

# ClayNet evolution towards mLearning and Information Visualization Technologies

Miguel Ángel Conde<sup>1</sup>, Carlos Muñoz<sup>2</sup>, Jorge Reyero<sup>2</sup>

<sup>1</sup> Dpto. Informática y Automática – Universidad de Salamanca  
“Campus Viriato”. Avda Cardenal Cisneros, 34, 49022, Zamora, España  
[mconde@usal.es](mailto:mconde@usal.es)

<sup>2</sup> Departamento de I+D+i CLAY Formación Internacional,  
C/Hoces del Duratón nº 57, 37008, Salamanca, España  
{carlos, j.reyero}@clayformacion.com  
<http://www.clayformacion.com/>

**Abstract.** ClayNet comes up as an eLearning platform based on portlets, innovative technologie which provides users functionalities of a Learning Management System (LMS). In the course of time and faced with the evolution of technolgies, it is necessary to widen those functionalities both as in content adaptation context to mobile devices as in what refers to information visualization. With these actualizations what is expected is improving students use of the couse..

## 1. Introducción

El *eLearning* es una variedad del proceso de aprendizaje con una aceptación cada vez mayor en todos los ámbitos. El concepto de *eLearning* se define de muchas formas diferentes, fundamentalmente debido a que los actores que de él hacen uso son muy diversos, cada uno con su idiosincrasia y su ámbito de aplicación. Desde la perspectiva que ofrece la experiencia en el desarrollo y explotación de plataformas *eLearning*, se podría definir el *eLearning* como “la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada estudiante, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias” [4]. El uso del *eLearning* requiere de la adaptación tecnológica al mismo, con tal cometido surgen las plataformas educativas. Una plataforma educativa es un espacio al cual se ha adaptado una aplicación TIC (Tecnologías de la Información y la Comunicación) que vincula los procesos de enseñanza-aprendizaje a un modelo pedagógico a través de un entorno virtual. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea o LMS (*Learning Management Systems*) y, por otro lado, las que se utilizan para la gestión de los contenidos digitales o LCMS (*Learning Content Management Systems*).

## 2 Miguel Ángel Conde<sup>1</sup>, Carlos Muñoz<sup>2</sup>, Jorge Reyero<sup>2</sup>

ClayNet 1.0 surge tratando de proporcionar la base tecnológica mencionada, que aporte a los usuarios ciertos servicios, a partir de los cuales pueda explotarse toda la potencialidad de las diferentes variedades de *eLearning*. Esta primera versión fue una aplicación web dependiente de la tecnología y carente de la portabilidad deseable para un entorno de estas características. Además, ClayNet 1.0 se orienta hacia el curso, proporciona servicios para el desarrollo, gestión y mantenimiento de los mismos, mientras que las tendencias actuales se aproximan más a un modelo de plataforma adaptado al alumno de forma individualizada.

Teniendo en cuenta estas consideraciones aparece la versión 2.0 de ClayNet, con la que se pretende establecer un punto de referencia en el campo de la enseñanza, cubriendo el ámbito desde un apoyo a la educación presencial hasta el desarrollo del aprendizaje totalmente en línea. Claynet 2.0 se ha desarrollado basándose en la tecnología *portlets* [1] e integrada en un portal.

En esta última versión de la plataforma ClayNet se va a buscar posibilitar a los alumnos la consumición de los recursos dispuestos en los cursos mediante el uso de dispositivos móviles. Con este nuevo servicio se trata de aportar un valor añadido a la plataforma permitiendo una realización real de los cursos en cualquier momento, lugar y situación y suponiendo un paso más en la evolución del proceso de aprendizaje hacia el *mLearning*.

Además de estas nuevas posibilidades también se van a aplicar nuevas herramientas a ClayNet de cara a potenciar la visualización de la información relativa a los diferentes elementos que se desean presentar a través de la misma. En concreto se aportan técnicas relativas a disciplinas como la visualización de la información para poder optimizar la forma en que se presentan los recursos y su estructura mediante el editor de cursos.

A continuación, en el segundo punto, se pasa a definir la plataforma ClayNet, base de los desarrollos a realizar, en el punto 3 se hablará de ClayNet y la concepción de *mLearning*, en el punto 4 de las técnicas de visualización de la información y se terminará en el punto 5 con una serie de conclusiones.

## 2. Plataforma ClayNet

ClayNet 2.0 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. Dicha plataforma 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 que pueden agruparse e interaccionar para formar un portal. Estos *portlets* constituyen aplicaciones web independientes. Entre las ventajas del desarrollo basado en *portlets* se pueden destacar:

- Desarrollo independiente. La independencia de los *portlets* permite desarrollarlos de forma separada, y en paralelo con otros desarrollos. Esta característica acelera el tiempo de implementación.
- Personalización. El aspecto final del portal dependerá de los *portlets* que contenga y éstos pueden ordenarse y cambiar esta ordenación de una manera sencilla.

- Seguridad frente a fallos. Un fallo en un *portlet* no se transmite al resto de la aplicación por lo que el único componente que deja de funcionar es el *portlet* que dio el fallo.

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 *eLearning*, 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 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>).

Nombre	Fecha de Inicio	Resumen	Precio
Perfiles profesionales em	01-08-2006	Una apuesta por la calidad en	65

Estadísticas				
Nuevos debates: 0	Nuevos mensajes: 0	Foros: 0	Debates: 0	Mensajes: 0

Fig. 2. Arquitectura de ClayNet con la adaptación móvil.

### 3. Claynet y mLearning

El objetivo principal de la aplicación del *mLearning* 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.

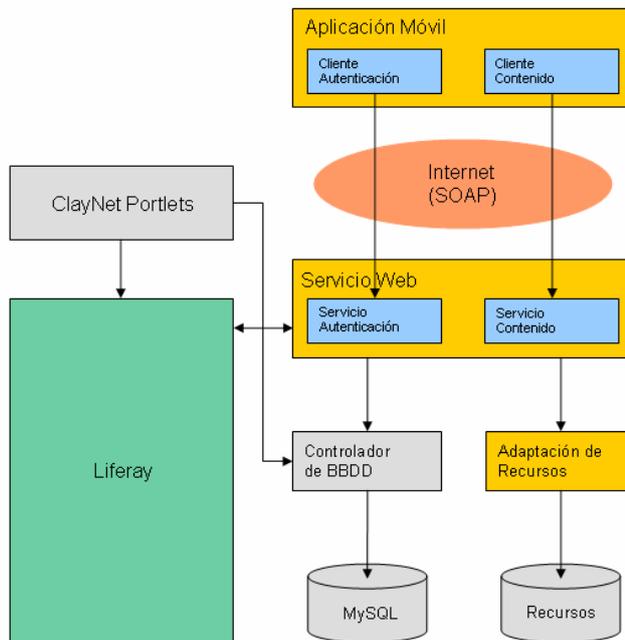
### 3.1. Sistema de movilidad de ClayNet

Se ha desarrollado 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. Puede observarse la integración de esos componentes sobre ClayNet en la Figura 2.

Para solventar los posibles problemas de conectividad de los móviles se posibilita la descarga de contenidos y el trabajo local sobre los mismos, permitiendo posteriormente una sincronización de los cambios



**Fig. 2.** Arquitectura de ClayNet con la adaptación móvil.

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 de acciones para llevar a cabo la adaptación, que dependerá del tipo de recurso.

### 3.2. Elementos del sistema de movilidad

La implantación de soluciones móviles en ClayNet requiere de varias partes como se ha observado en la Figura 2, a continuación se comentan con mayor detalle:

- **Servicio web.** En la parte correspondiente al servicio web se debe proporcionar dos elementos principales para realizar las acciones de autenticación y poder llevar a cabo peticiones de contenido, estos dos elementos serán:
  - **Servicio Autenticación y mantenimiento de sesión.** Se encarga de generar un objeto con las características del terminal en cuestión a partir del usuario y la contraseña. Una vez generado se devuelve al cliente y será reutilizado en cada transacción para su identificación.
  - **Servicio Contenido.** Proporciona 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. Hay distintos métodos para solicitar recursos dependiendo en el tipo de recurso deseado. 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, donde la información será almacenada en un fichero XML. Dependiendo del tipo de recurso instanciado, y cada vez que se realice esta acción se almacenará la información correspondiente para posibles solicitudes similares posteriores.
- **Adaptación de contenidos.** La adaptación de contenidos requiere en primer lugar de una comprobación de soporte de los mismos desde los dispositivos, de forma que si no se soporta se trataría de adaptar a un formato adecuado. Las transformaciones a realizar dependerán del tipo de recurso. En cualquiera de los casos se trataría de, mediante el uso de diferentes técnicas (escalado, bibliotecas de transformación, etc), modificar los recursos de modo que fueran aceptados por el terminal y supusieran la menor transferencia posible de datos.
- **Cliente móvil.** Se ha desarrollado 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. La interfaz gráfica que se ha definido permite 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. Se ha diseñado además 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 realizar una consulta.

## 4. ClayNet y la visualización de la información

Hoy en día las aplicaciones web tratan de buscar unas interfaces lo más atractivas posibles para los usuarios, pero que a su vez aporten toda la información necesaria. Estos dos conceptos no son incompatibles, actualmente existen gran cantidad de técnicas de visualización alguna de las cuales se aplican en los diferentes *portlets* que se incluyen en ClayNet. En concreto la aplicación estas técnicas es uno de los elementos que persiguen hacer evolucionar ClayNet hacia una plataforma más completa y atractiva.

### 4.1 El editor de cursos

Uno de los elementos concretos en los que se utilizan técnicas de visualización es el editor de cursos. Este *portlet* surge ante la necesidad de ofrecer una representación flexible e intuitiva de los elementos que pueden componer el curso, como son los recursos y las actividades. El editor del que se disponía hasta entonces simplemente permitía navegar por niveles, refrescando toda la pantalla de una forma no muy usable. Se busca la integración de esas funcionalidades, así como la adición de otras, a través de un único *portlet*, que no sufriera refrescos significativos, para lo que se pensó en el uso de la tecnología *ajax* (*Asynchronous JavaScript And XML*). El objetivo del editor será por tanto aportar elementos de forma más usable, accesible y atractiva de cara al usuario.

En primer lugar se desarrolla un *portlet* que permitiera la representación de la jerarquía de elementos involucrados en los cursos y utilizara una representación en forma de árbol, usando la tecnología *ajax* para potenciar la usabilidad del elemento (Figura 3).

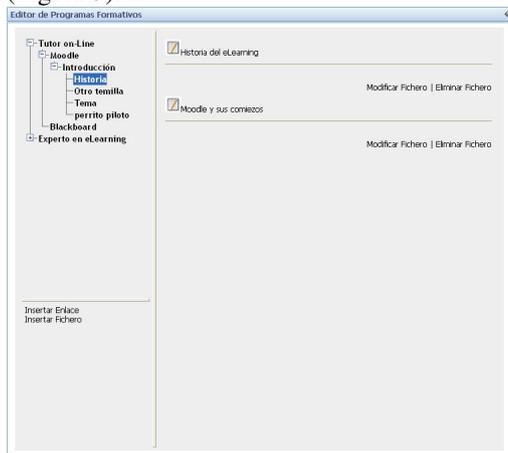


Fig. 3. Editor de cursos inicial

El mayor inconveniente de esa representación arbórea es que el espacio en el que se representa es pequeño, y si el número de elementos crece aparecerán dos *scroll* que

obligan al usuario a desplazarse por la lista horizontal y verticalmente pudiendo suponer una pérdida de contexto. Para solventar esta situación se analizaron diferentes posibilidades como se observa en el siguiente apartado.

#### 4.2. Evaluación de las opciones de visualización

Considerando el pequeño espacio del que se dispone para representar la información, y teniendo en cuenta que no se quiere abrir un nuevo elemento con la representación arbórea (ventana con árbol), se pueden descartar ciertas opciones como los árboles hiperbólicos y cónicos [6]. Este tipo de árboles necesitaría un espacio adicional para la representación. Los *Tree Browser* se descartan también puesto que se necesitaría una parte donde mostrar el fragmento del árbol maximizado, y en la parte central se desean mostrar los elementos. Otro tipo de representaciones también estudiadas suponían superponerse a la parte central de algún modo y también fueron descartadas (representaciones tipo Desafío). Ante estas limitaciones se decide usar el *Treemap*.

Los *Treemaps* son un tipo de visualización ideada a principios de los 90 por Ben Shneiderman, que permite representar jerarquías de forma que se optimiza el llenado del espacio y, además, ver los atributos de la misma e identificar patrones anómalos o propiedades de la jerarquía usando colores.

Las jerarquías se pueden representar de diversas maneras, por ejemplo mediante diagramas de Venn o mediante estructuras en árbol (por ejemplo como lo hace el explorador de Windows), que ocupan mucho espacio y no indican de forma compacta y efectiva las propiedades de las "ramas" y "hojas" del árbol.

Así pues, Shneiderman [5] decidió convertir el clásico árbol en una representación plana llenando todo el espacio disponible en el que cada fichero se convierte en un rectángulo de área proporcional a su tamaño. Los rectángulos se anidan unos dentro de otros en función del nivel al que se desciende en la jerarquía alternando la extensión vertical con la horizontal.

En realidad el área de un *Treemap* se puede hacer proporcional a la magnitud que sea de nuestro interés, lo que le da parte de su versatilidad. Por otro lado, los nodos y "hojas" del árbol son susceptibles de colorearse, asignando a cada color un significado concreto. [3]

En cuanto a las tecnologías para la implementación del *Treemap* existen múltiples posibilidades, considerando siempre las restricciones como que la tecnología debe generar un *Treemap* html y que este puede actualizarse con relativa frecuencia.

Entre la multitud de implementaciones de *Treemaps* disponibles se opta por Roomformilk (<http://www.roomformilk.com>). Este *Treemap* se utiliza como lector de noticias de comunidades *slashdot*. Es una representación que muestra titulares de noticias cuyo color depende de la antigüedad de las mismas. El aspecto es similar al que se muestra en la Figura 5.

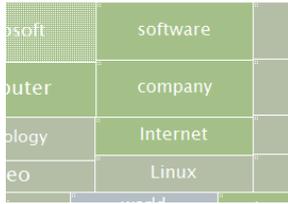


Fig. 5. Imagen del Roomformilk

Esta implementación del *Treemap* está basada en la tecnología php combinada con el uso de hojas de estilo. En Internet se puede obtener una versión de los archivos fuentes descargable que ha permitido al desarrollador adaptarla a Java, jsp y *ajax*.

#### 4.2. Implementación de Treemap sobre el editor de cursos

En el caso de ClayNet se aplica sobre el editor de cursos en sustitución de un árbol jerárquico que podría extenderse tanto vertical como horizontalmente. El *Treemap* permite observar los diferentes elementos de cada nivel de un vistazo y navegar entre los diferentes niveles existentes.

La implementación de la funcionalidad requerida para el *Treemap* supone la definición de una serie de métodos que se encargarán, desde generar el html para el pintado, a otros para la variación de los contenidos en función de los niveles, considerando que para cualquiera de las acciones de refresco se utilizará *ajax*. A continuación se describen brevemente alguno de los principales métodos:

- **renderTreemap.** Se encargará de la generación de la estructura html del *Treemap* a incluir y deberá ser invocado en cada ocasión que se requiera repintar este. Tendrá que recibir ciertos parámetros:
  - **Información de los elementos a dibujar.** Que permita determinar los nombres de los ítems y sus características asociadas.
  - **Fechas.** Para poder establecer el color del rectángulo correspondiente al elemento en función de la diferencia entre la fecha de creación y la fecha actual.
  - **Tipo.** Para definir en qué nivel se encuentran cada uno de los elementos y establecer así un valor de retorno al elemento anterior.
  - **Anchura, Altura y Profundidad.** Que permitan determinar los parámetros del *Treemap* de cara a posibles adaptaciones posteriores.
  - **Orientación.** De disposición de cada uno de los elementos.
  - **Dirección.** Dirección a la que apuntará cada elemento del *Treemap* dentro de la jerarquía de elementos del curso.
- **TreemapElementoX.** Método que se encargará de recuperar los datos de forma adecuada para cada elemento de la jerarquía del curso, dónde ElementoX podría referirse a Títulos, Módulos, Unidades o Temas, y que posteriormente utilizará *renderTreemap* para generar el HTML correspondiente a cada vista. Para hacer un refresco ágil se utilizaría la tecnología *ajax* como ya se ha mencionado.
- **Otras funciones de utilidad.** Para facilitar el uso del método de *render* se crean varios métodos que se invocarán en función del nivel del curso a dibujar. Éstos

ahorran la repetición innecesaria de código, puesto que debe considerarse que por cada nivel se podrán listar los elementos en varias ocasiones.

Una vez implantado el *Treemap* la plataforma gana en visibilidad, flexibilidad y permitiría mantener el contexto, aportando además un dinamismo adicional al editor de cursos, que quedaría de cómo se observa en la Figura 6.



Fig. 6. *Treemap* a nivel de títulos

## 5. Conclusiones

Con las evoluciones de las diferentes tecnologías se busca aportar mejoras que puedan beneficiar al usuario final de las aplicaciones. En el caso de ClayNet no podía estancarse su desarrollo sin aplicar las nuevas tecnologías, que posibilitarán que el usuario de la plataforma pueda utilizarla de forma mucho más eficaz.

Una de las líneas que se consideran prioritarias en el *eLearning* actual es la de la movilidad. En la sociedad actual cada vez es mayor el uso que se hace de los dispositivos móviles y las plataformas educativas deben adaptarse a este uso y a los nuevos modelos formativos asociados. El hecho de que se plantee otro tipo de formación supone que las tecnologías actuales posibiliten ese tipo de adaptación tanto a los docentes como a los alumnos consumidores de los cursos. ClayNet, con su adaptación de contenidos a dispositivos, posibilita que los usuarios puedan consumir recursos de la plataforma en cualquier momento y lugar, lo que supone una ventaja competitiva muy importante respecto a otras plataformas.

Debe comentarse también que los *portlets* existentes deben mejorarse para facilitar el uso de la plataforma a los usuarios de ahí que se apliquen nuevas técnicas de visualización que permitan obtener una mayor cantidad de información y mejorar en alto grado la usabilidad de la plataforma. Este última característica supone un

aumento cualitativo de la calidad del servicio ofrecido en busca de un mayor aprovechamiento de los mismos.

Para concluir debe comentarse que la plataforma ClayNet debe permanecer abierta a los nuevos cambios tecnológicos, es por eso que se continua investigando en nuevas evoluciones que permitan poner a disposición de los alumnos una herramienta más potente cada vez.

## 6. Bibliografía.

- [1] Abdelnur, A. and Hepper, S.: The JavaTM Portlet Specification Version 1.0. Sun Microsystems, Inc. (2003) <http://www.jcp.org/en/jsr/detail?id=168>
- [2] Conde González, M. Á., Carabias González, J., Martín Moreno, R. M<sup>a</sup>, González Pérez, I., García Peñalvo, F. J. "Portlet-based ARCHITECTURE FOR A LMS: CLAYNET 2.0". En F. J. García Peñalvo, J. Lozano Galera, F. Lamamie de Clairac Palarea (Eds.). Virtual Campus 2006 Post-proceedings. Selected and Extended Papers. CEUR Workshop Proceedings. <http://ceur-ws.org/>. ISSN 1613-0073. En prensa.
- [3] Dürsteler, J. 2001. Treemaps. Revista Infovis. Mensaje n° 51. <http://www.infovis.net/printFicha.php?rec=revista&num=51&lang=1&palabra=treemap>
- [4] García Peñalvo, F. J. 2005. Estado actual de los sistemas e-learning, Teoría de la Educación: Educación y Cultura en la Sociedad de la Información, Volumen 6(2). [http://www.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://www.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm)
- [5] Johnson, B. and Shneiderman, B. 1991. Treemaps: A Space-filling Approach to the Visualization of Hierarchical Information. In Proceedings of IEEE Visualization 91 Conference, pages 284-291.
- [6] Pirolli, P., Card, S. and Van Der Wege, M. 2003. The effects of information scent on visual search in the hyperbolic tree browser, ACM Transactions on Computer-Human Interaction (TOCHI), v.10 n.1, p.20-53, March 2003