otro elemento de nivel superior. Por ejemplo, hay un método que devuelve todos los módulos de un curso, otro que devuelve todas las unidades de un módulo, etc. La información de cada elemento incluye su identificador en la base de datos y su nombre.

Hay distintos métodos para solicitar recursos dependiendo en el tipo de recurso deseado. Los recursos se clasifican en cuatro tipos: imágenes, archivos de audio, archivos de video y archivos de texto. Estos métodos reciben el identificador del recurso y devuelven la URL del archivo adaptado junto con el tamaño de dicho archivo para que el usuario decida si quiere descargar el archivo o no.

Se ha diseñado un sistema de caché para los recursos adaptados para así evitar tener que repetir el proceso de adaptación cada vez que un recurso es accedido. Para poder manejar todas las instancias de recurso creadas se utiliza un fichero XML. Cada vez que se crea una nueva instancia de recurso, se añade un nuevo elemento al árbol XML. Este elemento contiene información sobre la ruta del archivo, su formato y sus características específicas (ancho y alto en píxeles, número de colores, codecs de audio y video usados, etc) que dependerán del tipo de recurso. De este modo, cada vez que el servidor recibe una petición de acceso a un recurso, comprueba si alguna instancia adaptada ya ha sido creada. Después, comprueba si esa instancia es compatible con las características del dispositivo móvil. Sólo será devuelta si es compatible, si no, se llevará a cabo una nueva adaptación.

4.2. Adaptación de contenidos

En cuanto al proceso de adaptación de contenidos en sí, el primer paso para todos los tipos de adaptación es comprobar si el dispositivo del usuario soporta el tipo de archivo de recurso original. Si no lo soporta, se generará, si es posible, una nueva copia en otro tipo de archivo que sí sea admitido por el dispositivo.

En el método que devuelve imágenes, la primera transformación que se lleva a cabo es escalar las dimensiones de la imagen para ajustarla al tamaño de la pantalla del terminal. Después, si es necesario, se reduce el número de colores de la imagen, pudiendo ser convertida a escala de grises si la pantalla del móvil no fuera en color. Finalmente, la imagen obtenida se guarda en un archivo que será enviado a la aplicación cliente.

En el caso de los archivos de audio (MP3, WAV, etc), el objetivo es reducir moderadamente la calidad del archivo para obtener un archivo de menor tamaño. Las transformaciones que se llevan a cabo son convertir la pista de audio de estéreo a mono y reducir la tasa de muestreo. La URL del nuevo archivo creado es devuelta al usuario.

En cuanto a los videos, la adaptación del audio es similar al caso de los archivos de audio y la imagen del video se escala para ajustarse a las dimensiones del dispositivo. El audio se codifica en PCM y el video en H.263, aceptado por

muchos terminales. Los formatos contenedores de vídeo que podrán ser adaptados son AVI (*Audio Video Interleave*) y MOV (*QuickTime*).

Por su parte, la adaptación de los documentos PDF o HTML se basa en la extracción de la información textual de los mismos y su transmisión como un archivo de texto plano.

4.3. Cliente móvil

Se está desarrollando una aplicación en J2ME, que actuará como cliente del servicio web, siguiendo la especificación JSR-172 (*J2ME Web Services*), empleando el perfil MIDP 2.0 junto con la configuración CLDC 1.1.

Las invocaciones de los métodos remotos del servicio web se realizan en hilos de ejecución independientes para evitar que un fallo en la conexión bloquee la aplicación. Además, dado que actualmente las conexiones existentes suelen ser lentas, se muestra una pantalla de espera informando de que la conexión con el servidor está teniendo lugar.

La interfaz gráfica que se está desarrollando permitirá al usuario navegar por la estructura de los cursos de una manera sencilla. Se han implementado un reproductor multimedia para reproducir los archivos de audio y de vídeo, un visor de imágenes y un visor de texto (figura 4).



Figura 4. Dos capturas de pantalla del cliente móvil

Se ha diseñado un sistema de caché de consultas sobre la memoria RMS (*Record Management System*) del móvil para permitir navegar offline por la estructura de los cursos y los recursos. También se dará la posibilidad de almacenar en la tarjeta de memoria del móvil los recursos descargados, si se dispone de ella y hay suficiente espacio libre, para así evitar tener que descargarlos cada vez que se quiera consultarlos.

5. Conclusión

Durante el desarrollo del proyecto de m-learning en ClayNet se ha tenido que tomar varias decisiones críticas relativas al propio desarrollo como, por ejemplo, emplear el API JSR-172 (J2ME Web Services) para cliente móvil, lo cual limita la compatibilidad de la aplicación a los terminales más modernos. Sin embargo, esto ha simplificado mucho el diseño y además, como los usuarios de telefonía móvil renuevan su terminal con bastante frecuencia, no supone un gran inconveniente. También se ha decidido emplear bibliotecas de software libre para realizar la adaptación de los distintos tipos de archivo, lo cual reduce un poco el abanico de formatos de archivos que se podrán adaptar, si bien los más usuales sí que son soportados.

Se está haciendo un gran esfuerzo para conseguir que la aplicación tenga un alto nivel de usabilidad, minimizando los tiempos de conexión y proporcionando una interfaz gráfica atractiva y, al mismo tiempo, sencilla de utilizar.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia y por el proyecto FEDER Keops (TSI2005-00960). Asimismo, queremos agradecer al Grupo de Investigación en Interacción y eLearning (GRIAL) de la Universidad de Salamanca por sus ideas y contribuciones para el desarrollo de este trabajo.

Referencias

- [1] Abdelnur, A. and Hepper, S.: The Java™ Portlet Specification Version 1.0. Sun Microsystems, Inc. (2003) <http://www.jcp.org/en/jsr/detail?id=168>
- [2] García, F. J. et al.: Portlet-based architecture for a LMS: CLAYNET 2.0. Virtual Campus 2006 Post-proceedings. Selected and Extended Papers, 181-194.
- [3] Hellers, N.: Aprendizaje portátil, la revolución que se viene. e-learning América Latina. (2004) http://www.elearningamericalatina.com/edicion/junio1_2004/na_1.php