

## Portlet-based architecture for a LMS: CLAYNET 2.0

Miguel Ángel Conde<sup>1</sup>, Jorge Carabias<sup>1</sup>,  
Rosa María Martín<sup>1</sup>, Inmaculada González<sup>1</sup> and  
Francisco José García<sup>2</sup>

<sup>1</sup>Departamento de I+D+i CLAY Formación Internacional,  
C/Hoces del Duratón nº 57, 37008, Salamanca, España  
{miguelangel, jorge, rosa, inma}@clayformacion.com

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

**Abstract:** Any system of e-learning requires the use of the technologies, that is to say, e-learning's platforms. ClayNet is a platform that contributes the desirable functionalities for the e-learning process, is integrated in a portal and constituted by assembled components with independent structure called portlets.

**Keywords:** Portlet, e-learning, forum, platform, ClayNet, portability.

### 1. Introducción.

El *e-learning*, que se define por tres elementos: la comunicación, el conocimiento y la tecnología, es una herramienta indispensable a la hora de establecer procesos de formación continua. Esta nueva forma de aprender establece una nueva relación alumno/profesor que hace desaparecer la jerarquía existente entre ambos. El *e-learning* consiste en la utilización y aprovechamiento de Internet para desarrollar proyectos formativos. Permite el acceso a una red de conocimiento dinámico que facilita a las personas un aprendizaje de una manera personalizada y flexible.

El uso del *e-learning* 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. Es decir, un sistema informático localizado en un sitio web que suele ser de acceso restringido con el fin de identificar el perfil del usuario. Este sistema informático habilita un espacio de trabajo compartido por alumnos y profesores en el que se intercambian documentos y actividades en el proceso enseñanza-aprendizaje a través de recursos, actividades de aprendizaje y herramientas de comunicación.

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 *e-learning*. Esta primera versión ha sido 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.

En este marco surge la plataforma de aprendizaje ClayNet 2.0, pretendiendo ser un punto de referencia en el campo de la enseñanza, cubriendo 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 2.0 se ha desarrollado basándose en la tecnología *portlets* (Abdelnur and Hepper, 1993) e integrada en un portal, que se define a continuación.

Un portal es un punto de entrada a Internet donde se organizan sus contenidos, ayudando al usuario y concentrando servicios y productos (Arnedo, 1999), de forma que le permitan a éste hacer cuanto necesite en Internet sin tener que salir de dicho sitio web. El objetivo pretende ser fidelizar a los usuarios para conseguir no sólo que éstos coloquen dicha página en su *bookmark*, sino que dicha página sea la página de inicio del navegador del usuario, con lo que se generará un tráfico en esa web que garantice la supervivencia en la Red.

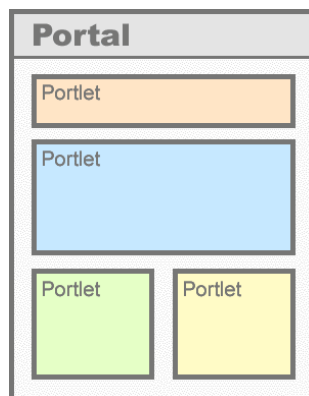
Hasta ahora un portal de carácter general ofrecía servicios de valor añadido como comunidades virtuales, espacio web gratuito, información de diverso tipo, personalización de la información, chat, *e-mail* gratuito, mensajes a teléfonos móviles, *software* gratuito, grupos de discusión, comercio electrónico, buscador, etc.

Sin embargo, este modelo empieza a quedarse obsoleto, pues a ese nivel tan amplio sólo pueden sobrevivir unos pocos y empieza a haber demasiados portales generalistas, orientados a usuarios principiantes y basados en servicios poco especializados, inadecuados para los usuarios más expertos, lo que está haciendo necesario redimensionar este mercado a través de la especialización geográfica, temática o corporativa.

ClayNet 2.0 es una plataforma que se encuentra integrada en un portal pero es independiente del mismo, esto supone que pueda funcionar de forma aislada e integrada sobre otras plataformas.

La plataforma que se está desarrollando esta compuesta por *portlets*. Son componentes web gestionados por un contenedor que, tras la petición de un usuario, generan y presentan contenidos dinámicos de forma identificable en la interfaz de usuario del portal como componentes de contenido. El *portlet* permite la personalización, la presentación, y la gestión de la seguridad.

El contenido generado por los *portlets* se denomina “fragmento”. Es código XHTML (*Extensible HyperText Markup Language*), HTML (*HyperText Markup Language*), WML (*Wireless Markup Language*), etc. Los fragmentos agregados resultantes de la operación de varios *portlets* constituyen un documento que se traduce en la interfaz del portal. Estos elementos se disponen a través de una “retícula” o rejilla como los que se observan en la Figura 1.



**Figura 1.** Retícula contenedora.

Los *portlets* que componen ClayNet 2.0 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.

A continuación se muestra la arquitectura utilizada, el desarrollo de uno de los módulos y una serie de conclusiones obtenidas.

## 2. Arquitectura de ClayNet 2.0.

ClayNet 2.0 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 interactuar 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.

En la Figura 2 se ilustra la plataforma ClayNet. Se observa el aspecto modular que el uso de *portlets* otorga a la plataforma. Cada componente se gestiona de manera independiente pudiendo ser maximizado o minimizado. También se permite añadir o quitar componentes de tal manera que el aspecto pueda ser alterado rápidamente. ClayNet mantiene *portlets* para elementos como la administración de la plataforma, visualización de contenidos, foros, etc.

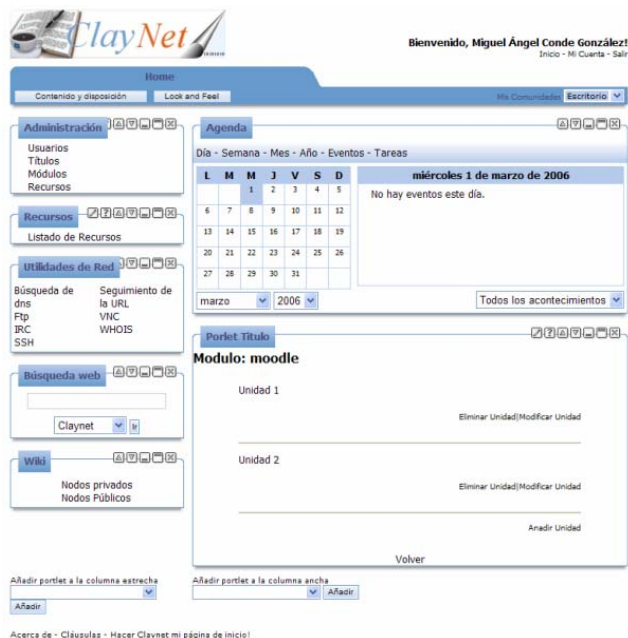
Como diferencias entre las plataformas de *e-learning* que se pueden encontrar en el mercado y una plataforma basada en *portlets* como ClayNet se pueden destacar:

- Personalización de los módulos para cada usuario.
- Robustez debida a, como se ha citado anteriormente, el aislamiento de los *portlets*.
- Dinamicidad y sencillez en el desarrollo de nuevos componentes.
- Posibilidad de realizar gestiones independientes como, por ejemplo, de seguridad.

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 de éstos, la JSR 168 (Abdelnur and Hepper, 1993), se ha elegido Liferay (<http://www.liferay.com>). Liferay es una herramienta de código abierto que permite la construcción de portales.

Entre las características destacables de Liferay se pueden citar:

- Soporte para la especificación JSR 168. Esta API es la utilizada para la construcción de los componentes de ClayNet.
- Soporte para WSRP (*Web Services for Remote Portal*) (Kropp et al., 2003), es un estándar de servicios web que permite publicar *portlets* en plataformas remotas y viceversa. Esta característica permite comprar y vender la funcionalidad de un *portlet* como servicio.
- Soporta el uso de temas para cambiar el aspecto.
- Funciona con multitud de servidores de aplicaciones como Jetty y Tomcat, o en servidores J2EE (*Java 2 Platform, Enterprise Edition*) como Borland ES, JBoss+Jetty/Tomcat, JOnAS+Jetty/Tomcat, JRun, OracleAS, Orion, Pramati, RexIP, Sun JSAS, WebLogic, y WebSphere.
- Multiplataforma. El hecho de que esté programado en el lenguaje Java confiere a Liferay la fácil portabilidad a sistemas como BSD (FreeBSD, NetBSD, OpenBSD, etc.), Linux (Fedora, Novell, Gentoo, etc.), Solaris, Mac OS X y MS Windows.
- Independencia de base de datos. Liferay usa Hibernate (<http://www.hibernate.org/>) como herramienta ORM (*Object Relational Mapping*) para la capa de persistencia lo que permite enlazar diferentes bases de datos como DB2, Firebird, Hypersonic, InterBase, JDataStore, MySQL, Oracle, PostgreSQL, SAP o SQL Server.
- Internacionalización. Liferay permite mostrar y recibir información en múltiples idiomas.



**Figura 2.** Vista de ClayNet 2.0.

Una vez descrita la base sobre la que se ha desarrollado ClayNet se pasa a ver cómo se realiza la integración. En general, la plataforma ClayNet se basa en un conjunto de *portlets* que tienen distinta funcionalidad y que se comunican entre sí. La Figura 3 ilustra la arquitectura general de la plataforma.

En dicha figura, se observa cómo el conjunto de *portlets* desarrollados se incluyen en la cima de la arquitectura de Liferay (elementos se muestran en color azul y verde). Los *portlets* engloban una serie de clases que cumplen con la especificación JSR 168 y con un conjunto de elementos JSP (*Java Server Pages*) (<http://java.sun.com/products/jsp/>). Estos elementos son gestionados por el contenedor de *portlets*. 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/>) y para la comunicación entre las clases de los *portlets* y la base de datos se ha desarrollado una biblioteca de clases que actúa como envoltorio de la base de datos y abstrae al resto de la aplicación del manejo de la misma proporcionando una sencilla interfaz que automatiza el manejo lo máximo posible.

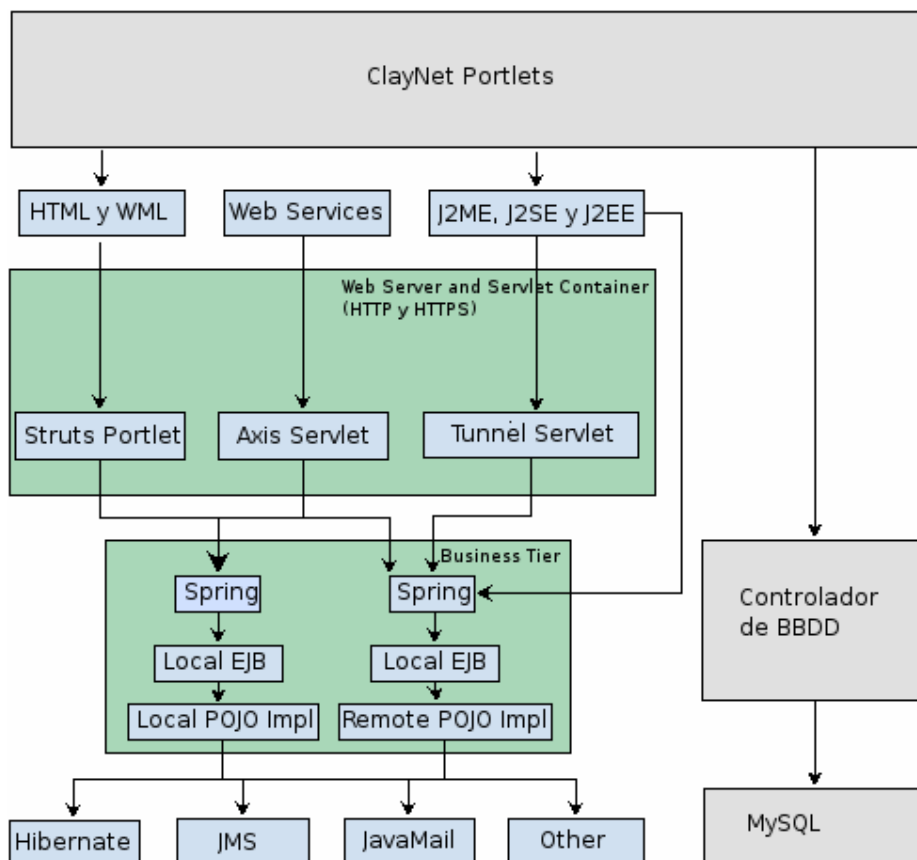


Figura. 3. Arquitectura de ClayNet 2.0.

### 3. Módulo de ejemplo: *Portlet ForoClayNet*.

Entre los numerosos *portlets* que integran ClayNet se encuentra el *portlet ForoClayNet*, desarrollado como componente de comunicación asíncrona disponible en todo momento para los usuarios de la plataforma.

En este apartado se presenta una breve descripción de este componente, pasando por la motivación que llevó a su desarrollo, los objetivos perseguidos y los aspectos más relevantes del diseño del *portlet*.

#### 3.1. Motivación.

Partiendo del hecho de que un foro es una de las herramientas de comunicación más útiles en los entornos de aprendizaje en línea, era evidente la necesidad de incluir en

la plataforma ClayNet un componente que proporcionara esta funcionalidad. Surgió así el propósito de desarrollo de un *portlet* que integrara todos los aspectos funcionales de un foro, esto es, que permitiera la comunicación asíncrona entre los usuarios en forma de hilos de debate.

### 3.2. Objetivos de *ForoClayNet*.

Desde el principio se pretendió desarrollar un foro de foros o “metaforo”, es decir, desarrollar un componente que gestionase todo el conjunto de foros creados en ClayNet y los debates iniciados en cada uno de ellos.

El *portlet* debería incluir todo tipo de funcionalidades comunes a los foros: crear nuevos foros y debates, abrir y cerrar debates, responder a los mensajes del foro, realizar búsquedas, marcar y calificar mensajes, etc.

Se debería permitir crear foros de diferentes tipos. El usuario podrá crear nuevos foros configurando sus características, permitiéndole salvar dicha configuración como “plantilla de foro”. Estas “plantillas” podrán ser utilizadas como base para la creación de nuevos foros. Se ha planteado que inicialmente se incluya un conjunto de plantillas de foro disponibles para el usuario. Entre ellas: plantilla para foros generales, plantilla para foros examen y plantilla para foros restringidos a un grupo de usuarios.

Así mismo, debería permitir la personalización de ciertos aspectos de interfaz del foro: vista de los debates, conjunto de etiquetas para mensajes, escalas de calificación, papelería de mensajes, colores, fuentes, ordenación de mensajes, etc.

Cada uno de los foros se enmarca en un contexto dentro de la plataforma ClayNet. Así, puede estar asociado a algún contenido de aprendizaje, o a la plataforma en sí. El acceso a un foro, por tanto, puede realizarse desde el *portlet ForoClayNet*, o desde el contenido de aprendizaje asociado. El contexto del foro determinará el conjunto de usuarios para los que es accesible.

Por último, señalar que el control de un usuario sobre un foro estará supeditado al rol de dicho usuario en la plataforma y los privilegios asignados por el administrador.

Se propuso, por tanto, la construcción de un *portlet* foro que abarcara el mayor número de posibilidades funcionales y de interfaz de usuario posibles, que sirviera como contenedor de todos los foros existentes en la plataforma, y que pusiera a disposición de los usuarios un conjunto inicial de plantillas de foro.

### 3.3. Desarrollo del *portlet ForoClayNet*.

El diseño del *portlet ForoClayNet*, así como del resto de componentes de la plataforma, se ha realizado siguiendo el patrón de diseño *Modelo Vista Controlador - MVC* (Buschmann et al., 1996). Con ello se consigue diferenciar y separar los elementos de información y funcionalidad propios del componente (modelo), de la vista de los datos que se presenta a los usuarios (vista).

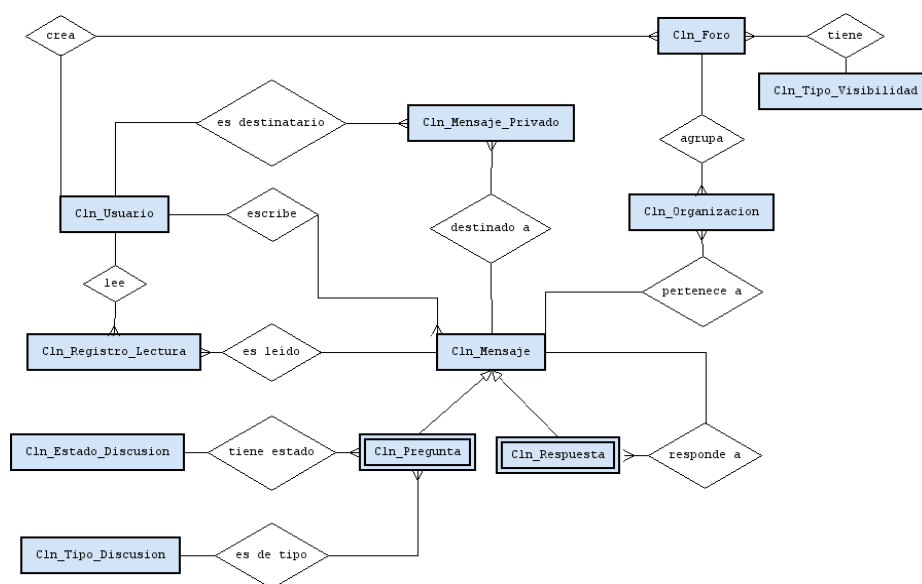
#### **Modelo de datos.**

Toda la información inherente al foro queda modelada en la base de datos, conformando el *Modelo* de la arquitectura.

Inicialmente el diagrama entidad-relación de la base de datos del foro tenía el aspecto de la Figura.4. Se puede observar cómo los usuarios de la plataforma pueden crear sus propios foros, y personalizar la organización de los debates.

Por otro lado, la mensajería privada, a diferencia de la pública, va dirigida a usuarios concretos.

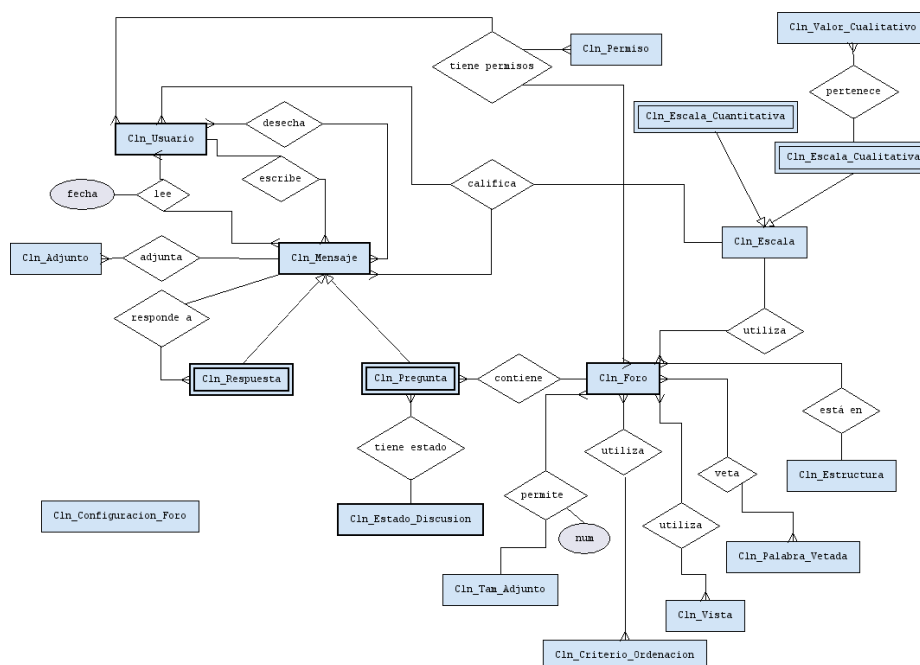
Señalar también que se mantiene información sobre los mensajes leídos por cada usuario, lo que permite notificar en todo momento de las novedades en los foros.



**Figura. 4.** Diagrama Entidad-Relación inicial de la base de datos.

El modelo de datos inicial ha evolucionado en el proceso de desarrollo software, incluyendo entidades y relaciones que permiten la consecución de los objetivos iniciales. En el momento actual, el diagrama entidad-relación tiene el aspecto de la Figura 5.





**Figura 5.** Diagrama Entidad-Relación actual de la base de datos.

Los aspectos más relevantes del modelo de datos actual son:

- No se mantiene información sobre el usuario que creó un foro. Ahora los foros son creados por un usuario con privilegios para ello, siendo irrelevante su identidad.
- Desaparece el concepto “tipo de debate”, lo que supone la eliminación de la entidad `Cln_Tipo_Discusion`, y su relación con `Cln_Pregunta`. Actualmente, los foros se crean con unas características concretas, y se puede salvar esta configuración de foro como “plantilla” para la creación de foros posteriores.
- Desaparece el concepto de “mensajería privada” tal y como se concebía en el modelo anterior. Actualmente, se puede conseguir un foro privado (accesible a un grupo concreto de usuarios) configurando sus características. Por tanto, desaparecen las entidades `Cln_Mensaje_Privado` y `Cln_Organizacion`, la cual servía para organizar los debates en el espacio privado de cada usuario.
- Se define en el sistema un conjunto de criterios de ordenación aplicables a los mensajes de los debates.
- Se define en el sistema un conjunto de vistas, esto es, de formas de presentación de los debates.
- Se mantiene una lista de tamaños máximos para los ficheros adjuntos.
- Se definen en el sistema un conjunto de permisos que determinan la forma en que los usuarios pueden acceder y participar en los foros. Desaparece por tanto la entidad `Cln_Tipo_Visibilidad` y surge la entidad

Cln\_Permission, que se asocia con Cln\_Foro y Cln\_Usuario en la relación ternaria “tiene permisos”.

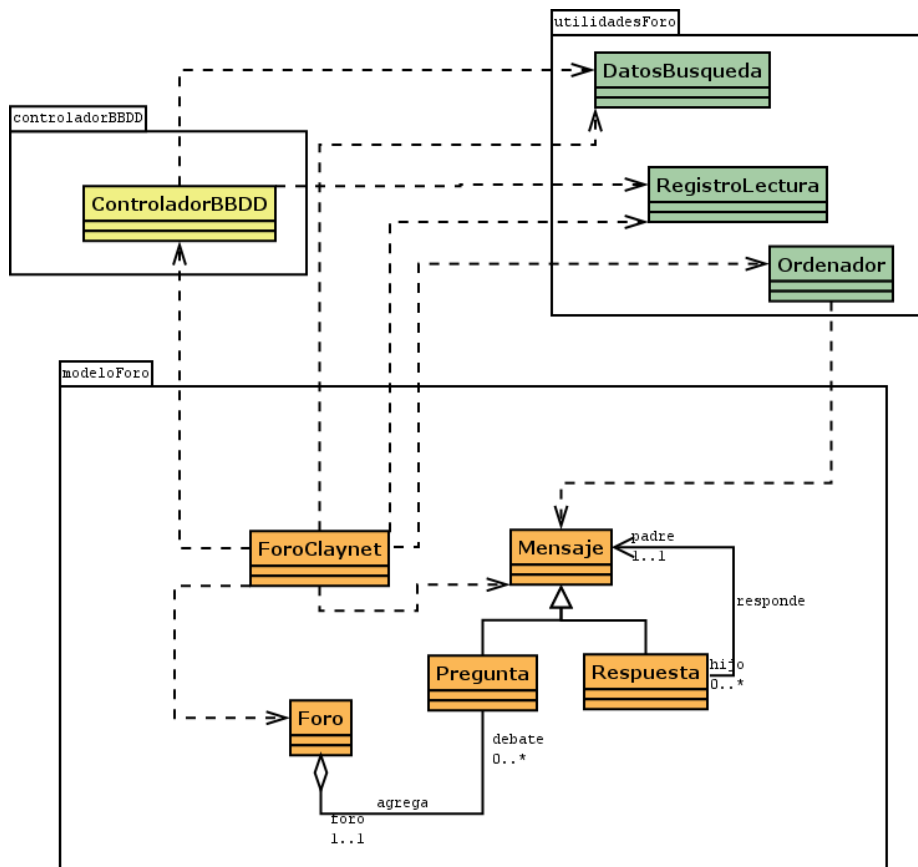
- La calificación de los mensajes de un foro se realiza en base a una de entre las varias escalas definidas en el sistema. Se contempla la existencia de escalas cualitativas y cuantitativas. Los usuarios pueden calificar los mensajes de un foro en base a la escala utilizada en éste. Esto se refleja en el modelo de datos con la relación ternaria “califica”.
- Se almacenan las palabras vetadas en los foros.
- El foro se crea asociado a un contexto, el cual es un contenido de aprendizaje creado en la plataforma ClayNet.
- Las características configurables del foro se reflejan en el modelo de datos con las asociaciones entre la entidad Cln\_Foro y las entidades Cln\_Estructura, Cln\_Vista, Cln\_Criterio\_Ordenacion, Cln\_Palabra\_Vetada, Cln\_Tam\_Adjunto, Cln\_Escala y Cln\_Permission.
- Se incluye el mantenimiento de los ficheros adjuntos de los mensajes, almacenando la información necesaria en la entidad Cln\_Adjunto.
- Se incluye el concepto “papelera” con la relación “desecha” entre Cln\_Usuario y Cln\_Mensaje. Con ello se permite a los usuarios desechar mensajes de los foros, los cuales no serán ya visibles para ellos.
- Se incluye la entidad Cln\_Configuración\_Foro, que mantendrá los atributos estáticos relacionados con la interfaz de usuario. De momento se contempla una única configuración, pero incluyendo esta entidad en el modelo, se permite la definición y mantenimiento de diferentes interfaces de usuario en el futuro.

#### **Controlador: Diagrama de clases del portlet ForoClayNet.**

Los datos persistentes del sistema son accedidos y manejados por el controlador, implementado por el conjunto de clases Java. Además de realizar las veces de controlador, estas clases implementan toda la funcionalidad del foro.

Se organizan en cuatro paquetes. El primero de ellos, modeloForo, contiene la información propia del contexto del problema, el paquete vistaForo, que contiene información de la presentación del foro, el paquete controladorForo, que interacciona directamente con la base de datos (BD), y el paquete utilidadesForo que contiene clases que van a utilizar el resto.

La Figura 6 recoge el diagrama inicial de clases del portlet ForoClayNet.



**Figura 6.** Diagrama de clases inicial del portlet *ForoClayNet*.

A lo largo del proceso de desarrollo software, el diagrama de clases inicial ha ido evolucionando hasta el momento actual, incluyendo las clases y relaciones que permiten la consecución de los objetivos iniciales. En el momento actual, el diagrama de clases tiene el aspecto de la Figura 7.

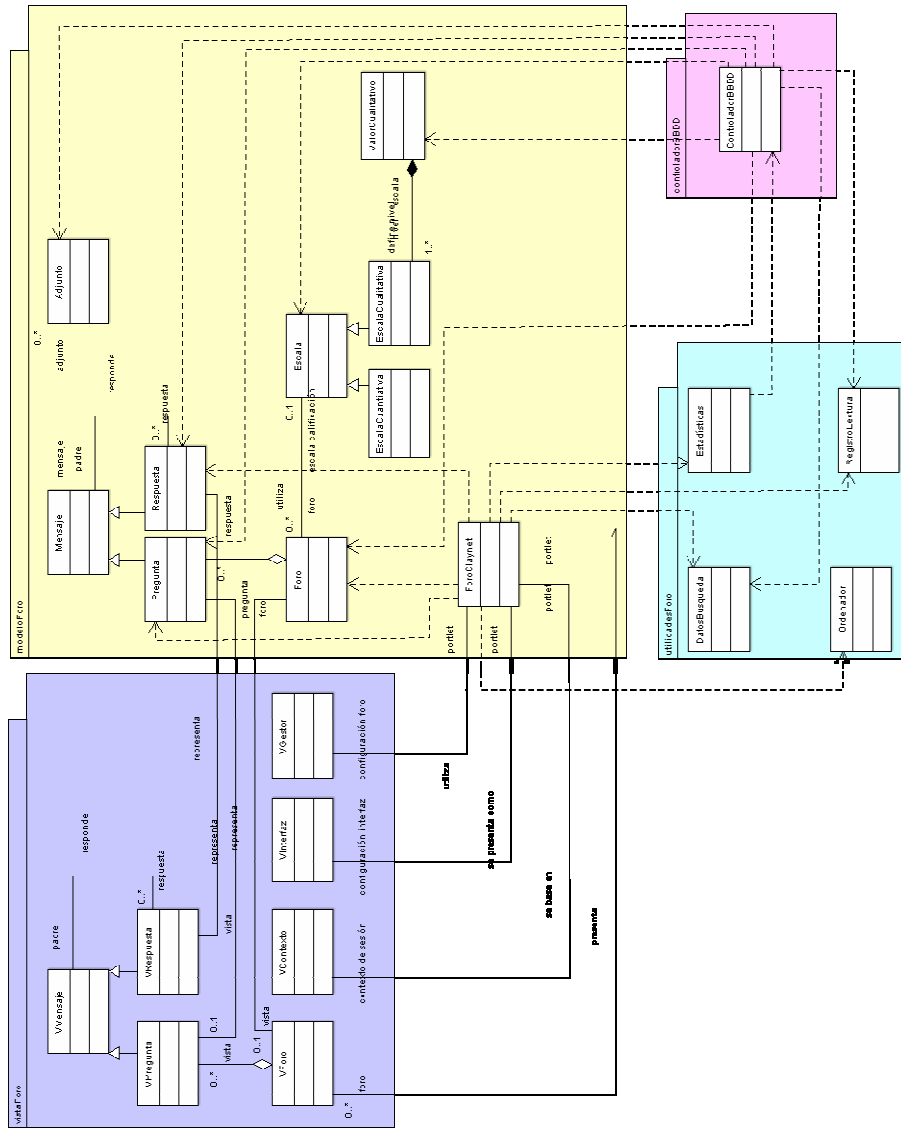


Figura 7. Diagrama de clases actual del portlet ForoClayNet.

Los aspectos más relevantes del nuevo diagrama son los siguientes:

- Se incluyen clases destinadas a la presentación de los datos al usuario. Estas clases se integran en el paquete *vistaForo*. De entre ellas, destacar la labor de *VContexto*, que mantiene los datos de la sesión de usuario en el foro. *VInterfaz* contiene los datos del aspecto de la interfaz de usuario, y *VGestor* mantiene los datos estáticos propios del *portlet* (tipos de permisos, tipos de ordenación, tipos de vista, etc.).
- Aparece la clase *Adjunto*, que modela un fichero adjunto a un mensaje
- Se crea la clase *Estadísticas*, que encapsula todos los datos sobre estadísticas de los foros para un determinado usuario.
- Se incluyen las clases *Escala*, *EscalaCuantitativa* y *EscalaCualitativa*, que forman una familia de escalas de calificación. La clase *ValorCualitativo*, por su parte, implementa un valor cualitativo integrado en alguna de las escalas cualitativas existentes.

#### **Vista.**

Por último, la Vista de la arquitectura incluye el conjunto de páginas JSP que dan forma a la interfaz de usuario del *portlet ForoClayNet*.

Se pretende que *ForoClayNet* ofrezca diferentes plantillas para el aspecto de la interfaz del foro, y que permita a los usuarios definir otras nuevas a partir de las existentes.

Desde el inicio del desarrollo se planteó como objetivo que la interfaz del foro permitiera en todo momento al usuario el acceso rápido a todas las partes funcionales del mismo (aquellas accesibles en base a los privilegios del rol de usuario). Se decidió entonces construir una interfaz basada en marcos, cada uno de los cuales se destinaría a contener los controles e información de un aspecto funcional del foro.

Por otro lado, dependiendo del rol del usuario conectado, y de su propia identidad, la interfaz de usuario tendrá ciertas particularidades (controles disponibles o no, información sobre novedades, foros y debates accesibles, etiquetas disponibles, etc.).

## **4. Conclusiones.**

Como conclusiones debe mencionarse que se ha conseguido construir una versión de plataforma educativa, que va a permitir explotar todas las posibilidades de este tipo de aprendizaje. La plataforma construida se encuentra integrada en un portal y está basada en *portlets*, como el que se ha descrito en el ejemplo. Esto va a aportar ciertas ventajas, como la posibilidad de desarrollo independiente y modularizado, la personalización de los módulos y la seguridad de la plataforma frente a fallos. Dichas ventajas aportan a ClayNet características como portabilidad, escalabilidad, robustez, fiabilidad, etc.

Es posible comparar ClayNet con algunas otras plataformas educativas que hacen uso de la tecnología de *portlets* como EducaMadrid (<http://www.educa.madrid.org/portal/web/educamadrid>). Debe considerarse que no se ha tenido acceso al código de la misma, pero sí al *feedback* de sus usuarios, lo que

permite aportar mejoras funcionales y tecnológicas a problemas ya planteados en ClayNet.

Lo que se pretende con ClayNet es solventar los problemas de otras plataformas de *e-learning* presentes en el mercado aportando nuevas soluciones y un nuevo enfoque a la tecnología educativa actual. Para conseguir este objetivo se persigue soportar los diferentes estándares educativos, adaptar ClayNet a la evolución de las tecnologías y del *e-learning* y usar la experiencia de los usuarios.

En el futuro se considera la posibilidad de dar soporte a un campus virtual a través de la plataforma. Dicho campus permitiría a los posibles alumnos desarrollar diferentes estudios que podrían ser certificados según las nuevas leyes europeas. Las calificaciones, destrezas y aptitudes alcanzadas por los alumnos se almacenarían en sus *portfolios* particulares y a partir de estos elementos podrían proporcionarse curriculums certificados a empresas.

## 5. Agradecimientos.

Este trabajo está parcialmente subvencionado por el Ministerio de Educación y Ciencia y por fondos FEDER mediante el proyecto KEOPS (TSI2005-00960).

## 6. Referencias.

- Abdelnur, A. and Hepper, S., 1993. Java™ Portlet Specification Version 1.0. Sun Microsystems, Inc. <http://www.jcp.org/en/jsr/detail?id=168>.
- Arnedo, T., 1999 De Portales a Plazas; Presente y Futuro de los Portales en Internet. *Internet '99*. Asociación de Usuarios de Internet. <http://www.aui.es/biblio/libros/mi99/3portales.htm>.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M., 1996. *Pattern Oriented Software Architecture: A System of Patterns*. John Wiley & Sons.
- Kropp, A., Leue, C., Thompson, R. (Eds.), 2003. Web Services for Remote Portlets Specification. OASIS Standard. <http://www.oasis-open.org/committees/download.php/3343>.