

GODACHEANT: E-LEARNING

Desarrollo de un editor online de objetos
de aprendizaje

Autores:

Alberto Álvarez Toledano
David Andrés Ferreras
Gonzalo Sánchez de Ocaña Rengifo

Profesor director:

Antonio Sarasa Cabezuelo

Proyecto de Sistemas Informáticos
Facultad de Informática
Universidad Complutense de Madrid
Curso 2007/08

Índice

RESÚMEN DEL PROYECTO	7
PALABRAS CLASE.....	8
INTRODUCCIÓN A LA APLICACIÓN GODACHEANT	9
Scorm.....	9
Estándares que conforman SCORM.....	9
Componentes de la especificación SCORM.....	9
Aplicación Godacheant.....	12
CASOS DE USO	15
Iniciar sesion.....	15
Darse de alta	16
Crear un OA en el editor	17
Abrir un OA en el editor.....	18
Guardar el OA actual en el editor.....	19
Crear un content preview.....	20
Importar un OA al editor	21
Exportar un OA	22
Salir de la aplicación	23
Publicar un OA	24
Búsqueda general de un OA	25
Búsqueda avanzada de un OA	26
Descargar un OA público	27
Añadir un OA público a la carpeta personal.....	28
Guardar un OA en la carpeta personal.....	29
Eliminar un OA de la carpeta personal.....	30
Importar un OA al editor desde la carpeta personal	31
Guardar un content preview en la base de datos	32
Eliminar un content preview de la base de datos	33
Descargar un content preview de la base de datos	34
Etiquetar un OA.....	35
Modificar las etiquetas de un OA.....	36
Publicar por Internet un OA	37
Generar nube de etiquetas	38
Refinar la búsqueda	39
Reiniciar la búsqueda.....	40
Retroceder en la búsqueda.....	41
Abrir un OA público.....	42
Etiquetar un OA público.....	43
DISEÑO	45
Cliente.....	45
Diagrama general de clases del cliente.....	46
Diagrama de clases del paquete arbolDirectorio	47
Diagrama de clases del paquete arbolImanifest	48
Diagrama de clases del paquete metedataLOM	49
Diagrama de clases del paquete socialización.....	50
Diagrama de clases del paquete publicación	51
Diagrama de clases del paquete arbol.....	52

Diagrama de clases del paquete misWidgets.....	53
Diagrama de clases del paquete xmlDOM	54
Diagrama de clases del paquete Servicios.....	55
Servidor	56
Diagrama general de clases del servidor	56
Diagrama de clases del paquete Auxiliary	57
TECNOLOGÍAS	59
Google Web Toolkit (GWT)	59
MySQL.....	63
Características de la version 5.0.22	63
Características adicionales.....	64
Tomcat.....	65
Java	67
Apache-Ant.....	68
XmlBeans	69
GESTION DE CONFIGURACIÓN.....	70
CONCLUSIONES.....	71
MANUAL DE USUARIO	73
Arrancar la aplicación mediante el explorador de Internet.....	73
Darse de alta como usuario.....	74
Iniciar sesión.....	75
Opciones del menú File	76
Crear un objeto de aprendizaje	77
Guardar un objeto de aprendizaje.....	77
Abrir un objeto de aprendizaje	77
Importar un objeto de aprendizaje al editor.....	78
Exportar un objeto de aprendizaje desde el editor.....	79
Generar un content preview para un objeto de aprendizaje.....	80
Salir de la aplicación	80
Opciones del menú Server	82
Guardar un objeto de aprendizaje en la carpeta personal del usuario	82
Eliminar un objeto de aprendizaje de la carpeta personal del usuario.....	83
Importar un objeto de aprendizaje de la carpeta personal del usuario al editor	84
Opciones del menú Publication	86
Publicar un objeto de aprendizaje.....	86
Búsqueda general de un objeto de aprendizaje.....	87
Búsqueda avanzada de un objeto de aprendizaje	89
Opciones del menú Database.....	90
Guardar en la base de datos una versión de un content preview	90
Eliminar un content preview almacenado en la base de datos	91
Descargar un content preview almacenado en la base de datos	91
Opciones del menú Foksonomy	93
Etiquetar un objeto de aprendizaje	93
Publicar un objeto por internet	94
Buscar objetos de aprendizaje mediante la nube de etiquetas.....	95
INSTALACIÓN DE LA APLICACIÓN GODACHEANT.....	97
Instalación de la máquina virtual de java	97
Instalación y configuración de MySQL.....	103
Configuración del conexiones.properties	120
Instalación Apache Tomcat	121

Configuración del server.properties	123
OTRAS INSTALACIONES	124
Instalación XmlBeans.....	124
Instalación del Apache-Ant-1.7.0.....	125
CREACIÓN DEL GODACHEANT.WAR Y DEL VALIDADOR.....	126
Introducción.....	126
Estructura del Proyecto:	126
Como crear el Godacheant.war	130
Desplegar la aplicación en el Tomcat de forma manual sin tener que crear el War	131
Estructura del Godacheant.war.....	131
Creación del validador a través del XmlBeans.....	132
Listado de la parte de código de los esquemas que se ha modificado.....	133
Common/elementNamens.xsd.....	133
Common/elementTypes.xsd	135
Common/vocabTypes.xsd	138
Common/vocabValues.xsd.....	141
Vocab/strict.xsd	147
Vocab/custom.xsd	148
Unique/strict.xsd.....	149
BIBLIOGRAFÍA	151
AUTORIZACIÓN DE DIFUSIÓN	152

RESÚMEN DEL PROYECTO

Este proyecto consiste en la realización de una herramienta online para gestionar objetos de aprendizaje. Permite editar y crear objetos de aprendizaje desde cero como cualquier editor, y generación de sus previsualizaciones en formato HTML. Además aporta funcionalidades novedosas como una completa gestión de cuentas de usuario con capacidad de almacenamiento online de objetos y previsualizaciones o la posibilidad de realizar búsquedas de objetos almacenados.

Pero lo que realmente lo diferencia de aplicaciones similares es la inclusión de un potente validador de objetos y de características de la web 2.0. El validador nos aporta la confianza de que el objeto creado se ajusta al estándar SCORM y funcionará en los entornos de ejecución existentes. Las citadas características de la web 2.0 se centran en la publicación de objetos tanto a nivel local como global, el etiquetado social y la búsqueda de objetos públicos etiquetados. Esto fomenta la colaboración y el intercambio ágil de información entre los usuarios del sistema.

El proyecto ha sido desarrollado con una potente tecnología llamada Google Web Toolkit (GWT), que permite hacer aplicaciones web con AJAX, y actualizar partes concretas de una Web sin recargar toda la página, lo cual deriva en que la aplicación sea más rápida y más vistosa para el usuario. Y lo más importante es que permite desarrollar en Java sin tener que aprender javascript y HTML, realizando automáticamente la traducción del código.

Con vistas a futuras ampliaciones, la aplicación es fácilmente extensible ya que esta bien modularizada en una parte cliente y otra servidor que es el que desarrolla toda la lógica de negocio que se llama a través de llamadas a procedimientos remotos que se localizan en el servidor.

Abstract

The goal of this Project is to create an online tool that manages learning objects. It allows editing and creating learning objects from the scratch like any editor, and it generates its previews in HTML format. It also adds new functionalities like complete management of user accounts with capacity for online storing of objects and previews or the possibility of searching through the library of stored objects.

But what really differences if from similar applications is the inclusion of a powerful objects validator and characteristics of the web 2.0. The validator gives the confidence that the created object adjusts to the SCORM standard and will work in an existing Runtime environment. The quoted characteristics of the web 2.0 are centered in the publication of objects, locally and globally, the social labeling and in the search of public labeled objects. This increases collaboration and an agile exchange of information between the users of the system.

The project has been developed with a powerful technology called Google Web Toolkit (GWT) that allows creating applications web with AJAX, and updating concrete parts of a web without reloading the entire page. This makes the application faster and showier for the user. And the most important thing is that it allows developing in Java without having to learn JavaScript and HTML, as GWT does automatically the code translation.

With views to future extensions, the application is easily extensible because it's very good organized in two parts, the client and the server that is the one who develops all the business logic that is called through remote procedure calls located in the server.

PALABRAS CLASE

- e-learning
- objeto de aprendizaje
- editor
- validación
- almacenamiento
- publicación
- búsqueda
- etiquetado

INTRODUCCIÓN A LA APLICACIÓN GODACHEANT

A continuación vamos a explicar en que consiste el SCORM que es la especificación que hemos utilizado para crear los objetos de aprendizaje de nuestra aplicación.

Después de explicar el SCORM hablaremos de una manera muy general las cosas que permite hacer la aplicación.

Scorm

SCORM (del inglés Sharable Content Object Reference Model) es una especificación que permite crear objetos pedagógicos estructurados. Los sistemas de gestión de contenidos en web originales usaban formatos propietarios para los contenidos que distribuían. Como resultado, no era posible el intercambio de tales contenidos. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM.

Los principales requerimientos que el modelo SCORM trata de satisfacer son:

Accesibilidad: capacidad de acceder a los componentes de enseñanza desde un sitio distante a través de las tecnologías web, así como distribuirlos a otros sitios.

Adaptabilidad: capacidad de personalizar la formación en función de las necesidades de las personas y organizaciones.

Durabilidad: capacidad de resistir a la evolución de la tecnología sin necesitar una reconcepción, una reconfiguración o una reescritura del código.

Interoperabilidad: capacidad de utilizarse en otro emplazamiento y con otro conjunto de herramientas o sobre otra plataforma de componentes de enseñanza desarrolladas dentro de un sitio, con un cierto conjunto de herramientas o sobre una cierta plataforma. Existen numerosos niveles de interoperabilidad.

Reusabilidad: flexibilidad que permite integrar componentes de enseñanza dentro de múltiples contextos y aplicaciones.

Estándares que conforman SCORM

SCORM es un conjunto de normas técnicas que permiten a los sistemas de aprendizaje en línea importar y reutilizar contenidos de aprendizaje que se ajusten al estándar. La organización ADL a cargo de este estándar no trabaja sola sobre este proyecto, sino en colaboración con numerosas organizaciones, que trabajan también con las especificaciones destinadas al aprendizaje en línea. De este modo, las especificaciones de las organizaciones siguientes han sido integradas a la norma SCORM.

No obstante, a menudo estas especificaciones han sido modificadas ligeramente a fin de hacer el conjunto más coherente.

Componentes de la especificación SCORM

La especificación SCORM esta dividida en "libros técnicos", Estos libros se agrupan en 3 tópicos principales:

Modelo de Agregación de Contenidos (Content Aggregation Model), que asegura métodos coherentes en materia de almacenamiento, de identificación, de condicionamiento de intercambios y de recuperación de contenidos.

Entorno de Ejecución (Run-Time Environment), describe las exigencias sobre el sistema de gestión del aprendizaje (SGA) que este debe implementar para que pueda gestionar el entorno de ejecución con el contenido SCORM.

Secuenciamiento y de navegación (Sequencing and Navigation), permite una presentación dinámica del contenido. Describe como el sistema interpreta las reglas de secuenciamiento introducidas por un desarrollador de contenidos, así como los eventos de navegación lanzados por el estudiante o por el sistema

Modelo de Agregación de Contenidos

El modelo de agregación de contenidos puede descomponerse en varias funcionalidades. La primera es la definición de «Learning Object Metadata» (LOM). Estos metadatos, utilizados dentro de los estándares de IEEE, de Ariadne y de IMS, permiten la definición de un diccionario de términos describiendo el contenido del objeto de aprendizaje. La segunda especificación une los metadatos y el/los archivo(s) XML, reutilizándose de IMS. Define como codificar los archivos XML a fin de que sean legibles por la máquina.

La última especificación trata del empaquetado. Define cómo empaquetar el conjunto de una colección de objetos de aprendizaje, sus metadatos, y las informaciones sobre la manera en que el contenido debe ser leído para el usuario. En la práctica, se trata de crear un archivo zip que contiene todos los ficheros apropiados, así como un fichero manifest., definiendo los contenidos de los diferentes ficheros y las relaciones entre ellos

Entorno de Ejecución

Una comunicación es necesaria entre el objeto pedagógico (más particularmente, el estudiante) y el sistema de aprendizaje (Learning Management System). Por ello, ADL ha trabajado en colaboración con AICC para establecer un envío estandarizado de la información entre los dos sentidos, y compatible con las tecnologías de Internet. Se ha definido una API (Application Program Interface) en Javascript, que suministra una manera estándar de comunicar con un LMS, independientemente de la herramienta utilizada para desarrollar el contenido.

Secuenciamiento y de navegación

Esta especificación describe el orden de la presentación de los contenidos según la navegación hecha por el usuario. Con este propósito se definen los llamados arboles de actividades, que definen las posibles ordenaciones según las acciones efectuadas por el usuario.

Acrónimos

Significado de los acrónimos utilizados

Acrónimo (en)	Significado (en)	Acrónimo (es)	Significado (es)
API	Application Program Interface		Interfaz de programación de aplicación
CAM	Content Aggregation Model	MAC	Modelo de Agregación de Contenido
LMS	Learning Management System	SGA	Sistema de Gestión del Aprendizaje
LOM	Learning Objects Metadata		Metadatos de los objetos de aprendizaje
SCORM	Sharable Content Object Reference Model		Modelo de Referencia de Objetos de Contenido Compartible
IEEE	Institute of Electrical and Electronics Engineers		Instituto de Ingenieros Eléctricos y Electrónicos
XML	eXtended Mark-up Language		Lenguaje de marcado extensible

Aplicación Godacheant

La aplicación es una herramienta on-line que se puede acceder desde cualquier parte del mundo a través de un ordenador que posea Internet y un navegador a través de la ruta <http://direccionServidor/Godacheant/Godacheant.html>.

A continuación se van a detallar de manera resumida las operaciones que permite realizar la aplicación.

Crear un objeto de aprendizaje

Nos permite crear un nuevo objeto de aprendizaje desde cero. Siempre que ejecutemos este comando la aplicación nos preguntará si estamos seguros de eliminar el objeto que hay actualmente abierto en el editor.

Guardar un objeto de aprendizaje

Mientras estamos creando un objeto nuevo también podemos ir guardando nuestros avances en la construcción.

Abrir un objeto de aprendizaje

Con esta opción del menú principal simplemente visualizamos en el editor el contenido del manifiesto en la ventana central.

Importar un objeto de aprendizaje al editor

Esta funcionalidad consiste en abrir un objeto de aprendizaje, que tenemos guardado localmente en disco, en el editor, para ello es necesario borrar el objeto de aprendizaje actual.

Exportar un objeto de aprendizaje desde el editor

Nos permite guardar una copia real del objeto de aprendizaje que estamos creando, o hemos creado ya. Esta función es muy importante ya que es el paso previo para poder almacenar objetos en nuestra carpeta personal del servidor.

Generar un content preview para un objeto de aprendizaje

El content preview de un objeto de aprendizaje es el contenido del objeto de aprendizaje. Con él podemos ver todos y cada uno de los detalles que aporta en formato html, junto con todos los recursos de los que dispone. Por supuesto la organización del archivo será la que hemos definido en el momento de creación del objeto.

Guardar un objeto de aprendizaje en la carpeta personal del usuario

Nos permite guardar un objeto de aprendizaje en nuestra carpeta personal alojada en el servidor.

Eliminar un objeto de aprendizaje de la carpeta personal del usuario

Nos permite eliminar uno de los objetos de aprendizaje que tenemos alojado en nuestra carpeta personal.

Importar un objeto de aprendizaje de la carpeta personal del usuario al editor

Nos permite abrir uno de los objetos que tenemos almacenados en nuestra carpeta privada en el editor. Cuando realicemos esta acción la aplicación nos dirá que si seguimos eliminaremos el objeto que actualmente tenemos en el editor.

Publicar un objeto de aprendizaje

Nos permite publicar uno de nuestros objetos de aprendizaje, desde donde será visible para el resto de usuarios.

Búsqueda general de un objeto de aprendizaje

Realiza una búsqueda general sobre todos los objetos públicos,

Búsqueda avanzada de un objeto de aprendizaje

A diferencia de con la general, podemos elegir los campos por los que buscar. El buscador solo buscará coincidencias para los campos que queramos.

Guardar en la base de datos una versión de un content preview

Permite guardar una versión de un content preview del objeto que tenemos abierto en el editor.

Eliminar un content preview almacenado en la base de datos

Nos permite eliminar un content preview de la base de datos.

Descargar un content preview almacenado en la base de datos

Sirve para recuperar cualquier archivo que hayamos subido a la base de datos y descargarlo localmente.

Etiquetar un objeto de aprendizaje

Podemos aplicar etiquetas a los objetos de aprendizaje que indiquen su temática, para orientar a los demás usuarios y poder realizar búsquedas sobre ellas.

Publicar un objeto de aprendizaje por internet

Esta funcionalidad sirve para hacer un objeto de aprendizaje visible desde fuera de la aplicación a través de conocidas páginas web de socialización como digg o del.icio.us y compartirlos con usuarios que busquen objetos de su temática.

Buscar objetos de aprendizaje por sus etiquetas

La aplicación permite realizar búsquedas de objetos de aprendizaje por sus etiquetas, generando una nube de etiquetas que orienta la búsqueda resaltando aquellas etiquetas más importantes.

CASOS DE USO

Iniciar sesion

Iniciar sesión
Descripción: El usuario introduce su nombre de usuario y contraseña para abrir el editor e identificarse.
Actores: Usuario, cliente y servidor
Precondiciones: Es necesario darse de alta previamente.
Flujo Normal: <ol style="list-style-type: none">1. El usuario introduce su nombre de usuario y su contraseña y pulsa “Enter”2. El cliente manda el nombre de usuario y la contraseña al servidor3. El servidor recibe los datos y comprueba con la base de datos si el usuario existe y si coincide la contraseña4. El servidor confirma al cliente que los datos son correctos5. El cliente arranca el editor y lo deja listo para ser usado
Flujo Alternativo: <ol style="list-style-type: none">4. El nombre de usuario no existe o la contraseña es incorrecta, y el cliente muestra un mensaje informando del error.
Postcondiciones: El usuario dispone de la aplicación arrancada y el editor abierto, con un objeto de aprendizaje en blanco, listo para ser usado.

Darse de alta

Darse de alta
Descripción: Se crea una nueva cuenta de usuario.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario introduce el nombre de usuario elegido y la confirma la contraseña dos veces.2. El cliente manda los datos al servidor3. El servidor consulta con la base de datos si no existe un usuario con el mismo nombre y da de alta al nuevo usuario.4. El cliente recibe la confirmación y muestra la ventana para iniciar sesión.
Flujo Alternativo: <ol style="list-style-type: none">3. Ya existe un usuario con el mismo nombre y muestra un mensaje informando de ello al usuario.
Postcondiciones: El usuario ya tiene creada una cuenta y puede iniciar sesión.

Crear un OA en el editor

Crear un OA con el editor
Descripción: Crear un objeto de aprendizaje nuevo en blanco eliminando el que hubiese anteriormente abierto.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">5. El usuario selecciona la opción “Nuevo objeto” del menú para crear un objeto en blanco.6. El cliente pregunta al usuario si desea eliminar el OA que hay actualmente abierto con el editor y el usuario acepta.7. El cliente manda la petición al servidor.8. El servidor elimina todo el contenido del anterior objeto existente en el editor, crea un nuevo objeto de aprendizaje y manda el contenido del manifiesto al cliente.9. El cliente lo recibe y visualiza tanto el contenido del nuevo objeto como el contenido del manifiesto al usuario.
Flujo Alternativo: <ol style="list-style-type: none">2. El usuario selecciona la opción cancelar y no se elimina el objeto actual del editor ni se crea un nuevo.
Postcondiciones: El usuario dispone de un nuevo objeto de aprendizaje en blanco y listo para ser editado.

Abrir un OA en el editor

Abrir un OA en el editor
Descripción: Visualizar el contenido del manifiesto del objeto actualmente abierto del editor.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción “Abrir un OA” del menú principal.2. El cliente manda la solicitud al servidor.3. El servidor busca el manifiesto del OA actual, lee el contenido y se lo manda al cliente.4. El cliente recibe el contenido del manifiesto y lo visualiza a través del editor al usuario.
Flujo Alternativo: <ol style="list-style-type: none">3. No existe manifiesto para el OA actual y notifica de ello al cliente. El cliente muestra un mensaje al usuario de error.
Postcondiciones: El usuario dispone del manifiesto, del OA actual, visible en el editor.

Guardar el OA actual en el editor

Guardar el OA actual abierto en el editor
Descripción: Se guardan todos los cambios efectuados en el objeto de aprendizaje actual abierto con el editor.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción del menú principal “Guardar OA”.2. El cliente notifica al servidor de la solicitud de guardar el OA actual y le envía el contenido del manifiesto3. El servidor recibe la solicitud y guarda el manifiesto en la carpeta temporal de la sesión actual del usuario con todos los cambios efectuados desde la última vez que se guardó.
Flujo Alternativo:
Postcondiciones: El usuario dispone del objeto de aprendizaje actual guardado en servidor de forma temporal.

Crear un content preview

Crear un content preview
Descripción: Se crea un content preview para la versión actual del objeto de aprendizaje abierto en el editor.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario elige la opción “Content preview” del menú principal2. El cliente envía al servidor el contenido del manifiesto actual para que genere correctamente el content preview.3. El servidor recibe la solicitud y genera el content preview para el xml recibido4. El cliente recibe la confirmación y pregunta al usuario si desea abrirlo, guardarlo o cancelar.5. El usuario decide guardarlo localmente y tras haber elegido destino se guarda el archivo.
Flujo Alternativo: <ol style="list-style-type: none">5. a. El usuario decide abrirlo, por lo que se visualiza el contenido del archivo sin almacenarse localmente.5. b. El usuario decide cancelar y no se almacena el archivo localmente.
Postcondiciones: El usuario dispone de una versión de un content preview del OA abierto actualmente en el editor, almacenado localmente en disco.

Importar un OA al editor

Importar un OA al editor
Descripción: Se importa y se visualiza en el editor el contenido de un objeto de aprendizaje guardado localmente en disco.
Actores: Usuario, cliente y servidor
Precondiciones: Es necesario que el usuario disponga de un OA, correctamente generado, guardado localmente.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción del menú principal “Importar paquete zip”2. El cliente pregunta al usuario si desea eliminar el OA actual3. Tras haber aceptado, se borra el contenido el OA actual y se abre un explorador de archivos para que el usuario seleccione un objeto a importar4. El cliente manda la solicitud al servidor5. El servidor valida el objeto de aprendizaje, y lo descomprime en la carpeta temporal de la sesión actual del usuario6. El cliente recibe la confirmación y visualiza el contenido del objeto de aprendizaje en el editor.
Flujo Alternativo: <ol style="list-style-type: none">3. El usuario rechaza y no decide abrir ningún OA.5. El objeto de aprendizaje que el usuario desea abrir no pasa el validador, mostrándose una lista con los errores que contiene el objeto.
Postcondiciones: Si flujo normal: el usuario dispone del objeto de aprendizaje seleccionado abierto y listo para usarse en el editor. Si flujo alternativo: Si 3. No cambia el estado actual del editor Si 5. No se llega a visualizar el objeto elegido por el usuario y se ha eliminado el objeto que previamente estaba abierto en el editor.

Exportar un OA

Exportar un OA
Descripción: Se comprime el contenido del objeto de aprendizaje actualmente abierto en el editor para obtener una copia almacenada localmente en disco.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario elige la opción “Exportar paquete zip” del menú principal.2. El cliente manda la solicitud al servidor3. El servidor genera el contenido del objeto de aprendizaje actual en un archivo comprimido zip6. El cliente recibe la confirmación y pregunta al usuario si desea abrirlo, guardarlo o cancelar.4. El usuario decide guardarlo localmente y tras haber elegido destino se guarda el archivo
Flujo Alternativo: <ol style="list-style-type: none">5. a. El usuario decide abrirlo, por lo que se visualiza el contenido del archivo sin almacenarse localmente.5. b. El usuario decide cancelar y no se almacena el archivo localmente.
Postcondiciones: El usuario dispone de una versión comprimida en .zip del OA abierto actualmente en el editor, almacenado localmente en disco.

Salir de la aplicación

Salir de la aplicación
Descripción: Se cierra sesión, el editor y se muestra una pantalla de fin.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción “Salir” del menú principal2. El cliente notifica de ello al servidor3. El servidor elimina la sesión actual del usuario junto con el objeto de aprendizaje actual4. El cliente recibe la confirmación y muestra una pantalla de fin al usuario
Flujo Alternativo:
Postcondiciones: Se ha cerrado la sesión y el usuario se dispone a cerrar la aplicación por completo.

Publicar un OA

Publicar un OA
Descripción: El usuario decide publicar un objeto de aprendizaje almacenado en su carpeta personal.
Actores: Usuario, cliente y servidor
Precondiciones: El usuario debe disponer de algún OA en su carpeta personal
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona desde el cliente un objeto de su carpeta personal y lo publica.2. El servidor recibe el objeto, lo almacena en el repositorio y actualiza la base de datos.3. El servidor manda un mensaje de confirmación4. El cliente recibe el mensaje y lo muestra al usuario.
Flujo Alternativo: <ol style="list-style-type: none">2. Ya existe un OA publicado por el usuario con el mismo nombre indicándose al cliente El cliente pregunta al usuario si quiere sobrescribir el OA existente<ol style="list-style-type: none">a. El usuario decide sobrescribir el OA y reemplazarlo por el nuevo. Ir a 3 flujo normal.b. El usuario decide no sobrescribirlo y cancela la publicación del OA.
Postcondiciones: <p>Si flujo normal: el usuario publica un objeto de aprendizaje que posteriormente podrá ser consultado por otros usuarios.</p> <p>Si flujo alternativo: el usuario publica el OA seleccionado pero elimina el que hubiese anteriormente con el mismo nombre.</p>

Búsqueda general de un OA

Búsqueda general de OA's públicos
Descripción: Se realiza una búsqueda general por todos los objetos de aprendizaje públicos.
Actores: Usuario, cliente y servidor.
Precondiciones: Que el usuario haya introducido algún carácter en la línea de texto.
Flujo Normal: <ol style="list-style-type: none">1. El usuario realiza una búsqueda general.2. El cliente manda al servidor la búsqueda solicitada por el usuario3. El servidor busca todas las coincidencias posibles y se las manda al cliente.4. El cliente visualiza todo los OA's encontrados que coinciden con la búsqueda solicitada.
Flujo Alternativo: 4 ^a . Si no existen resultados, el cliente visualiza un mensaje informando de ello.
Postcondiciones: El usuario dispone de una lista de resultados, divididos en páginas, y para cada uno de ellos dispone de la posibilidad de descargarlo localmente o de añadirlo a su carpeta personal.

Búsqueda avanzada de un OA

Búsqueda avanzada de un OA
Descripción: Se realiza una búsqueda avanzada por los objetos de aprendizaje públicos según los criterios elegidos por el usuario.
Actores: Usuario, cliente y servidor.
Precondiciones: El usuario ha rellenado alguno de los campos de búsqueda.
Flujo Normal: <ol style="list-style-type: none">1. El usuario realiza una búsqueda avanzada rellenando uno o más campos de búsqueda y pincha en el botón buscar2. El cliente manda al servidor todos los criterios introducidos por el usuario3. El servidor realiza una búsqueda sólo para aquellos objetos de aprendizaje que contengan información relevante con los criterios requeridos4. El servidor manda la lista de resultados obtenidos al cliente.5. El cliente ordena la lista de resultados recibida según el número de coincidencias de cada objeto encontrado con los criterios seleccionados y los visualiza por pantalla
Flujo Alternativo: <ol style="list-style-type: none">5. Si no existen coincidencias con la búsqueda solicitada el cliente informa de ello.
Postcondiciones: El usuario dispone de una lista de resultados, divididos en páginas, y para cada uno de ellos dispone de la posibilidad de descargarlo localmente o de añadirlo a su carpeta personal.

Descargar un OA público

Descargar un OA público
Descripción: Se realiza una descarga local de un objeto de aprendizaje previamente publicado.
Actores: Usuario, cliente y servidor.
Precondiciones: Se ha tenido que realizar una búsqueda general o avanzada habiendo encontrado al menos un resultado
Flujo Normal: <ol style="list-style-type: none">1. El usuario pincha en el botón “descargar” del objeto de aprendizaje encontrado.2. El cliente manda la petición al servidor indicándole el OA del que se trata.3. El servidor recibe la petición, busca el objeto solicitado e inicia la descarga.4. Una vez finalizada, el usuario le da a aceptar y elige destino para la descarga, copiándose el archivo localmente.
Flujo Alternativo: <ol style="list-style-type: none">4ª. El usuario le da a cancelar y la descarga se anula.
Postcondiciones: El usuario dispone de una copia local del objeto de aprendizaje seleccionado.

Añadir un OA público a la carpeta personal

Añadir un OA público a la carpeta personal
Descripción: Se añade un OA, previamente publicado, a la carpeta personal del usuario.
Actores: Usuario, cliente y servidor.
Precondiciones: Se ha tenido que realizar una búsqueda general o avanzada habiendo encontrado al menos un resultado
Flujo Normal: <ol style="list-style-type: none">1. El usuario pincha en el botón “subir” del objeto de aprendizaje encontrado.2. El cliente manda la petición al servidor indicándole el OA del que se trata.3. El servidor recibe la petición, busca el objeto solicitado y lo incluye en la carpeta personal del usuario actualizando la base de datos.
Flujo Alternativo: <ol style="list-style-type: none">3. Ya existe un objeto de aprendizaje con el mismo nombre en la carpeta personal del usuario y el servidor informa al cliente de ello.<ol style="list-style-type: none">a. El usuario decide sobrescribir el OA y reemplazarlo por el nuevo.b. El usuario decide no sobrescribirlo y la subida del OA se cancela.
Postcondiciones: El usuario dispone de un nuevo objeto de aprendizaje en su carpeta personal.

Guardar un OA en la carpeta personal

Guardar un OA en la carpeta personal
Descripción: El usuario decide subir un objeto de aprendizaje a su carpeta personal.
Actores: Usuario, cliente, servidor
Precondiciones: El usuario debe disponer de un OA ya creado guardado en disco localmente.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona el objeto de aprendizaje que desea subir.2. El cliente manda la petición al servidor junto con el objeto y el nombre de usuario.3. El servidor comprueba que cumple el formato apropiado y valida el OA.4. Una vez validado lo almacena en la carpeta personal del usuario y lo inserta en la base de datos.5. El servidor manda un mensaje de confirmación al cliente.
Flujo Alternativo: <ol style="list-style-type: none">3^a. El archivo no es del formato apropiado, y manda un mensaje de error al cliente. El cliente muestra el mensaje de error al usuario.3^b. El objeto de aprendizaje no es validado al no cumplir con el estándar y el manda al cliente la lista de errores que contiene el OA. El cliente muestra por pantalla al usuario la lista de errores que contiene el OA y se cancela el almacenamiento del archivo en el servidor.
Postcondiciones: Si flujo normal: el usuario dispone del objeto de aprendizaje seleccionado en su carpeta personal Si flujo alternativo: no se llega a almacenar el OA en la carpeta personal del usuario.

Eliminar un OA de la carpeta personal

Eliminar un OA de la carpeta personal
Descripción: El usuario selecciona un OA de su carpeta personal para ser eliminado.
Actores: Usuario, cliente y servidor.
Precondiciones: El usuario debe disponer de algún objeto de aprendizaje en su carpeta personal.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona un objeto de aprendizaje de su carpeta personal y pincha en el botón “eliminar”2. El cliente manda la orden al servidor con el nombre de OA y el de usuario.3. El servidor elimina físicamente el OA de la carpeta y su entrada en la base de datos.
Flujo Alternativo:
Postcondiciones: El usuario deja de disponer el OA seleccionado en su carpeta personal.

Importar un OA al editor desde la carpeta personal

Importar un OA al editor desde la carpeta personal
Descripción: El usuario selecciona un objeto de su carpeta personal para visualizarlo en el editor.
Actores: Usuario, cliente y servidor
Precondiciones: Es necesario que el usuario disponga de al menos un OA en su carpeta personal.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona un objeto de aprendizaje de su carpeta personal y pincha en “importar”.2. El cliente manda la solicitud al servidor junto con el nombre de usuario y el nombre del OA.3. El servidor importa el archivo.4. El cliente visualiza el objeto de aprendizaje elegido con el editor.
Flujo Alternativo: <ol style="list-style-type: none">4. El cliente pregunta al usuario si desea eliminar el OA actual abierto en el editor y el usuario decide no eliminarlo. Se cancela la importación del OA.
Postcondiciones: El usuario dispone del objeto de aprendizaje elegido abierto en el editor para poder manejarlo.

Guardar un content preview en la base de datos

Guardar un content preview en la base de datos
Descripción: El usuario selecciona una versión de un content preview de un objeto de aprendizaje, previamente generada y almacenada localmente en disco, para almacenarla en la base de datos.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona el archivo a subir.2. El cliente manda la solicitud de almacenar el archivo en la base de datos al servidor.3. El servidor recibe la solicitud y comprueba que el archivo sea del formato adecuado, y que el OA actual abierto en el editor sea el que corresponde al content preview.4. El servidor almacena el archivo en la base de datos.5. Confirma de ello al cliente.
Flujo Alternativo: <ol style="list-style-type: none">3. El archivo recibido no cumple con alguna de las restricciones<ol style="list-style-type: none">3^a. El archivo no es del formato adecuado y no se almacena en la base de datos3b. El OA actual del editor no corresponde con la versión del content preview que se quiere almacenar en la base de datos y se cancela la operación.4. El usuario ya dispone de un content preview con el mismo nombre en la base de datos y informa al cliente de ello.<ol style="list-style-type: none">4^a. El usuario decide sobrescribir el content preview antiguo por el nuevo.4b. El usuario decide cancelar y no sobrescribirlo.
Postcondiciones: <p>Si flujo normal: el usuario dispone de una versión en la base de datos del content preview del OA actual.</p> <p>Si flujo alternativo:</p> <ol style="list-style-type: none">3^a, 3b. 4b: no cambia nada.4^a: el usuario dispone de una nueva versión del content preview actual y elimina la versión anterior que tuviese almacenada en la base de datos.

Eliminar un content preview de la base de datos

Eliminar un content preview de la base de datos
Descripción: El usuario decide eliminar un content preview disponible en la base de datos.
Actores: Usuario, cliente y servidor
Precondiciones: Es necesario que el usuario disponga de al menos un content preview almacenado en la base de datos.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona un content preview disponible en la base de datos y pincha el botón eliminar.2. El cliente manda la orden al servidor junto al nombre de usuario y el nombre de archivo.3. El servidor elimina la entrada de la base de datos correspondiente al archivo seleccionado del usuario que lo solicita.
Flujo Alternativo:
Postcondiciones: El usuario deja de disponer del OA seleccionado en la base de datos.

Descargar un content preview de la base de datos

Descargar un content preview de la base de datos
Descripción: El usuario decide descargarse un content preview que tiene almacenado en la base de datos.
Actores: Usuario, cliente y servidor.
Precondiciones: Es necesario que el usuario disponga de al menos un content preview almacenado en la base de datos.
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona un content preview que tenga almacenado en la base de datos y pincha en el botón “descargar”.2. El cliente manda la solicitud al servidor junto con el nombre de usuario y el nombre de archivo.3. El servidor selecciona el archivo solicitado e inicia la descarga del mismo.4. El cliente pregunta al usuario si acepta la descarga, el usuario la acepta y se almacena el archivo elegido localmente en disco.
Flujo Alternativo: <ol style="list-style-type: none">4. El usuario no acepta la descarga y le da a cancelar. No se llega a almacenar el archivo seleccionado.
Postcondiciones: El usuario dispone de una copia local del content preview elegido de la base de datos.

Etiquetar un OA

Etiquetar un OA
Descripción: El usuario etiqueta un objeto de aprendizaje
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción “Tag object” del menú Folksonomy.2. El cliente manda el nombre de usuario al servidor.3. El servidor recibe los datos y devuelve todos los objetos de ese usuario.4. El cliente recibe la lista de objetos y actualiza el desplegable con los objetos.5. El usuario selecciona el objeto que desea etiquetar, añade las etiquetas que desea y pulsa el botón “Tag it”.6. El cliente envía el objeto, usuario, orden de no modificar y la lista de etiquetas al servidor.7. El servidor recibe los datos y actualiza la base de datos con las nuevas etiquetas.
Flujo Alternativo: <ol style="list-style-type: none">5. El usuario no tiene ningún objeto que etiquetar y todas las opciones están deshabilitadas.
Postcondiciones: El objeto elegido por el usuario queda etiquetado con las nuevas etiquetas y las que tuviera anteriormente distintas a las nuevas.

Modificar las etiquetas de un OA

Modificar las etiquetas de un objeto
Descripción: El usuario modifica las etiquetas que tiene aplicadas un objeto.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción “Tag object” del menú Folksonomy.2. El cliente manda el nombre de usuario al servidor.3. El servidor recibe los datos y devuelve todos los objetos de ese usuario.4. El cliente recibe la lista de objetos y actualiza el desplegable con los objetos.5. El usuario selecciona el objeto que desea etiquetar, añade las etiquetas que desea, marca el checkbox “Reset object tags” y pulsa el botón “Tag it”.6. El cliente envía el objeto, usuario, la orden de modificar y la lista de etiquetas al servidor.7. El servidor recibe los datos, y actualiza la base de datos borrando las etiquetas anteriores y añadiendo las nuevas etiquetas.
Flujo Alternativo: <ol style="list-style-type: none">5. El usuario no tiene ningún objeto y todas las opciones están deshabilitadas.
Postcondiciones: El objeto elegido por el usuario queda etiquetado solo con las nuevas etiquetas

Publicar por Internet un OA

Publicar por Internet un OA
Descripción: El usuario publica por internet un objeto de aprendizaje
Actores: Usuario, cliente, servidor y web remota
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción "Tag object" del menú Folksonomy.2. El cliente manda el nombre de usuario al servidor.3. El servidor recibe los datos y devuelve todos los objetos de ese usuario.4. El cliente recibe la lista de objetos y actualiza el desplegable con los objetos.5. El usuario selecciona en el desplegable el objeto que desea publicar y clicla sobre el nombre de una de las web de publicación.6. El cliente envía a la web remota la URL estática del objeto de aprendizaje.7. El usuario inicia sesión o se registra en la web remota, rellena los campos y acepta.8. La web remota publica la URL del objeto de aprendizaje.
Flujo Alternativo: <ol style="list-style-type: none">5. El usuario no tiene ningún objeto y todas las opciones están deshabilitadas.
Postcondiciones: El objeto elegido por el usuario queda publicado en la web elegida con las etiquetas que el usuario escribe.

Generar nube de etiquetas

Generar nube de etiquetas
Descripción: El usuario desea realizar una búsqueda de objetos de aprendizaje por sus etiquetas y se genera una nube de etiquetas.
Actores: Usuario, cliente y servidor
Precondiciones:
Flujo Normal: <ol style="list-style-type: none">1. El usuario selecciona la opción “Tags cloud” del menú Folksonomy.2. El cliente manda al servidor el nombre de usuario.3. El servidor lo recibe y devuelve una lista de objetos públicos o propiedad del usuario, con sus etiquetas.4. El cliente lo recibe y genera la nube de etiquetas y muestra los resultados.
Flujo Alternativo: <ol style="list-style-type: none">4. No hay objetos etiquetados en el servidor y no se genera la nube ni hay resultados de búsqueda.
Postcondiciones: El usuario dispone de una nube de etiquetas y una lista de todos los objetos de aprendizaje

Refinar la búsqueda

Refinar la búsqueda
Descripción: El usuario refina la búsqueda para encontrar objetos más específicos.
Actores: Usuario, cliente y servidor
Precondiciones: Ya se ha generado la nube de etiquetas y opcionalmente se ha refinado la búsqueda alguna vez.
Flujo Normal: <ol style="list-style-type: none">1. El usuario clicca sobre alguna etiqueta de la nube.2. El cliente manda al servidor el nombre de usuario y una lista con las etiquetas que se deben buscar.3. El servidor lo recibe y devuelve una lista de objetos públicos o propiedad del usuario que tienen todas las etiquetas que buscamos, con sus etiquetas.4. El cliente lo recibe y genera la nueva nube de etiquetas y muestra los resultados.
Flujo Alternativo:
Postcondiciones: El usuario dispone una lista ordenada de objetos de aprendizaje que tienen todas las etiquetas que buscamos y una nube con sus etiquetas.

Reiniciar la búsqueda

Reiniciar la búsqueda
Descripción: El usuario reinicia la búsqueda.
Actores: Usuario, cliente y servidor
Precondiciones: Ya se ha generado la nube de etiquetas y opcionalmente se ha refinado la búsqueda alguna vez.
Flujo Normal: <ol style="list-style-type: none">1. El usuario clica sobre “Beginning”.2. El cliente manda al servidor el nombre de usuario.3. El servidor lo recibe y devuelve una lista de objetos públicos o propiedad del usuario, con sus etiquetas.4. El cliente lo recibe y genera la nube de etiquetas y muestra los resultados.
Flujo Alternativo: <ol style="list-style-type: none">1. El usuario cierra la pestaña Tags cloud.2. El usuario vuelve a seleccionar la opción “Tags cloud” del menú “Folksonomy”.
Postcondiciones:

Retroceder en la búsqueda

Retroceder en la búsqueda
Descripción: El usuario desea ver una de las nubes de etiquetas anteriores.
Actores: Usuario, cliente y servidor
Precondiciones: Ya se ha generado la nube de etiquetas y se ha refinado la búsqueda alguna vez.
Flujo Normal: <ol style="list-style-type: none">1. El usuario clica sobre alguna de las etiquetas de la lista en la parte inferior de la pantalla que están interviniendo en la búsqueda2. El cliente manda al servidor el nombre de usuario y una lista con las etiquetas que se deben buscar.3. El servidor lo recibe y devuelve una lista de objetos públicos o propiedad del usuario que tienen todas las etiquetas que buscamos, con sus etiquetas.4. El cliente lo recibe y genera la nueva nube de etiquetas y muestra los resultados.
Flujo Alternativo:
Postcondiciones: El usuario dispone de una de las nubes de etiquetas y una lista de resultados anteriores.

Abrir un OA público

Abrir un OA público
Descripción: El usuario abre en el editor uno de los OA públicos.
Actores: Usuario, cliente y servidor.
Precondiciones: Se ha tenido que realizar una búsqueda habiendo encontrado al menos un resultado
Flujo Normal: <ol style="list-style-type: none">1. El usuario pincha en el botón “Open” del objeto de aprendizaje encontrado.2. El cliente manda la petición al servidor indicándole el OA del que se trata.3. El servidor recibe la petición, busca el objeto solicitado y se lo envía al cliente.4. El cliente muestra un mensaje de advertencia de que se va a eliminar el objeto que esté en uso.5. El usuario pulsa “Aceptar”.6. El cliente abre el objeto en el editor.
Flujo Alternativo: <ol style="list-style-type: none">5. El usuario pulsa “Cancelar” y no sucede nada
Postcondiciones:

Etiquetar un OA público

Etiquetar un OA público
Descripción: El usuario etiqueta uno de los OA públicos.
Actores: Usuario, cliente y servidor
Precondiciones: Se ha tenido que realizar una búsqueda habiendo encontrado al menos un resultado
Flujo Normal: <ol style="list-style-type: none">1. El usuario pincha en el botón “Tag” del objeto de aprendizaje encontrado.2. El cliente abre la pestaña “Tag object” con el objeto seleccionado y las opciones destinadas al propietario del objeto deshabilitadas.3. El cliente añade las etiquetas que desea y pulsa el botón “Tag it”.4. El cliente envía el objeto, usuario propietario del objeto, orden de no modificar y la lista de etiquetas al servidor.5. El servidor recibe los datos y actualiza la base de datos con las nuevas etiquetas.
Flujo Alternativo:
Postcondiciones: El objeto elegido por el usuario queda etiquetado con las nuevas etiquetas y las que tuviera anteriormente distintas a las nuevas.

DISEÑO

A continuación se muestran los diagramas de clase del diseño que se ha establecido para el proyecto. Hay dos partes bien definidas que son el cliente y el servidor, al tratarse de una aplicación Web.

El cliente es el encargado de gestionar todo lo relativo a los aspectos gráficos del editor y de controlar la comunicación con el servidor cuando el usuario ejecute algún comando.

El servidor es el que almacena y gestiona los datos, y el encargado de responder a las distintas solicitudes del cliente.

Cliente

El diagrama principal del cliente que se muestra abajo define las clases principales de todas las funcionalidades que se implementan. Vemos como la mayor parte de clases que aparecen hacen referencia a los aspectos gráficos. Contiene una serie de editores para distintos aspectos (manifiesto, metadatos, socialización, publicación...), así como pantallas y paneles.

El cliente a su vez se organiza en distintos paquetes de clases e interfaces, que a su vez se pueden agrupar en tres tipos.

En primer lugar tenemos los paquetes `arbolManifest`, `arbolDirectorio`, `metadataLOM`, `socialización` y `publicación`. Estos son los 5 paquetes que contienen cada uno algún editor, independientes unos de otros, y sus respectivas estructuras de datos. `ArbolManifest` alberga todo lo relacionado con el editor que muestra el manifiesto del objeto de aprendizaje. `ArbolDirectorio` hace referencia al árbol del directorio de archivos que tenemos actualmente visible, para el objeto de aprendizaje. `MetadataLOM` contiene todo lo relativo con la edición de metadatos de los objetos de aprendizaje. Contiene un editor específico para ello. `Socialización` hace referencia a la parte de etiquetado de los objetos de aprendizaje, a la búsqueda usando la nubes de etiquetas asociadas a los objetos públicos y de la publicación de objetos por Internet. `Publicación` contiene todo lo necesario para publicar objetos y un editor para realizar búsquedas y poder recuperar los archivos publicados.

Por otro lado tenemos los paquetes `arbol`, `misWidgets`, `varios` y `xmlDOM`. Estos se usan como apoyo a todos los anteriores y definen distintas estructuras de datos y manejadores para su correcto uso.

Por último tenemos el paquete `Servicios`. Este es el encargado de gestionar la comunicación con el servidor a base de los denominados “servicios”. Existe un servicio por cada distinta interacción que exista entre el cliente y el servidor (Ej: abrir un OA, salir de la aplicación, publicar un OA, importar un OA, obtener mis objetos...). Cada uno de estos servicios es un `RemoteService` y tiene asociado un hermano gemelo en la parte del servidor, como veremos a continuación.

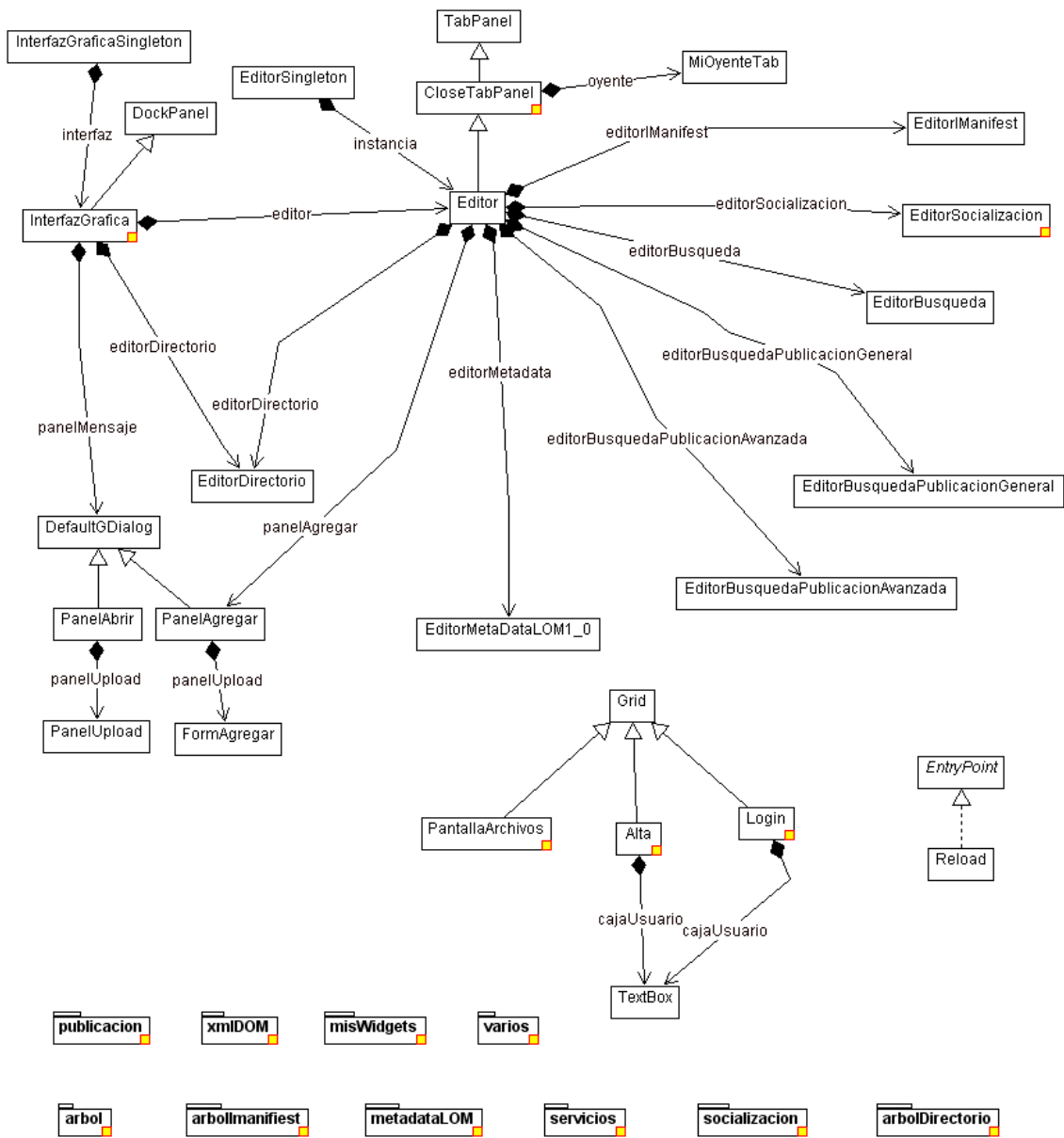


Diagrama general de clases del cliente

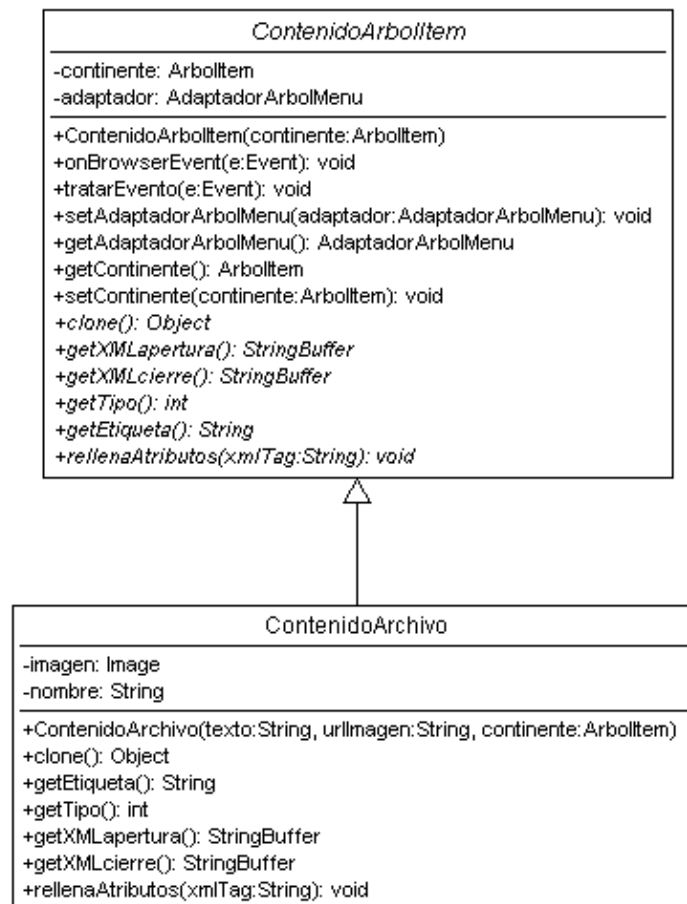
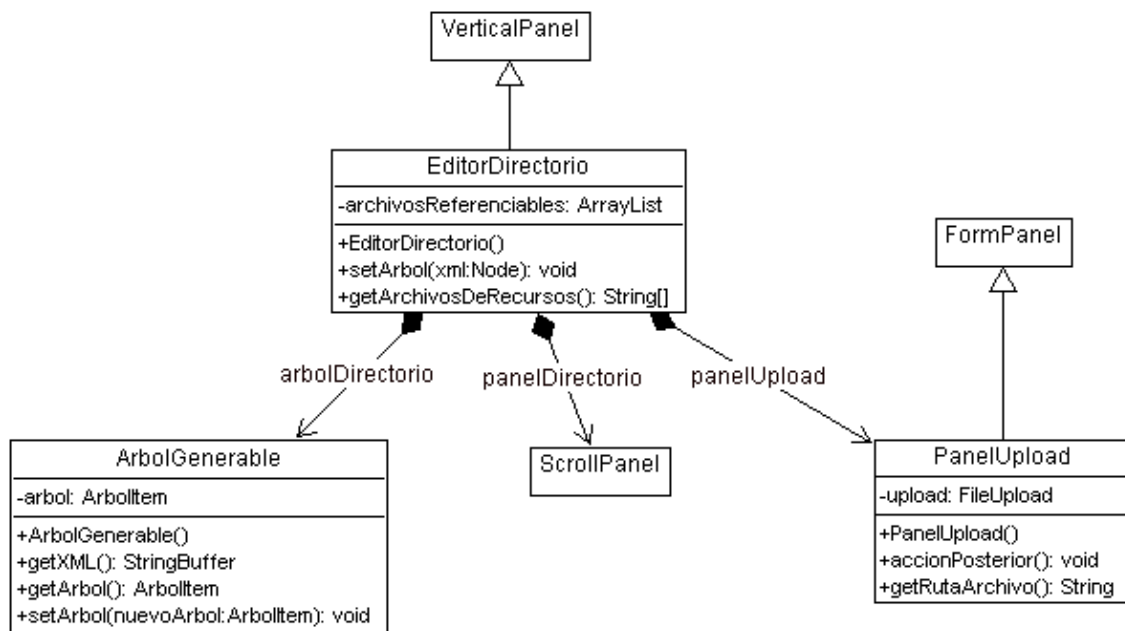


Diagrama de clases del paquete arbolDirectorio

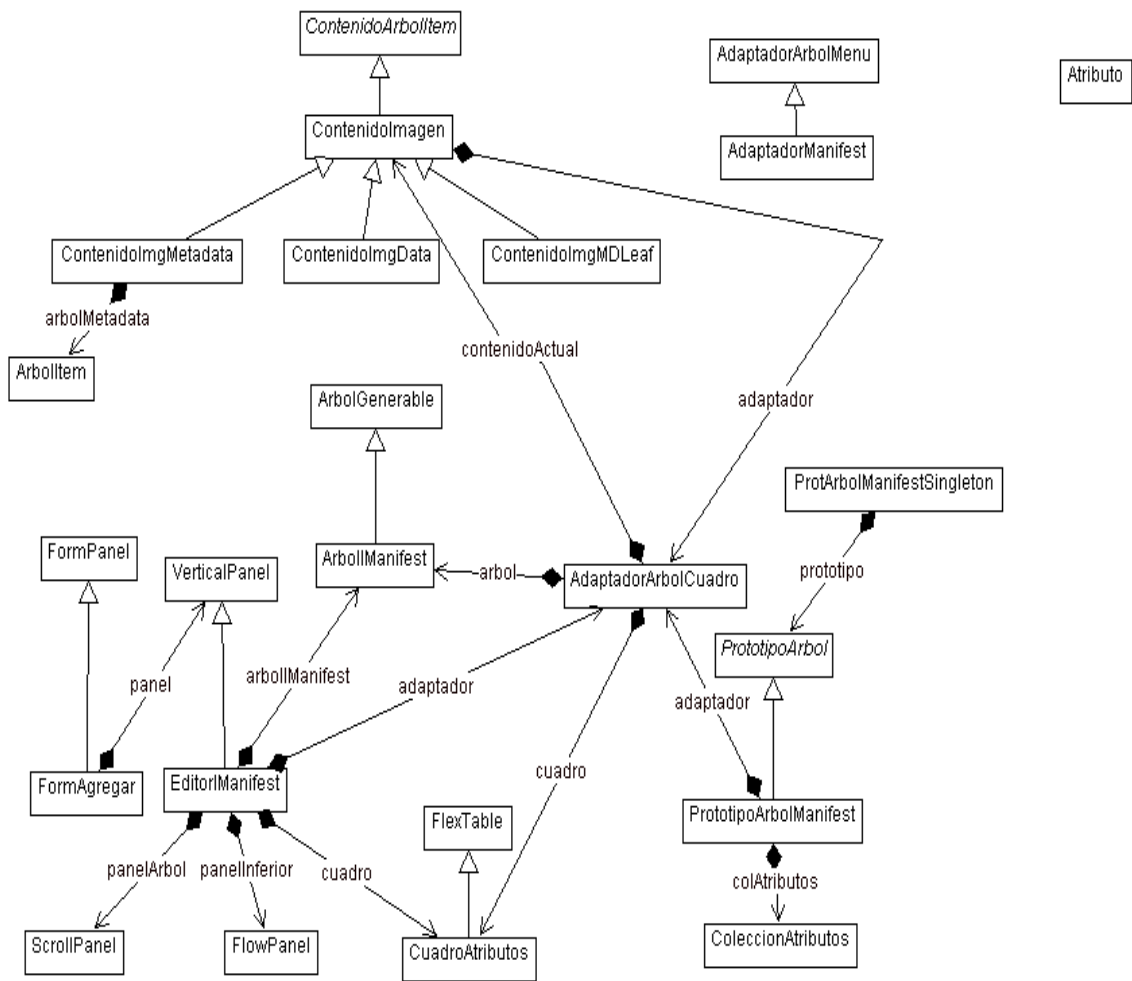


Diagrama de clases del paquete arbolManifest

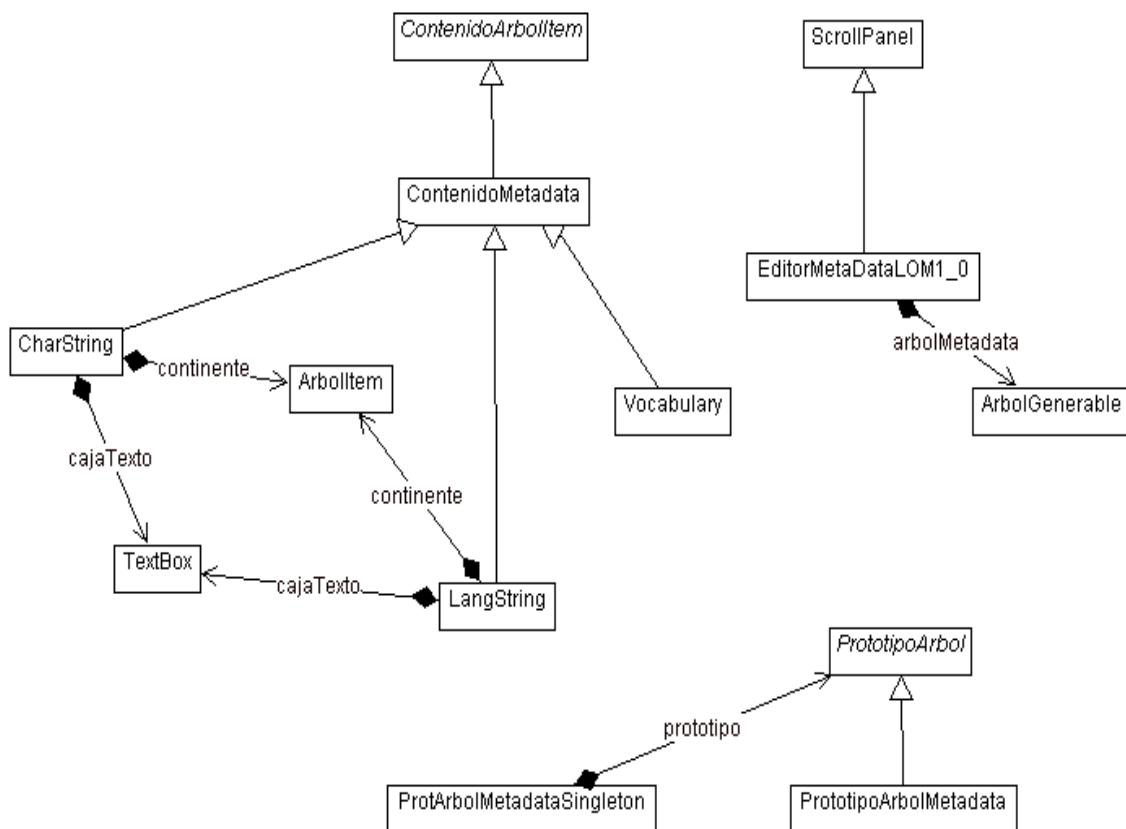


Diagrama de clases del paquete metadataLOM

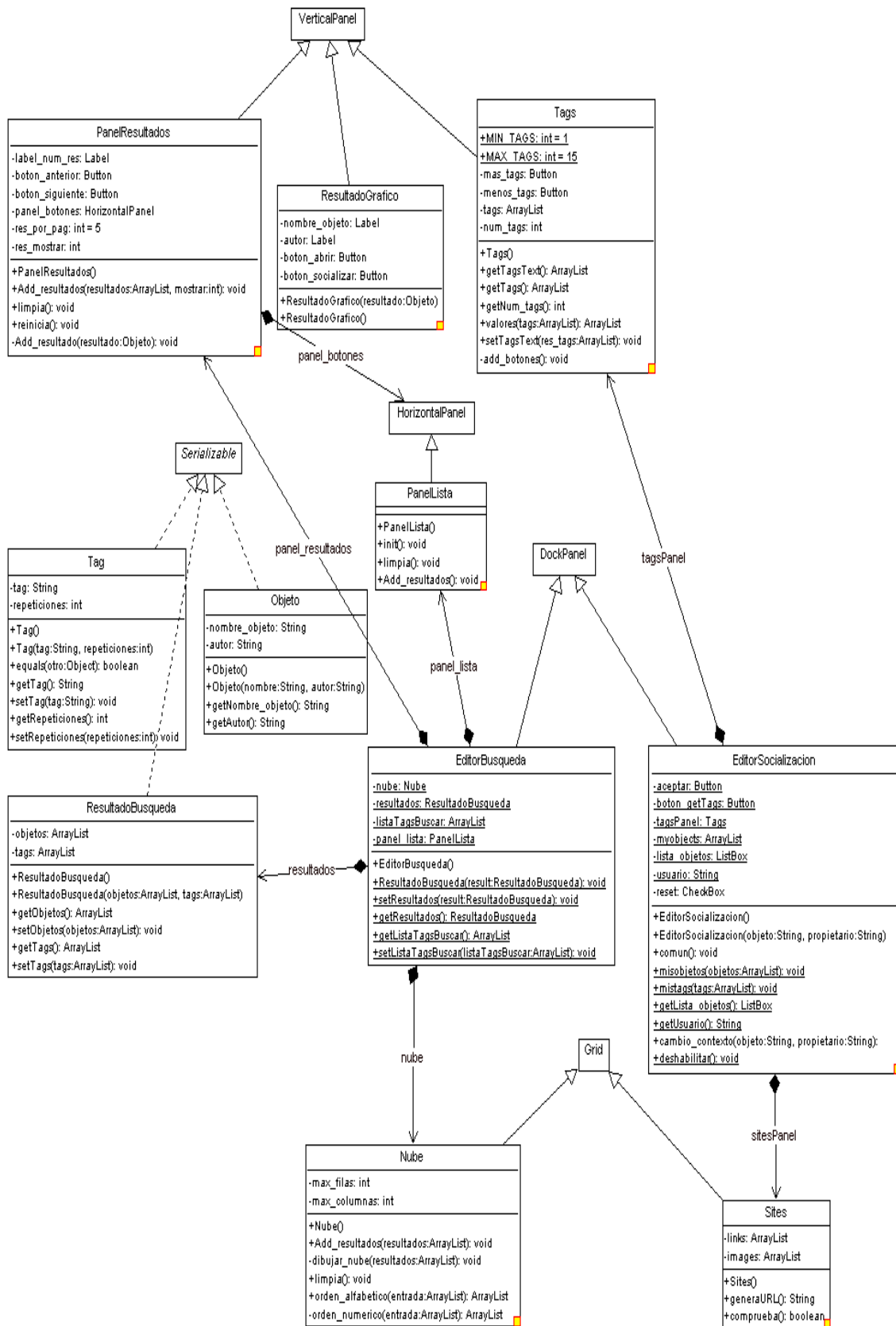


Diagrama de clases del paquete socialización

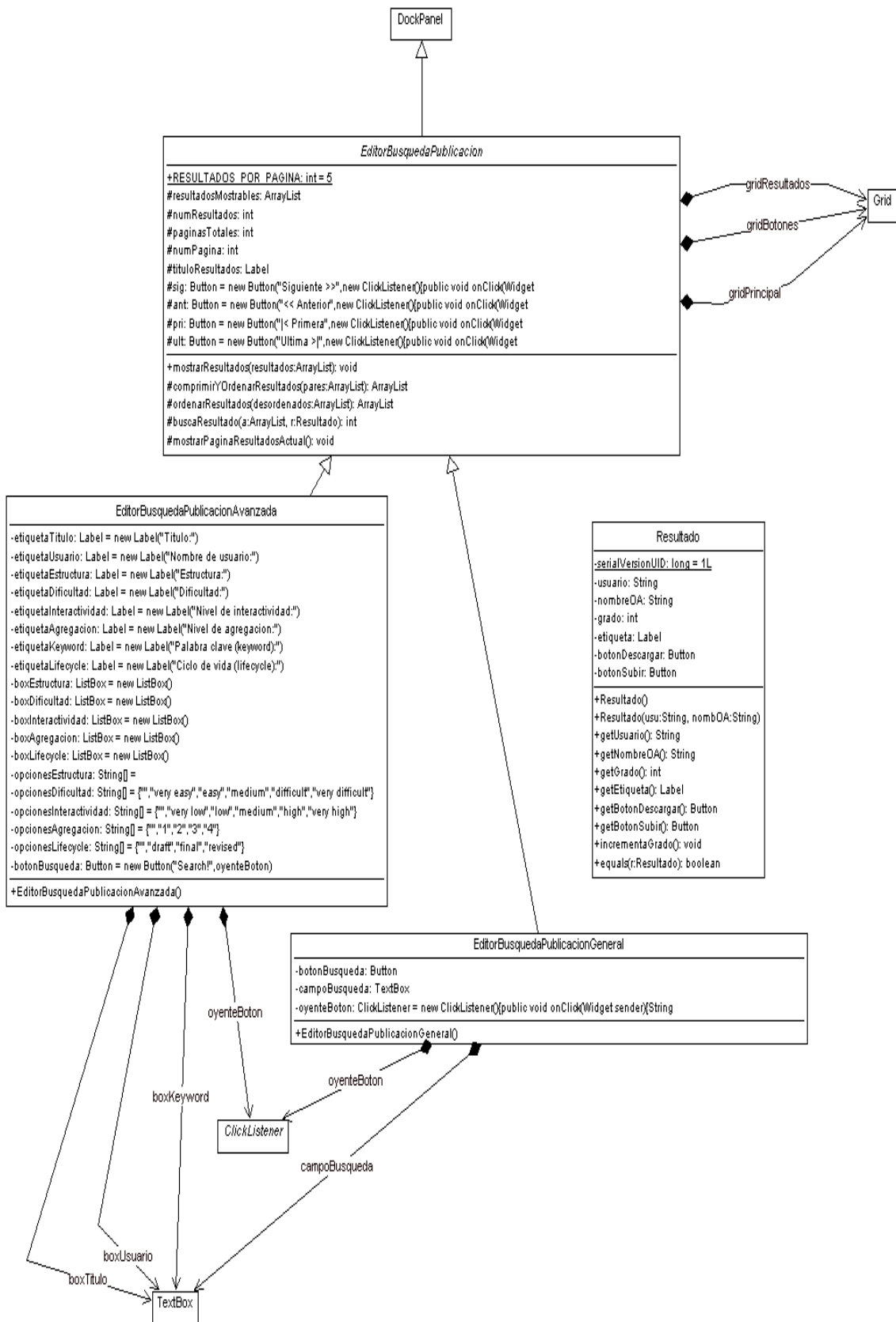


Diagrama de clases del paquete publicación

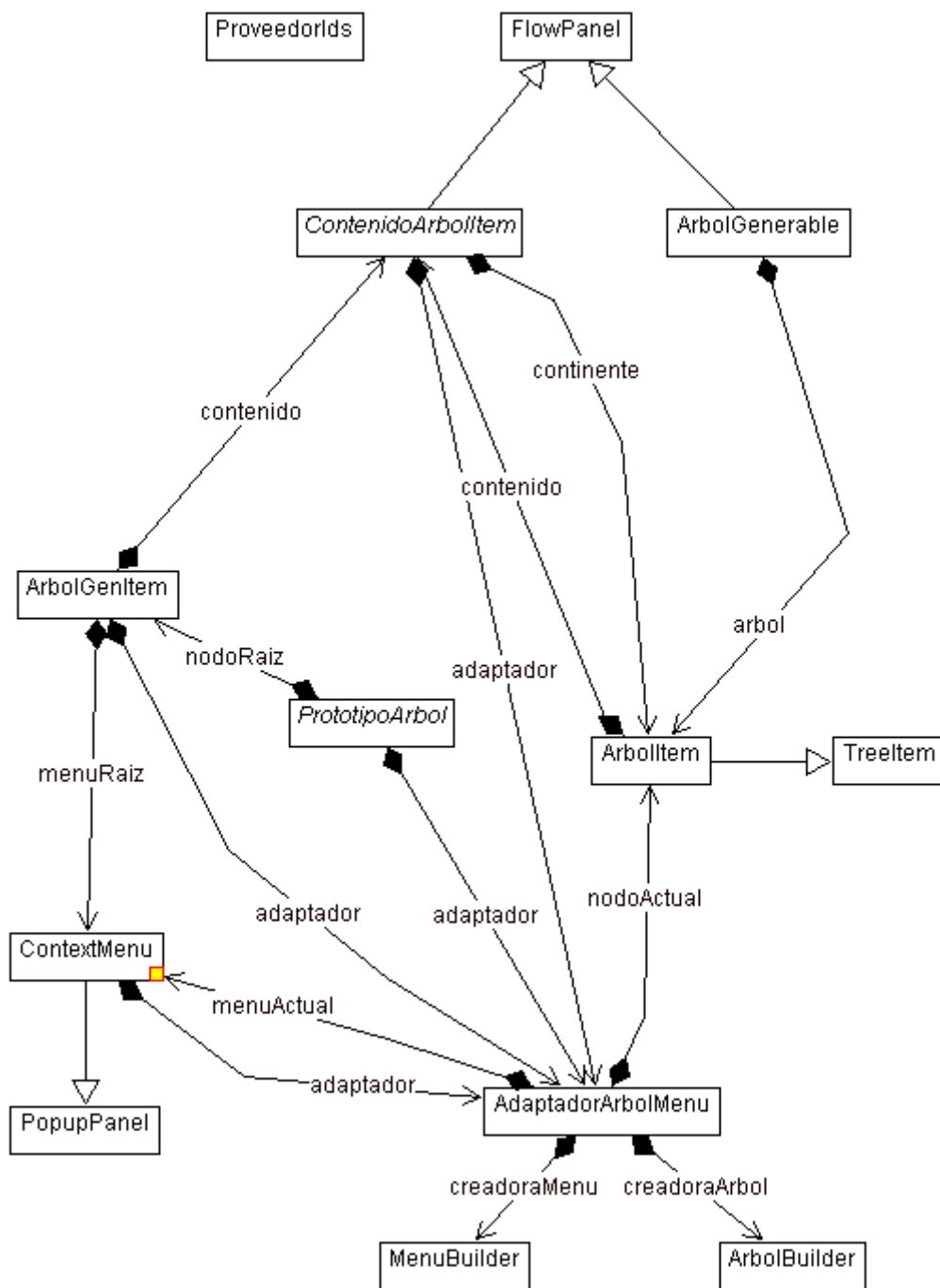


Diagrama de clases del paquete arbol

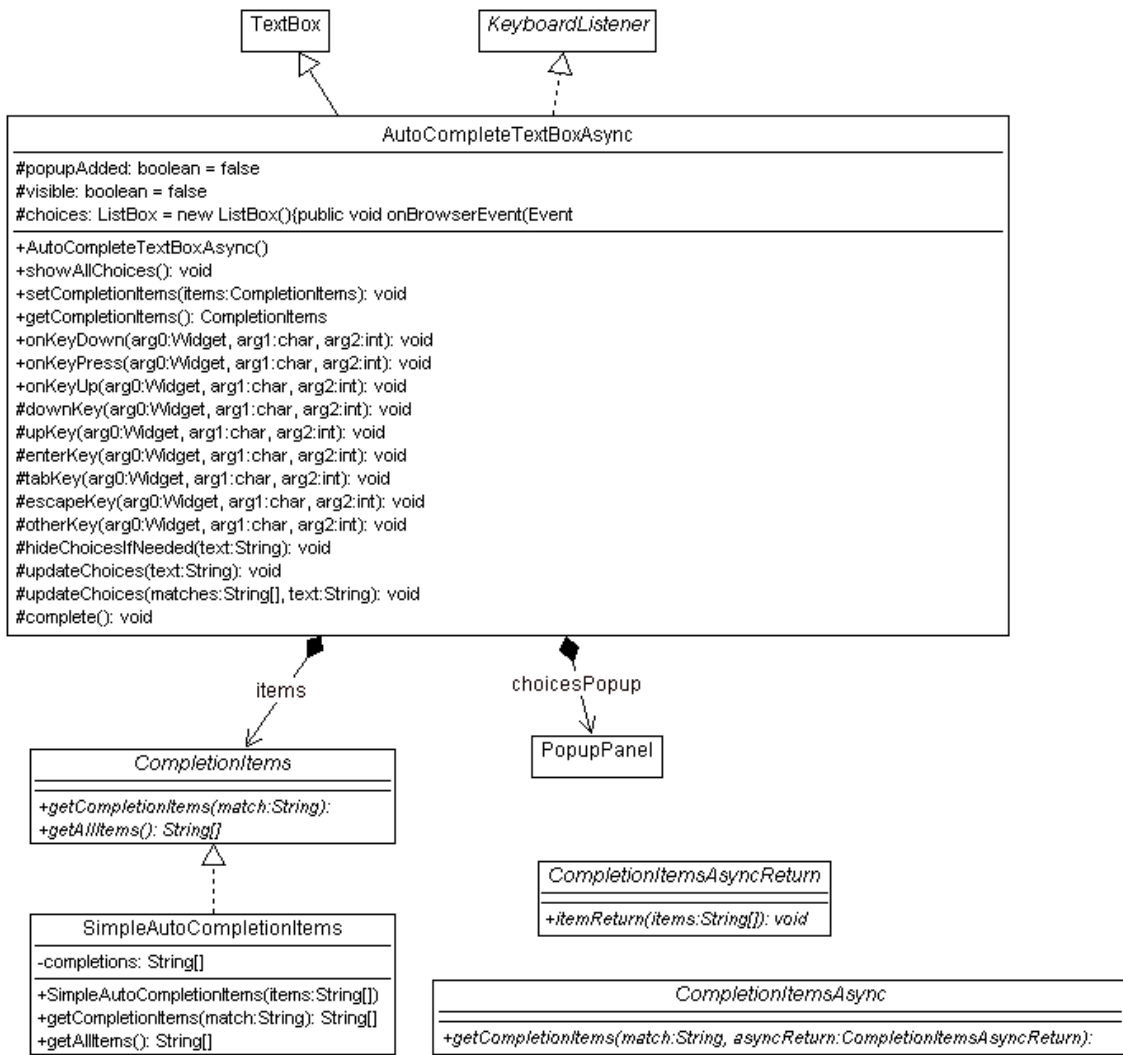


Diagrama de clases del paquete misWidgets

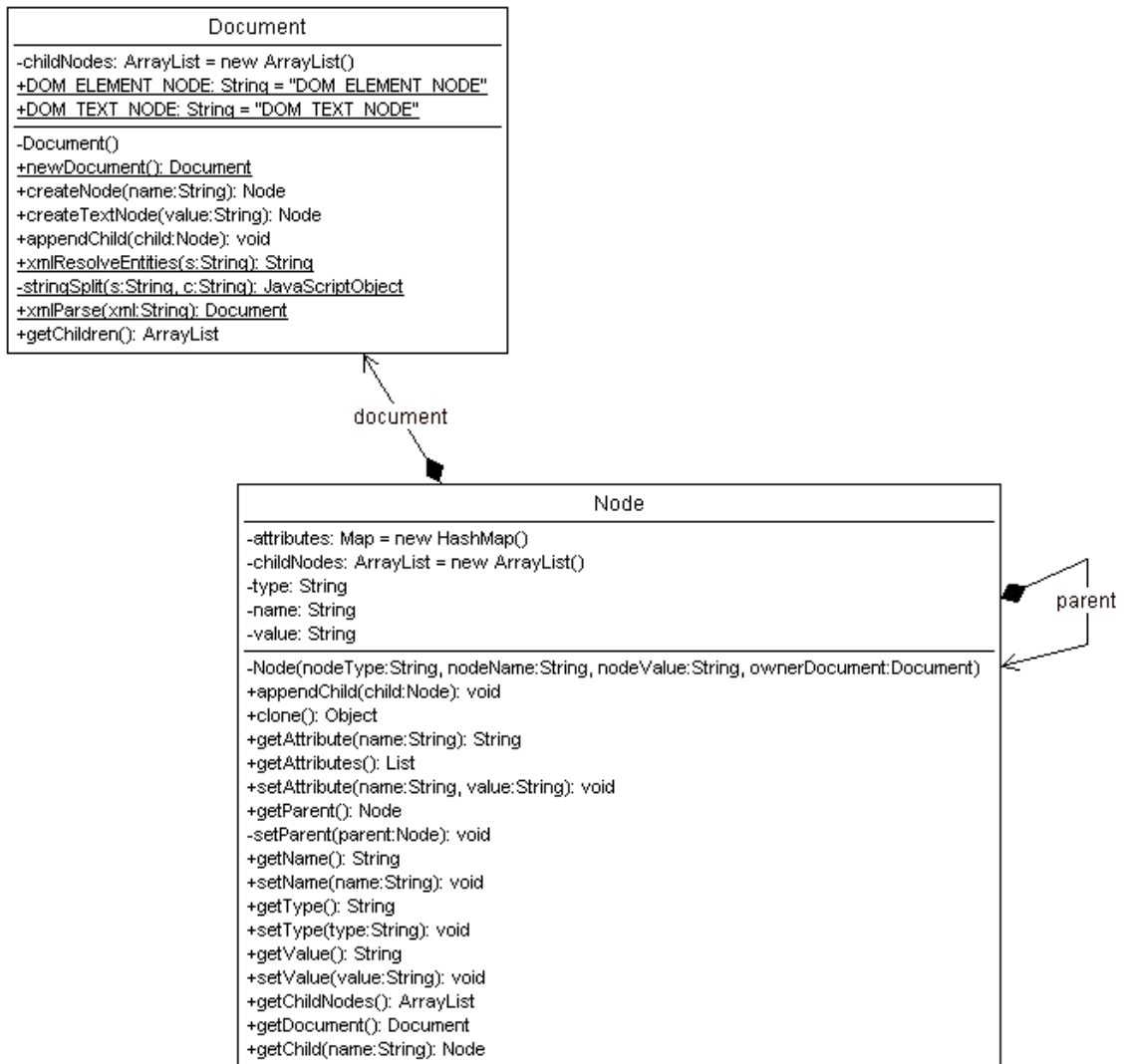
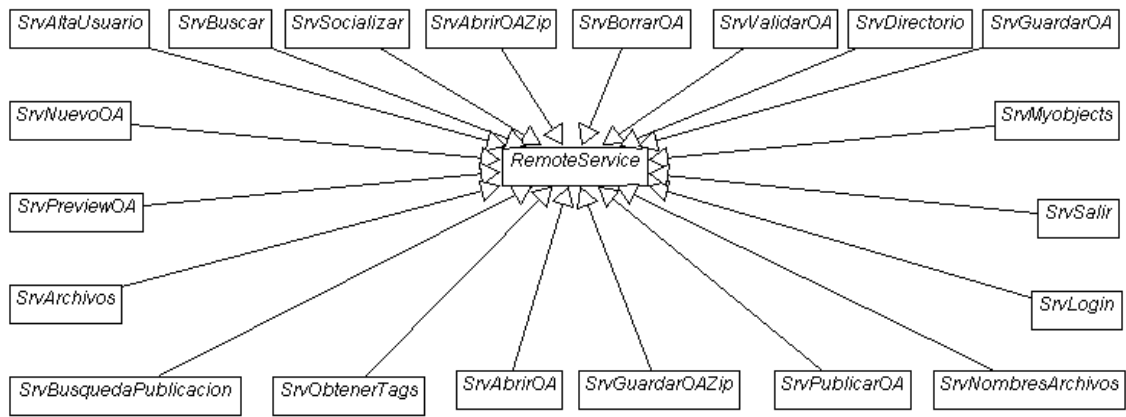


Diagrama de clases del paquete xmlDOM



Servicios



Diagrama de clases del paquete Servicios

Servidor

El servidor únicamente cuenta con la implementación de los distintos servicios que puede solicitar el cliente y de un paquete de clases auxiliares. Cada uno de los servicios empieza siempre por las letras “Srv” seguido del nombre del servicio en cuestión. Todos y cada uno de los servicios del servidor implementan el interfaz correspondiente del paquete Servicios del cliente, como se puede observar en el diagrama. En este caso, los servicios del servidor son RemoteServiceServlet.

A continuación se muestran los dos diagramas de clase que definen el contenido del servidor

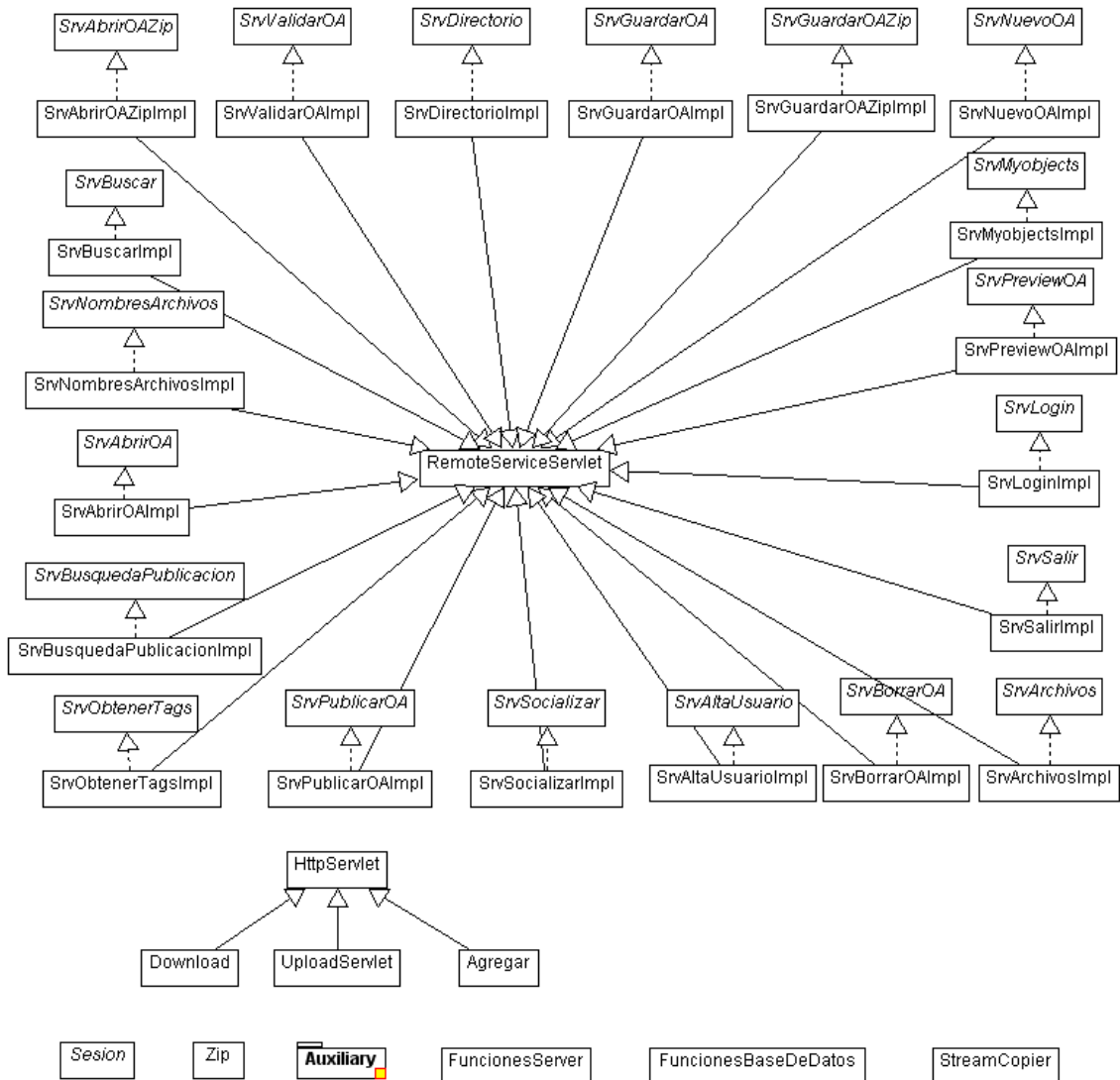


Diagrama general de clases del servidor

ArchivosMySQLDto
-usuario: String -size: String -nombreArchivo: String -nombreContentPreview: String -tamArchivo: float -archivoBinario: byte[]
+ArchivosMySQLDto() +getUsuario(): String +setUsuario(usuario:String): void +getSize(): String +setSize(size:String): void +getNombreArchivo(): String +setNombreArchivo(nombreArchivo:String): void +getArchivoBinario(): byte[] +setArchivoBinario(archivoBinario:byte[]): void +getTamArchivo(): float +setTamArchivo(tamArchivo:float): void +getNombreContentPreview(): String +setNombreContentPreview(nombreContentPreview:String): void

UtilidadesFicheros
-MAX_FILE_SIZE: int = 100000000
+readTextFileXML(archivo:String): String +readTextFile(fileToRead:File): String +leerArchivoTxt(fileToRead:File): String +readBinaryFile(fileToRead:File): byte[] +writeTextFile(fileToWrite:File, contenido:String): void +writeBinaryFile(fileToWrite:File, contenido:byte[]): void +deleteFile(fileToDelete:File): void

IdentificadorAleatorioServer
+randomID(): String

Archivo
+copyAllFiles(sc:ServletContext, baseUrl:String, dirIn:File, dirOut:File): void +copyFile(file in :File out, buildPath:boolean): void +deleteDirectory(path:File): boolean +deleteContent(dirPath:File): void +guardarXML(rutaArchivo:String, xml:String): void

Diagrama de clases del paquete Auxiliary

TECNOLOGÍAS

Google Web Toolkit (GWT)

Google Web Toolkit (GWT) es un framework de desarrollo en Java de código abierto, que te permite escapar de la “matriz” de tecnologías usadas actualmente para escribir aplicaciones AJAX, las cuales son difíciles de manejar y propensas a errores. Con GWT, puedes desarrollar y depurar aplicaciones AJAX usando el lenguaje de programación Java en el entorno de desarrollo de tu preferencia (me refiero al sistema operativo y a los IDEs). Cuando has acabado tu aplicación (que la has escrito en Java), GWT compila y traduce dicho programa a JavaScript y HTML compatible con cualquier navegador web.

Éste es el ciclo de Desarrollo de GWT:

1. Usa tu entorno de desarrollo integrado (IDE) favorito para escribir y depurar una aplicación en Java, usando las librerías GWT que necesites.
2. Usa el compilador de Java a JavaScript de GWT para transformar tu aplicación en un conjunto de archivos JavaScript y HTML que puedes colgar en cualquier servidor y ejecutar desde un navegador web.
3. Verifica que tus aplicaciones trabajan sobre todos y cada uno de los navegadores que consideres que tus clientes usarán.

Características de Google Web Toolkit

Componentes de la interfaz de usuario dinámicos y re-utilizables. Crea un Widget para construir otros. Coloca los Widgets automáticamente en Paneles. Se pueden enviar tus propios Widgets a otros desarrolladores en archivos JAR.

RPC

Para comunicarte desde el navegador que lanza tu aplicación con tu servidor web, solamente necesitas definir clases de Java serializables para las peticiones y respuestas. En producción, GWT serializa automáticamente las peticiones del navegador y de-serializa las respuestas desde el servidor web. El mecanismo de RPC de GWT puede incluso manejar jerarquía de polimorfismo en clases, y puedes manejar las posibles excepciones.

Administración del historial del navegador

Las aplicaciones en AJAX no necesitan utilizar el botón “atrás” (back) del navegador. Y GWT no es la excepción, es decir, no es necesario que llames a otras páginas para realizar las diferentes acciones, ni recargar el navegador ni nada.

Depuración en tiempo real

Para cuando tu aplicación esté lista, el código de la misma es traducido a JavaScript, pero mientras lo estás desarrollando este corre sobre una Java virtual machina (JVM). Lo que significa que en la fase de Desarrollo tienes la posibilidad de depurar tu aplicación con los avanzados sistemas de debugging y manipulación de excepciones incluidos en IDEs como Eclipse.

Compatibilidad con los navegadores

Tus aplicaciones en GWT serán automáticamente soportadas por navegadores como FireFox, Internet Explorer, Mozilla, Safari, y Opera sin ningún tipo de operación para la detección de los mismos, en la mayoría de los casos.

Integración con Junit

Mediante la integración de JUnit en GWT tu puedes probar tus aplicaciones y depurarlas en un navegador mientras las construyes, incluso, puedes testear llamadas asíncronas a procedimientos remotos RPC.

GWT es un proyecto de código abierto

Todo el código de GWT está disponible bajo la licencia Apache 2.0.

¿Porqué traducir código Java a JavaScript?

La tecnología Java ofrece una plataforma de desarrollo productiva, y con GWT, se puede convertir en una plataforma sólida para el desarrollo de tus aplicaciones AJAX. Aquí están algunos de los beneficios de desarrollar con GWT:

- Puedes usar cualquiera de tus IDEs favoritos (Eclipse, IntelliJ, JProfiler, JUnit).
- Los errores comunes en JavaScript (errores de sintaxis, por ejemplo) son fácilmente detectados mientras desarrollas la aplicación, y no cuando el usuario final lo esté ejecutando.
- El “refactoring” automático en Java está muy de moda en estos días.
- Los diseños en Java basados en la programación orientada a objetos es fácil de comunicar y entender, por lo que hace la base de tu código AJAX más comprensible con menos documentación

Usando Google Web Toolkit

En GWT puedes usar componentes de interfaz de usuario llamados Widgets, para construir aplicaciones AJAX con GUIs atractivas. Al igual que en la mayoría de los lenguajes de programación, los componentes de la UI son agrupados en paneles que determinan la ubicación de los mismos.

GWT soporta una gran cantidad de widgets que son útiles en el desarrollo de aplicaciones AJAX, incluyendo árboles, pestañas, barras de menú y menús de diálogo. GWT también soporta invocación de métodos remotos (RPC) y otras características.

Depuración y desarrollo de aplicaciones en GWT

Las aplicaciones GWT pueden ser ejecutadas en dos modos:

- *Modo hosted (hosted mode)* – En modo hosted, tu aplicación corre como bytecodes de Java sobre una máquina virtual. Por lo general gastarás más tiempo desarrollando en modo hosted, ya que allí cuentas con todas las ventajas que te proporciona Java para depurar usando un IDE como Eclipse.
- *Modo Web (Web mode)* – En modo web, tu aplicación corre como HTML + JavaScript sobre un navegador, traducido desde tu código fuente Java original con el compilador de GWT (Java-to-JavaScript compiler). Cuando tu aplicación está terminada, lo único que debes hacer es subirla a un servidor web, y los usuarios finales accederán a ella a través de un navegador en “modo web”.

Para soportar el modo hosted, GWT cuenta con un navegador especial que está “embebido” a la máquina virtual de Java.

La arquitectura de Google Web Toolkit

GWT tiene cuatro componentes principales: un compilador Java-a-JavaScript, un navegador web "hosted", y dos librerías de clases:



Los componentes son:

Compilador GWT Java-a-JavaScript

El Compilador GWT Java-a-JavaScript traduce del lenguaje de programación Java a JavaScript. El compilador se utiliza cuando necesites correr tu aplicación en modo web.

Navegador web "Hosted" de GWT

El Navegador web "Hosted" de GWT te permite correr y ejecutar GWT aplicaciones en modo hosted, donde lo que estás corriendo son bytecodes de Java sobre una máquina virtual sin compilarlos a JavaScript. Para lograr esto, el navegador GWT incrusta un controlador de browser especial (un control del Internet Explorer sobre Windows o un control de Gecko/Mozilla sobre Linux) con hooks dentro de la máquina virtual de Java.

Emulación de librerías JRE

GWT contiene implementaciones en JavaScript de las librerías de clases más usadas en Java, incluyendo la mayoría de las clases del paquete `java.lang` y un subconjunto de clases del paquete `java.util`. El resto del estándar de librerías de Java no es soportado nativamente con GWT. Por ejemplo, las clases de los paquetes como `java.io` no se utilizan en aplicaciones web ya que estas acceden a recursos en la red y al sistema de archivos local.

Librería de clases de interfaz de usuario de GWT

Las librerías de clases de interfaz de usuario de GWT son un conjunto de interfaces y clases personalizadas que te permiten crear "widgets" para el navegador, como botones, cajas de texto, imágenes, y texto. Éste es el núcleo de las librerías de interfaz de usuario para crear aplicaciones GWT.

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Características de la version 5.0.22

- Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Soporte a multiplataforma
- Procedimientos almacenados
- Triggers
- Cursores
- Vistas actualizables
- Soporte a VARCHAR
- INFORMATION_SCHEMA
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial)
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación(savepoints) con InnoDB
- Soporte para SSL
- Query caching
- Sub-SELECTs (o SELECTs anidados)
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para multiples maestros por esclavo.
- indexing y buscando campos de texto completos usando el motor de almacenamiento MyISAM
- Embedded database library
- Soporte completo para Unicode

- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster
- Shared-nothing clustering through MySQL Cluster

Características adicionales

- Usa GNU Automake, Autoconf, y Libtool para portabilidad
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL
- Completo soporte para operadores y funciones en cláusulas select y where.
- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).
- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.
- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.

MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

Tomcat

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Entorno

Tomcat es un servidor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios

Características del producto

Tomcat 3.x (distribución inicial)

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets
- Funciones básicas HTTP

Tomcat 4.x

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina
- Motor JSP rediseñado con Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

Tomcat 5.x

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Análisis rápido JSP

Tomcat 6.x

- Implementado de Servlet 2.5 y JSP 2.1
- Soporte para Unified Expression Language 2.1
- Diseñado para funcionar en Java SE 5.0 y posteriores
- Soporte para Comet a través de la interfaz CometProcessor

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

Apache-Ant

Apache Ant es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (build). Es similar a Make pero sin las engorrosas dependencias del sistema operativo.

Esta herramienta, hecha en Java, tiene la ventaja de no depender de las órdenes de shell de cada sistema operativo, sino que se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas, siendo idónea como solución multi-plataforma.

Ant es un proyecto de código abierto de la Apache Software Foundation.

XmlBeans

XMLBeans es una tecnología para procesar XML construyendo sus clases Java correspondientes con los tipos que poseen en los archivos xsd.

Atraves de unos comandos que posee la herramienta se puede validar un xml atraves de sus xsd o crear un jar que tenga la capacidad de validar los xml para poder integrarlo en una aplicación.

Aparte de ser una herramienta que permite validar los XML también permite hacer más cosas a través de otros comandos, aquí muestro alguno de los comandos que posee la herramienta:

svalidate (Streaming Instance Validator)

Valida la definición de un esquema y las instancias dentro de un esquema.

validate (Instance Validator)

Valida una instancia de XML contra un esquema.

scomp (Schema Compiler)

Compila un esquema en clases y metadatos propios de XMLBeans y genera las clases java correspondientes a los tipos del esquema.

GESTION DE CONFIGURACIÓN

En lo relativo a la gestión de configuración, el proyecto se divide en dos grandes áreas bien diferenciadas, para las que se han empleado distintas técnicas:

- El código fuente.
- La memoria.

Código fuente

Basandonos en la buena experiencia pasada en el proyecto de Ingeniería del Software, decidimos emplear un servidor CVS para controlar las versiones del código. Este sistema presenta una serie de ventajas importantes tales como:

- Tener acceso en cualquier momento y lugar con una conexión a internet a la última versión del código fuente
 - Trabajar sin preocuparse del código que pueda estar modificando otra persona.
 - Posibilidad de volver a versiones anteriores de código en caso de necesidad
- etc

Podría haberse empleado un servidor SVN que es tecnología mas actual que el CVS, pero dado que conocíamos bien CVS por haber trabajado con él durante un año, y la gran integración en la plataforma Eclipse, decidimos usar CVS. Además para las necesidades del proyecto era suficiente.

Concretamente se han empleado los servicios de <http://sourceforge.net/>

Memoria

Para editar la memoria se ha empleado un sistema colaborativo online, concretamente Google Docs. Esto nos ha permitido, al igual que el CVS, disponer siempre de la última versión de la memoria y editarla online con la comodidad que conlleva, sin necesidad de más que un navegador de internet.

<http://docs.google.com/>

Reflexiones finales

El uso de las herramientas de gestión de configuración anteriormente expuestas han sido esenciales para el trabajo en grupo y han aportado una gran flexibilidad y comodidad a la hora de trabajar. Sin duda la experiencia previa en el uso de estas herramientas ha condicionado la elección y nos ha ahorrado períodos de adaptación y aprendizaje.

CONCLUSIONES

La contribución fundamental de este proyecto ha sido la construcción de una aplicación capaz de realizar prácticamente todas las funcionalidades que nos propusimos al empezar. Es una aplicación que aporta cosas muy novedosas en su campo como el validador integrado, el hecho de funcionar en web, gestión de usuarios y almacenamiento, características de la web 2.0 etc. Pensamos que todo esto hará del uso de la aplicación una experiencia más completa que el uso de aplicaciones similares, y que resultará muy útil y cómoda a sus usuarios.

En lo referente a nuestra formación, a lo largo del curso hemos adquirido experiencia en tecnologías desconocidas para nosotros como:

- Bases de datos, MySQL.
- Aplicaciones web, Google Web Toolkit.
- Arquitectura cliente-servidor con llamadas RPC.

Además hemos puesto en práctica y perfeccionado procedimientos adquiridos en la asignatura de Ingeniería del Software. Pensamos que ambas cosas pueden tener un gran peso en nuestro futuro profesional.

La aplicación generada se presta además a futuras ampliaciones y mejoras, que si bien pensamos, ha sido imposible realizarlas por falta de tiempo, como:

1. Secuenciamiento de navegación, a través del cual se describe el orden de la presentación de los contenidos según la navegación hecha por el usuario. Con este propósito se definirían los llamados árboles de actividades, que definirían las posibles ordenaciones según las acciones efectuadas por el usuario.
2. Un editor gráfico para acercar la creación de objetos de aprendizaje a personas sin conocimientos previos en la materia.
3. Otro editor gráfico para añadir secuenciación fácilmente a objetos de aprendizaje, ya que mediante reglas es bastante complicado.

MANUAL DE USUARIO

Arrancar la aplicación mediante el explorador de Internet

Para poder arrancar esta aplicación Web es necesario disponer de un explorador de Internet (Internet Explorer, Firefox, NetScape, Opera...). Una vez abierto el explorador introducimos lo siguiente en la barra de dirección, como se indica en la figura 1.



Figura 1: barra de dirección

Una vez introducida la dirección correctamente se mostrará un pequeño formulario (figura 2) para darse de alta por primera vez en la aplicación o para iniciar sesión en caso de ya disponer de alguna cuenta.

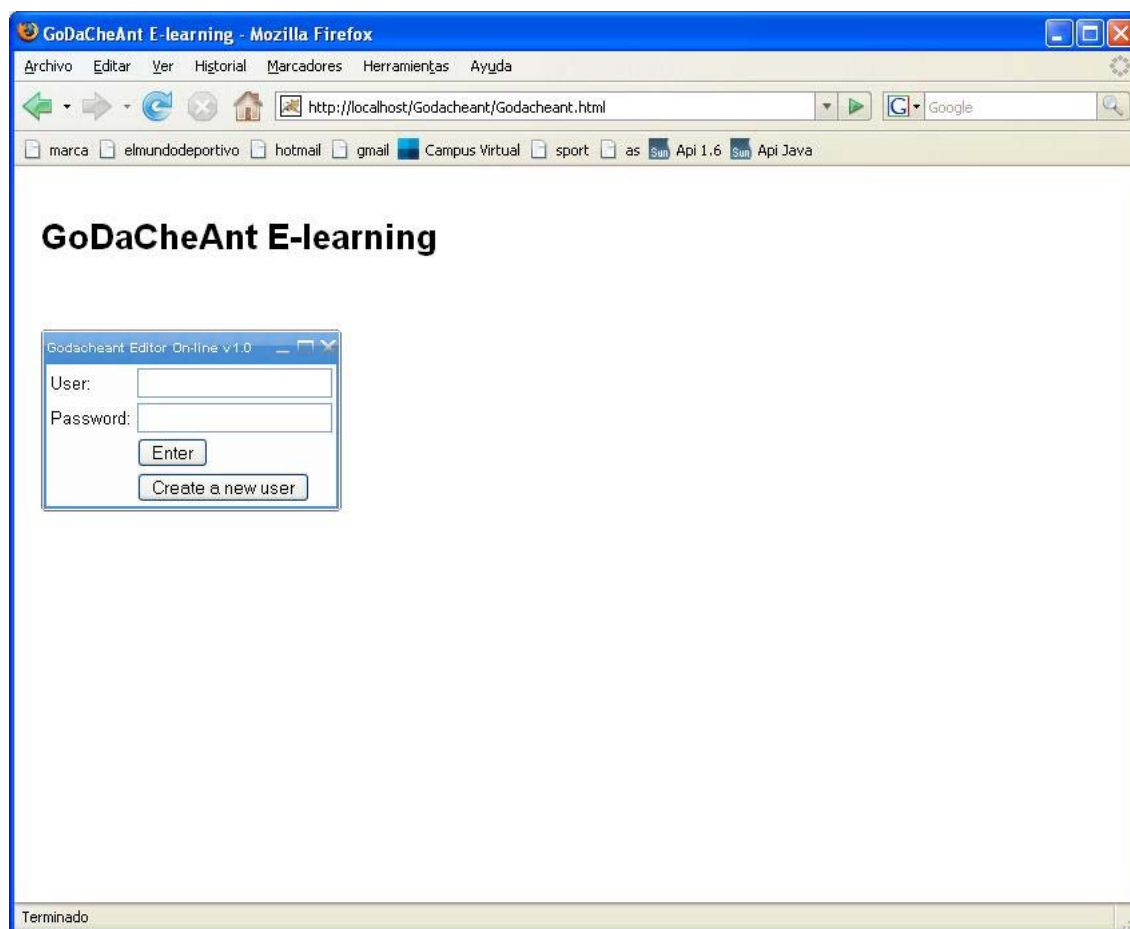


Figura 2: formulario para iniciar sesión o darse de alta

Darse de alta como usuario

Si es la primera vez que ejecutamos la aplicación será necesario darse de alta en el sistema. Para ello pinchamos en el botón “Create a new user” de la figura 2, tras lo cual se mostrará otro formulario distinto como el de la figura 3

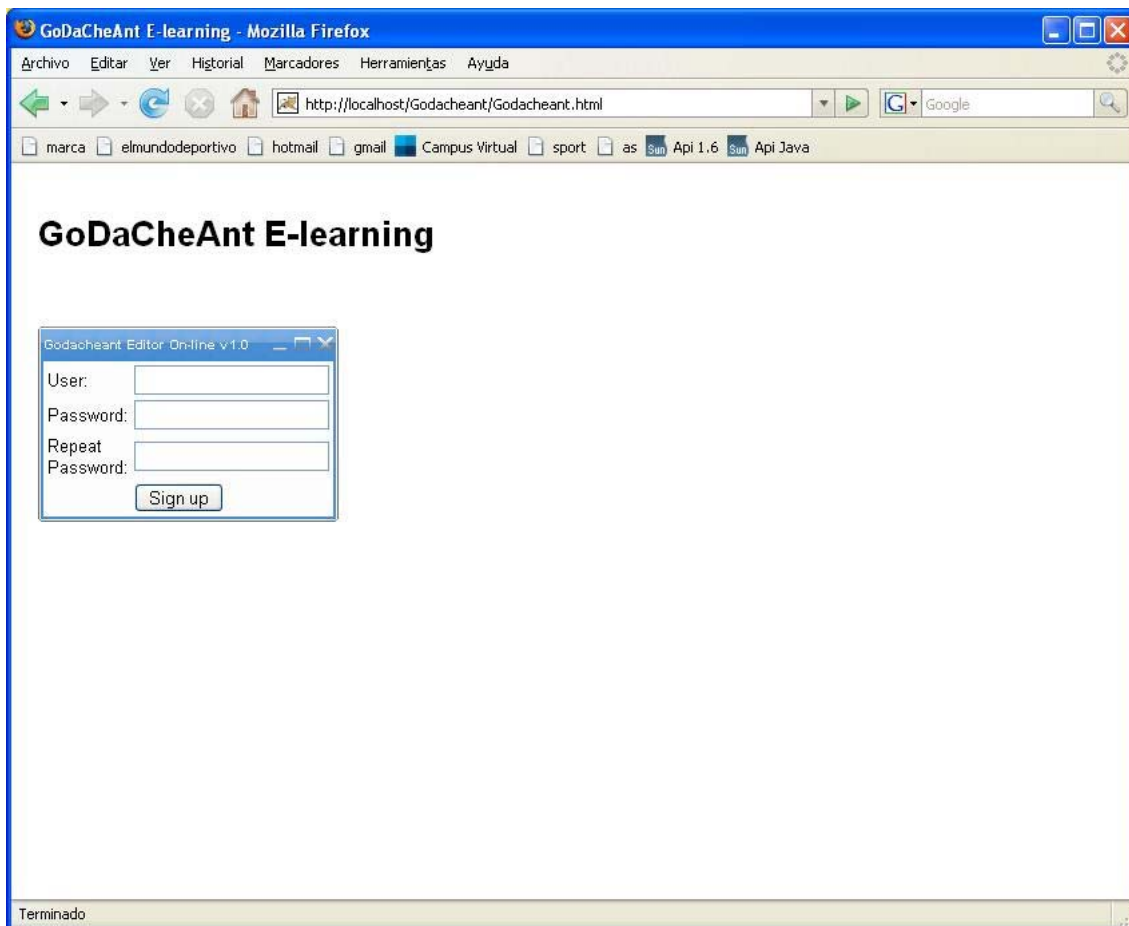


Figura 3: formulario de alta de nuevo usuario

Este formulario dispone de tres entradas, una para introducir el nombre de usuario y dos para confirmar la contraseña. Una vez introducidos pulsamos en el botón “Sign up”. Tanto el nombre de usuario como la contraseña deben ser de 1 o más caracteres. En caso contrario se mostrará un mensaje como el de la figura 4. Si el nombre de usuario introducido ya existe, de mostrará un mensaje como el de la figura 5 y deberemos elegir otro nombre de usuario. Si todo es correcto se volverá a mostrar el formulario inicial (figura 2) y se habrá creado una cuenta de usuario. A partir de este momento se podrá iniciar sesión.

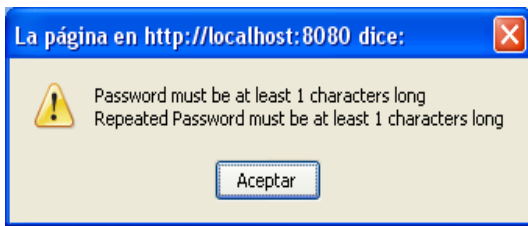


Figura 4: Error al introducir el password

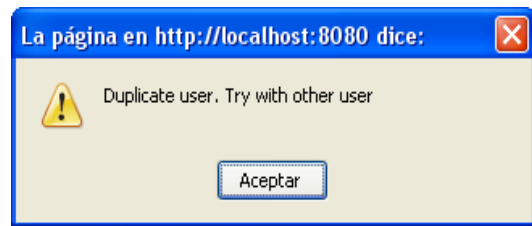


Figura 5: Error de usuario duplicado

Iniciar sesión

Es en este momento cuando debemos introducir el nombre de usuario y la contraseña para poder arrancar el editor. Una vez introducidos, pinchamos en el botón “Enter”, y tras unos segundos tendremos abierto en una ventana más grande nuestro editor (figura 6), listo para empezar a trabajar.

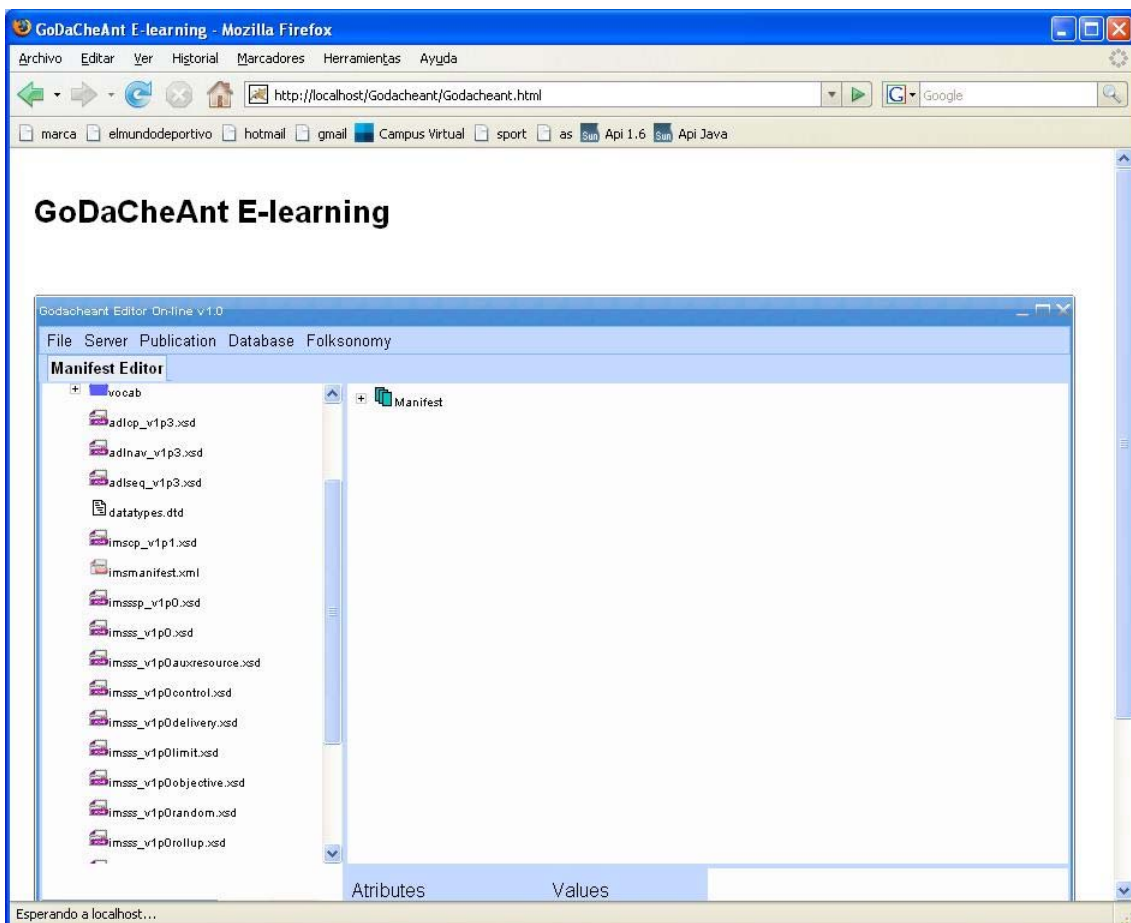


Figura 6: El editor al iniciar sesión

En caso de haber introducido un password incorrecto se mostrará un mensaje de error como el de la figura 7. En este caso deberemos volver a introducirlo correctamente.

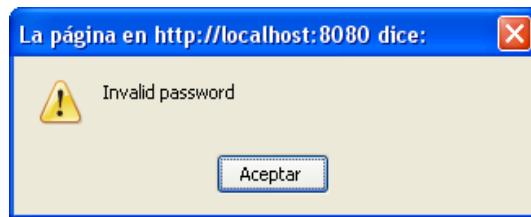


Figura 7: Error de password incorrecto

En caso de ser el nombre de usuario el campo que hemos introducido de forma incorrecta se mostrara un mensaje como el de la figura 8.

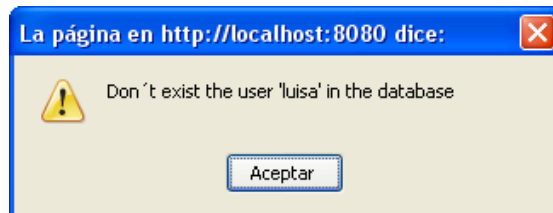


Figura 8: El usuario introducido no existe

Opciones del menú File

File es el menú principal (figura 9), y nos aporta todas las funcionalidades posibles para crear un nuevo objeto de aprendizaje, guardarlo, abrir su contenido, importar un objeto en formato zip, o exportarlo. También permite generar un content preview para el objeto que tenemos abierto y salir de la aplicación.

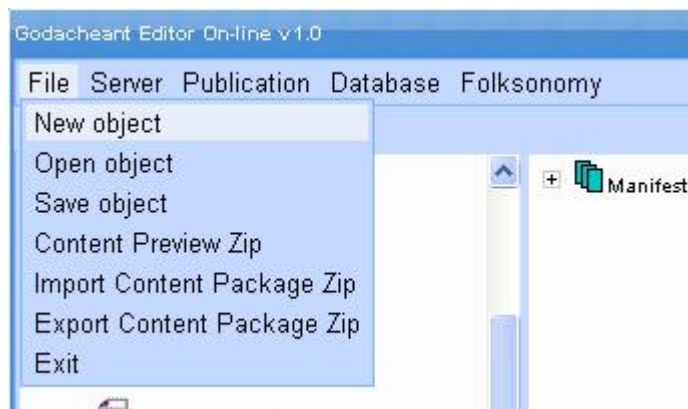


Figura 9: Menú File

Crear un objeto de aprendizaje

La primera de las opciones es crear un nuevo objeto de aprendizaje (“New object”). Siempre que ejecutemos este comando la aplicación nos preguntará si estamos seguros de eliminar el objeto que hay actualmente abierto en el editor, mostrando un mensaje como el siguiente

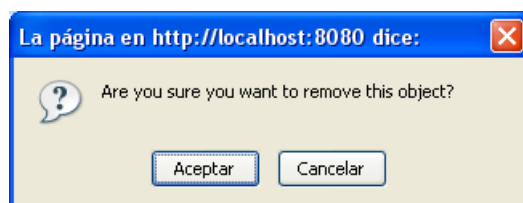


Figura 10: Confirmación de borrado del objeto actual

En caso de aceptar se eliminará todo el contenido del objeto que teníamos abierto y se creará uno nuevo con los elementos básicos ya creados, obteniendo un aspecto del editor igual que tenemos cuando iniciamos sesión (figura 6).

Si por el contrario decidimos cancelar, mantenemos intacto el objeto que tenemos abierto en el editor, pero por supuesto no se llega a crear un nuevo objeto.

Guardar un objeto de aprendizaje

Por supuesto además de crear objetos nuevos también podemos ir guardando nuestros avances en la construcción de un nuevo objeto (“Save object”). Cabe destacar siempre y cuando no se exporte el objeto que se esté creando, realizando una copia local en disco, no se obtendrá una copia real del objeto, ya si salimos de la aplicación si haber realizado una copia local del mismo, habremos perdido todo el trabajo. Todo esto es debido a que el objeto que estamos editando se está guardando en el servidor y solo podemos obtener una copia real si lo exportamos.

Abrir un objeto de aprendizaje

Con esta opción del menú principal simplemente visualizamos en el editor el contenido del manifiesto en la ventana central. Esto es útil en caso de haberlo cerrado sin querer o para visualizar los cambios que se han realizado después de haber guardado el objeto.

Importar un objeto de aprendizaje al editor

Esta funcionalidad consiste en abrir un objeto de aprendizaje, que tenemos guardado localmente en disco, en el editor. Para ello es necesario borrar el objeto de aprendizaje actual, dando a aceptar a la pregunta que nos hace (figura 10). Tras esto se nos muestra una pequeña ventana con tres botones: examinar, upload y ok.

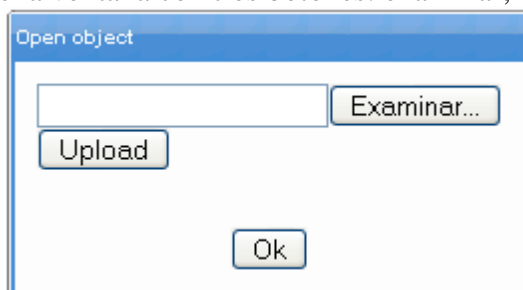


Figura 11: Ventana para importar un objeto

En primer lugar pinchamos en “examinar”, tras lo cual se abre un explorador de archivos de nuestro equipo. Una vez encontrado y seleccionado el objeto que queremos subir al editor, pinchamos en el botón “upload”. Tras un pequeño tiempo tenemos el objeto de aprendizaje abierto en nuestro editor. Para cerrar la ventana pinchamos en “ok”.

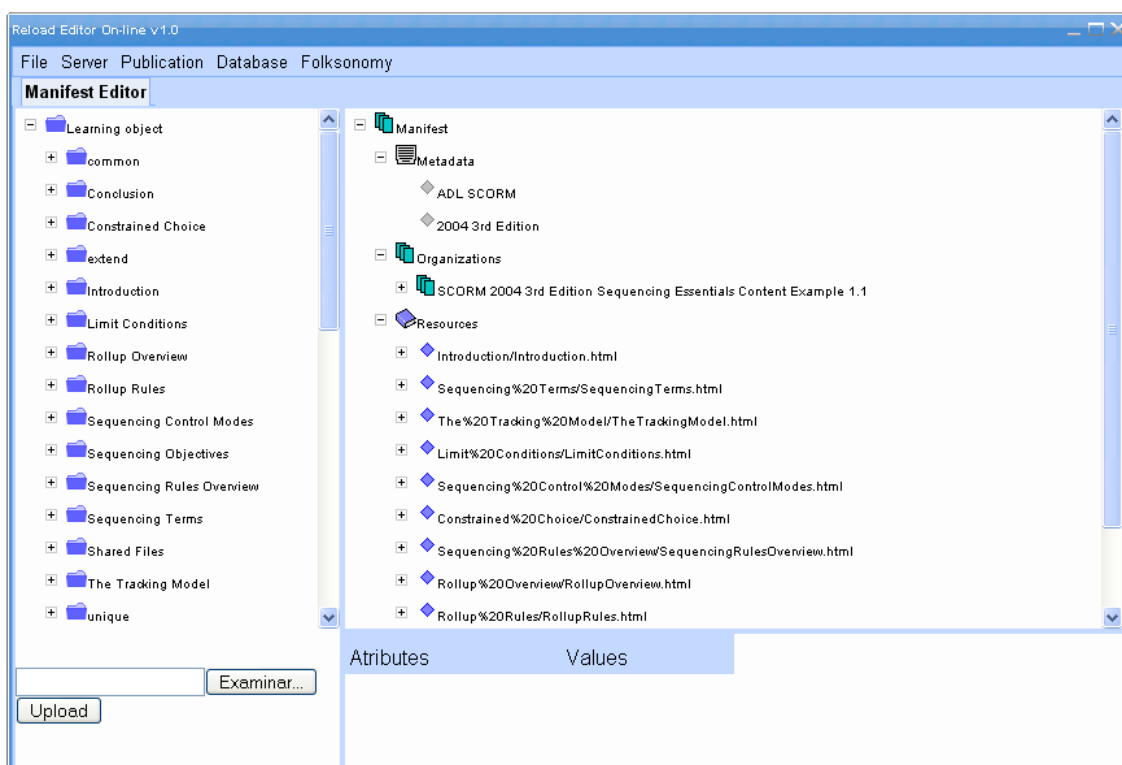


Figura 12: Vista del editor tras importar un objeto de aprendizaje

En caso de que el archivo seleccionado no sea un objeto de aprendizaje se mostrará un mensaje de error como el siguiente:

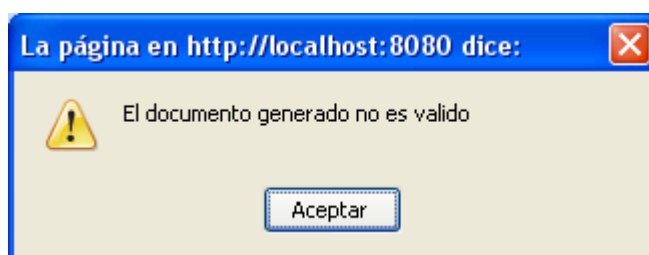


Figura 13: Mensaje de error por intentar subir algo que no es un OA

Exportar un objeto de aprendizaje desde el editor

Con esta opción del menú File podemos guardar una copia real del objeto de aprendizaje que estamos creando, o hemos creado ya. Esta función es muy importante ya que es el paso previo para poder almacenar objetos en nuestra carpeta personal del servidor.

Su aplicación es muy sencilla ya que vale con pulsar en la opción del menú “Export Content Package Zip” y en un instante se mostrará una pequeña ventana que nos pregunta si queremos guardar un archivo (figura 14). El archivo viene por defecto con un nombre que empieza por “pack” y seguido de una serie de números y letras. Esta riestra de caracteres es simplemente un identificador aleatorio que genera la aplicación para cada sesión. El formato de los objetos de aprendizaje es siempre formato ZIP o de archivo comprimido cuando se exporta o si importa al editor. Si pulsamos en guardar podremos cambiar el nombre del archivo y poner el que queramos y guardarlo localmente en el equipo.

A partir de este momento disponemos de una copia local del objeto de aprendizaje que se está editando, tal y como lo dejamos en el momento antes de exportarlo.

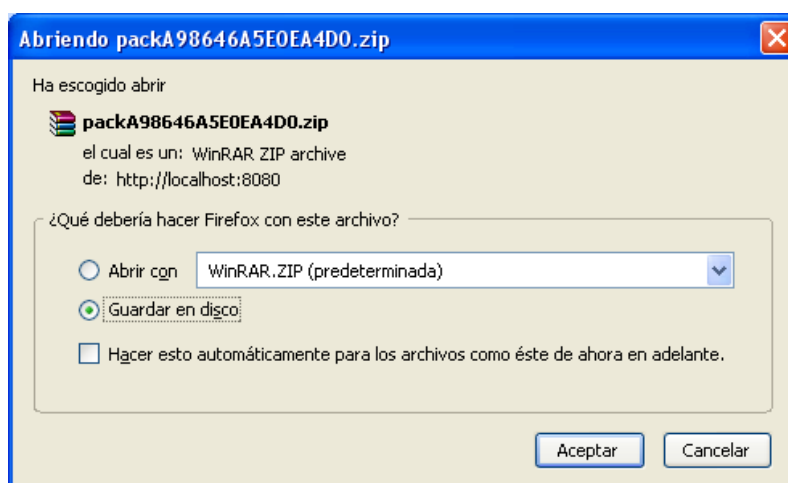


Figura 14: Ventana de descarga del archivo exportado

Generar un content preview para un objeto de aprendizaje

El content preview de un objeto de aprendizaje es el contenido del objeto de aprendizaje. Con él podemos ver todos y cada uno de los detalles que aporta en formato html, junto con todos los recursos de los que dispone. Por supuesto la organización del archivo será la que hemos definido en el momento de creación del objeto.

Por lo tanto con esta opción podemos generar un content preview para cualquier objeto de aprendizaje siempre y cuando lo tengamos abierto en el editor.

Si ejecutamos esta funcionalidad podremos ver que tras unos segundos se abre una ventana de descarga del archivo generado al igual que ocurre al exportar un OA. En este caso el nombre del archivo comienza por “prev” seguido del identificador de sesión anteriormente mencionado. De la misma forma que en el punto anterior si le damos a guardar, podremos elegir un nuevo nombre y destino.

Una vez descargado podemos abrir el archivo que acabamos de descargar y hacer doble clic en el archivo “ReloadContentPreview.htm”. Este es el índice del content preview y como vemos en la imagen podemos navegar por él y ver todo su contenido.

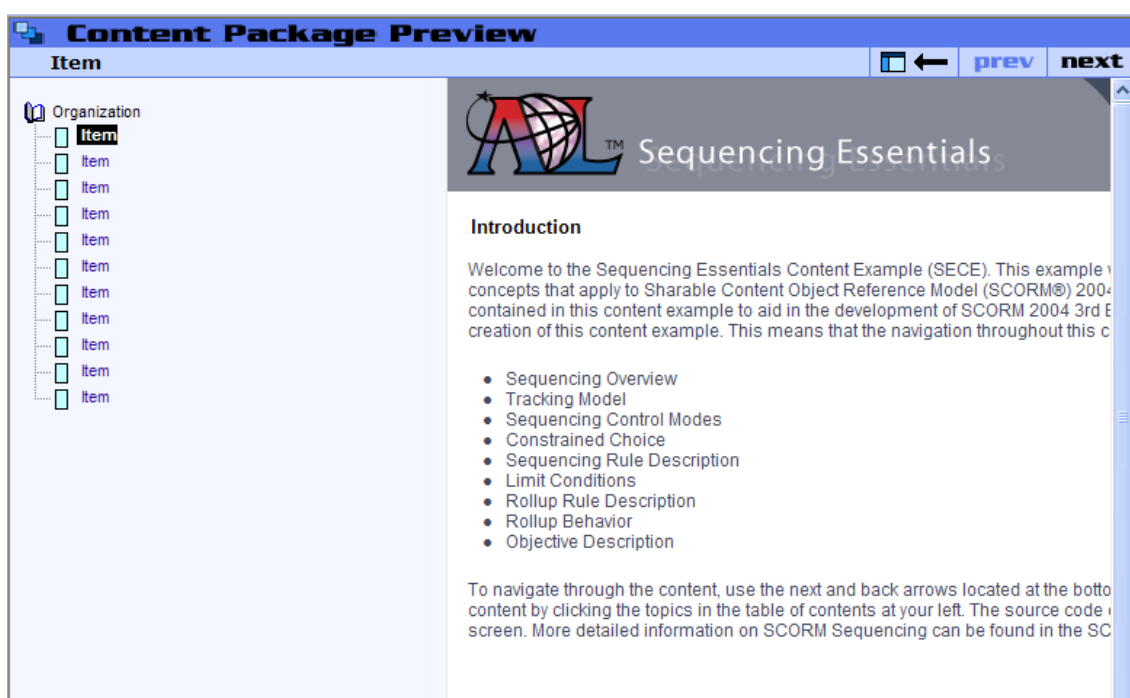


Figura 15: Ejemplo de content preview

Salir de la aplicación

Cuando queramos salir de la aplicación será necesario pinchar en la última de las opciones del menú principal. Una vez fuera se muestra una ventana de fin como la siguiente



Figura 16: pantalla de fin

Opciones del menú Server

Server es el menú necesario para manejar todo lo relacionado con la carpeta personal de objetos de aprendizaje que disponemos en el servidor. Estas opciones son, subir un nuevo objeto, eliminar un objeto existente e importar uno de los objetos almacenados al editor.

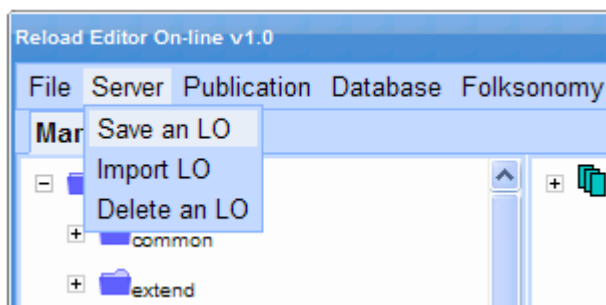


Figura 17: Contenido del menú server

Guardar un objeto de aprendizaje en la carpeta personal del usuario

Para poder guardar un objeto de aprendizaje en nuestra carpeta personal basta con pinchar en la opción “Save an LO” del menú server. Se nos mostrará una pequeña ventana igual que la que vimos en la figura 11, en el apartado de importar un OA. Los pasos para subir un objeto son los mismos que entonces. Pinchamos en el botón “examinar” para buscar y seleccionar el archivo. A continuación pinchamos en “upload” y si todo es correcto tendremos ya almacenado el objeto de aprendizaje en nuestro pequeño almacén privado. Para verificarlo pinchamos en “Delete an LO” o en “Import LO”. Con ambas opciones, que veremos a continuación, se listan todos los objetos que tenemos actualmente en nuestra carpeta personal de usuario (figura 18).

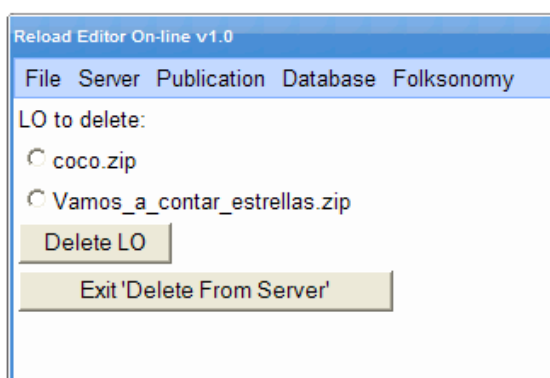


Figura 18: lista de objetos de la carpeta personal

En caso de que el sistema nos muestre un error como el de la figura 19, significará que el objeto de aprendizaje que estamos intentando subir al servidor no ha pasado el filtro del validador. La aplicación dispone de un validador de objetos de aprendizaje, basado en el SCORM 2004 extendido, que sólo permite subir objetos que

estén bien contruidos. En caso de no ser así nos mostrará un mensaje con la lista de errores.

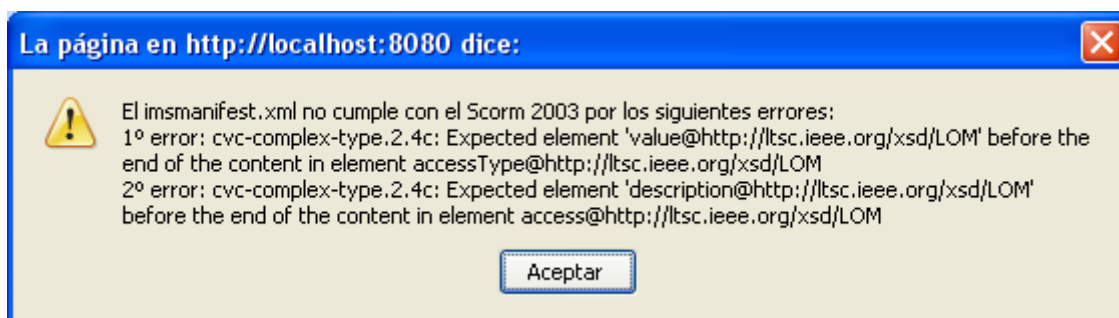


Figura 19: Lista de errores del validador al intentar subir un OA al servidor

Otra situación posible es cuando intentemos guardar un objeto de aprendizaje en nuestra carpeta personal y ya exista uno con el mismo nombre. En ese caso el sistema nos preguntará con un mensaje como el de la figura 20, si queremos sobrescribir el archivo o si cancelamos la subida del nuevo objeto. Si pinchamos en aceptar perderemos el objeto de aprendizaje que teníamos almacenado y se sustituirá por el que hemos seleccionado de disco. Si no aceptamos no cambiará nada y mantendremos nuestro pequeño almacén tal y como estaba.

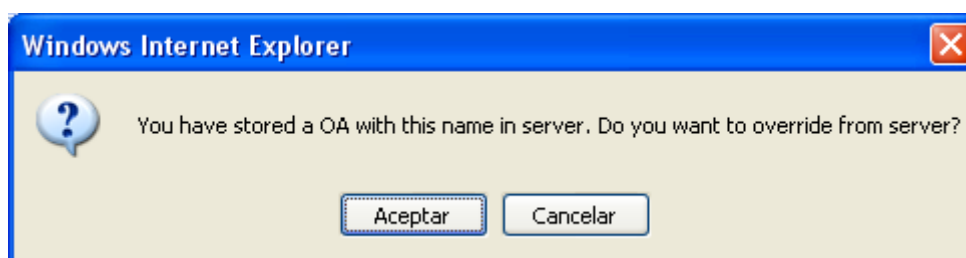


Figura 20: Mensaje preguntando si queremos sobrescribir un OA en el servidor

Eliminar un objeto de aprendizaje de la carpeta personal del usuario

Cuando queramos eliminar un objeto de aprendizaje de nuestra carpeta personal, bastará con seleccionar la opción “Delete an LO” del menú server. Se mostrará el contenido en forma de lista (figura 18). En este momento seleccionamos uno de los objetos de la lista y pinchamos en “Delete LO”. A partir de este momento dejamos de tener el objeto de aprendizaje en nuestra carpeta personal. Además del objeto de aprendizaje en si, también eliminamos todas las versiones que haya, del content preview del objeto, almacenadas en la base de datos, como se verá más adelante. En caso de ser público también se eliminarán todas los tags que tuviese (menú folksonomy).

Si decidimos que no queremos eliminar ningún objeto y queremos cerrar el listado de objetos, basta con pinchar en el botón titulado “Exit Delete from server” y dejará de ser visible.

Si pinchamos en el botón de borrado sin haber seleccionado ningún objeto el sistema nos mostrará un mensaje como el que sigue avisando de ello

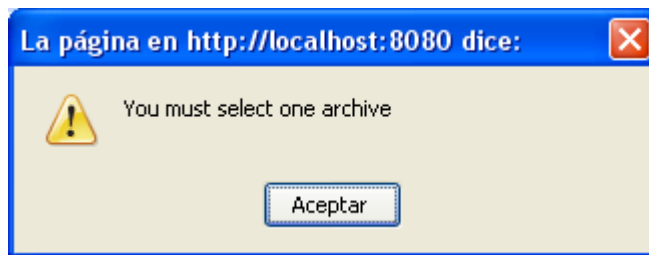


Figura 21: No se ha seleccionado ningún OA

Importar un objeto de aprendizaje de la carpeta personal del usuario al editor

Como última opción del menú Server, tenemos la posibilidad de abrir uno de los objetos que tenemos almacenados en nuestra carpeta privada en el editor. Cuando realicemos esta acción la aplicación nos dirá que si seguimos eliminaremos el objeto que actualmente tenemos en el editor (figura 10). Tras haber aceptado, se nos mostrará la lista de objetos personales. Los pasos son análogos al apartado anterior, donde eliminábamos un objeto. Sólo cambian los nombres de los botones. En este caso el botón "Load LO" sustituye a "Delete an LO" para importar el objeto seleccionado. Por lo tanto tras haber seleccionado el objeto, y haber pinchado en el botón "Load LO", tendremos abierto en el editor el contenido del objeto de aprendizaje.

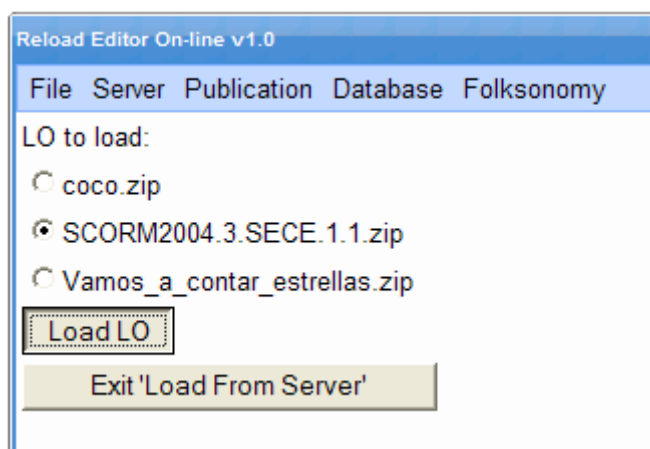


Figura 22: Lista de objetos de personales y objeto seleccionado para importar al editor

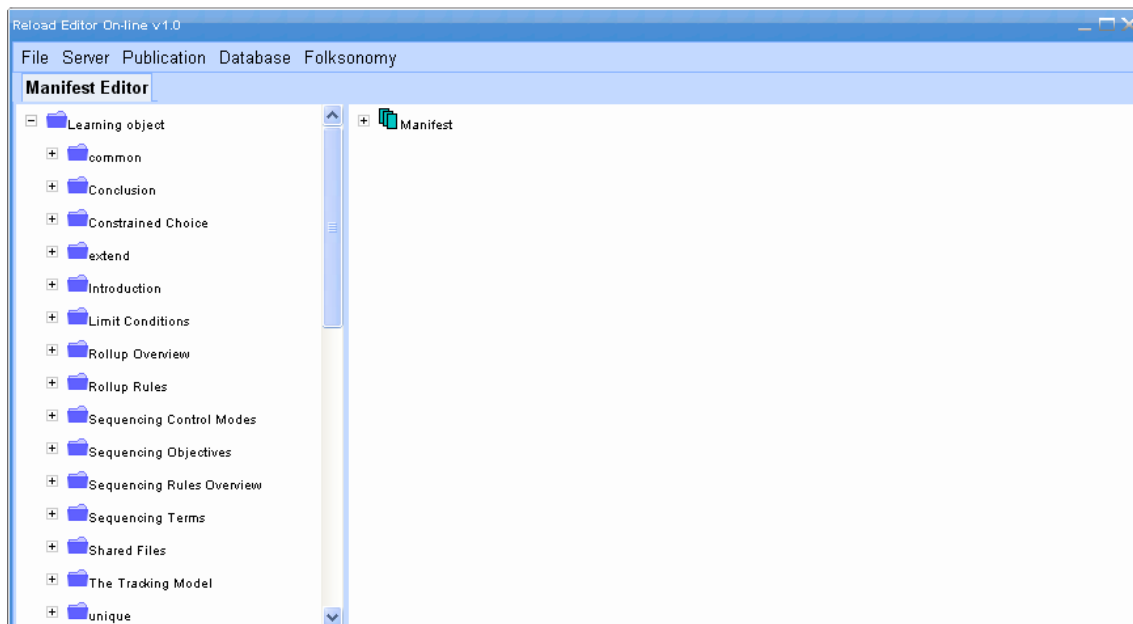


Figura 23: Aspecto del editor tras haber importado el objeto de la carpeta personal

Al igual que ocurría con el borrado de objetos de la carpeta personal, si intentamos importar un objeto al editor sin haber seleccionado ninguna entrada de la lista de archivos, el sistema mostrará un mensaje de error como el de la figura 21.

Si queremos salir sin importar ningún objeto pinchamos en el botón “Exit Load from Server”.

Opciones del menú Publication

El menú Publication ofrece la posibilidad de publicar cualquier objeto de aprendizaje, siempre y cuando lo tengamos almacenado en la carpeta personal. Publicar un objeto de aprendizaje significa que cualquier otro usuario podrá, a través del buscador (que ofrece este mismo menú), consultar, descargar e incluso editarlo y modificarlo y volver a publicar otra versión distinta. Por lo tanto con esta funcionalidad logramos que cuando un usuario lo desee pueda hacer público un objeto de aprendizaje.

Además de poder publicar, este menú también nos da la posibilidad de recuperar cualquier objeto público, incluso los que haya publicado el propio usuario, mediante la ayuda de un buscador de objetos de aprendizaje (ya sea de forma general o avanzada).

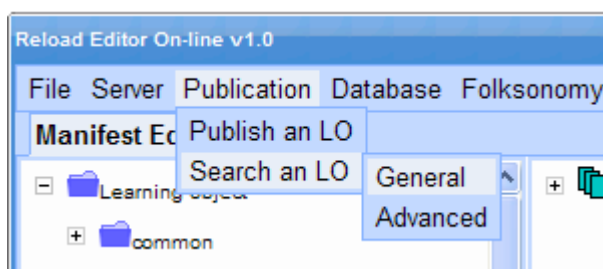


Figura 24: Menú Publication

Publicar un objeto de aprendizaje

Los pasos a dar para poder publicar algún objeto de aprendizaje son muy similares a los que se dan para importar un OA de la carpeta personal al editor o para eliminar un OA de la carpeta personal. Pinchamos en la opción del menú “Publish an LO” y se nos listará todo el contenido de nuestra carpeta personal.

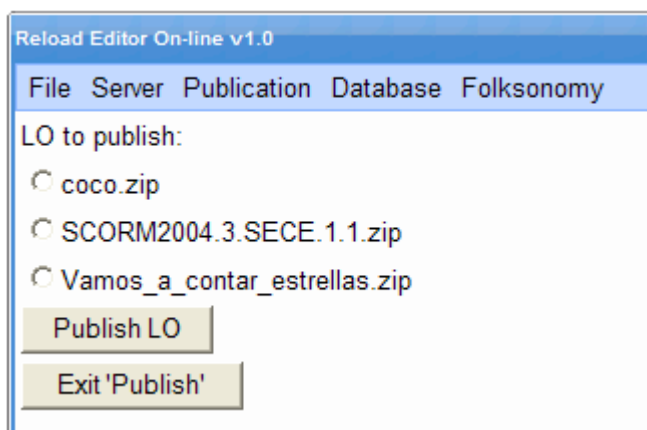


Figura 25: lista de posibles objetos a publicar

Llegados a este punto seleccionamos el objeto que queremos publicar y pinchamos en “Publish LO”. Si todo funciona correctamente tras un pequeño instante, el sistema nos mostrará un mensaje, confirmando que el objeto de aprendizaje seleccionado se ha publicado correctamente.



Figura 26: Mensaje de confirmación de publicación

Si el sistema nos muestra un mensaje como el de la figura 27, querrá decir que ya hemos un publicado un OA con ese mismo nombre, y nos pregunta si queremos sobrescribir el que ya existe. Si aceptamos se borraría todo el contenido del OA que teníamos antes publicado y se sustituye por el nuevo.

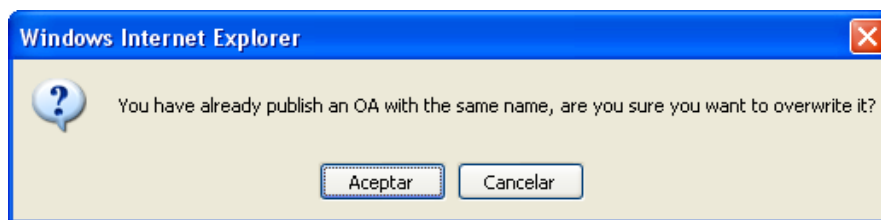


Figura 27: Mensaje de sobre escritura de OA público

AL igual que ocurría en anteriores apartados, si pinchamos en “Publish LO” sin haber seleccionado un objeto, el sistema nos recordará que debemos seleccionar uno. Si queremos salir sin publicar ningún objeto pinchamos en “Exit Publish”.

Búsqueda general de un objeto de aprendizaje

Si queremos realizar una búsqueda general todos los objetos públicos, seleccionamos la opción “General”, dentro de “Search an LO”, del menú Publication. Se abre una nueva pestaña dentro de la aplicación, en la que podemos ver un campo de texto y un botón de búsqueda. Si introducimos nuestra búsqueda en el campo de texto y pulsamos en el botón “Search!”, realizaremos una búsqueda por todos los objetos de aprendizaje que haya publicados en el servidor.

Los resultados encontrados se muestran siempre en forma de tabla, como podemos observar en el ejemplo de la figura 28.

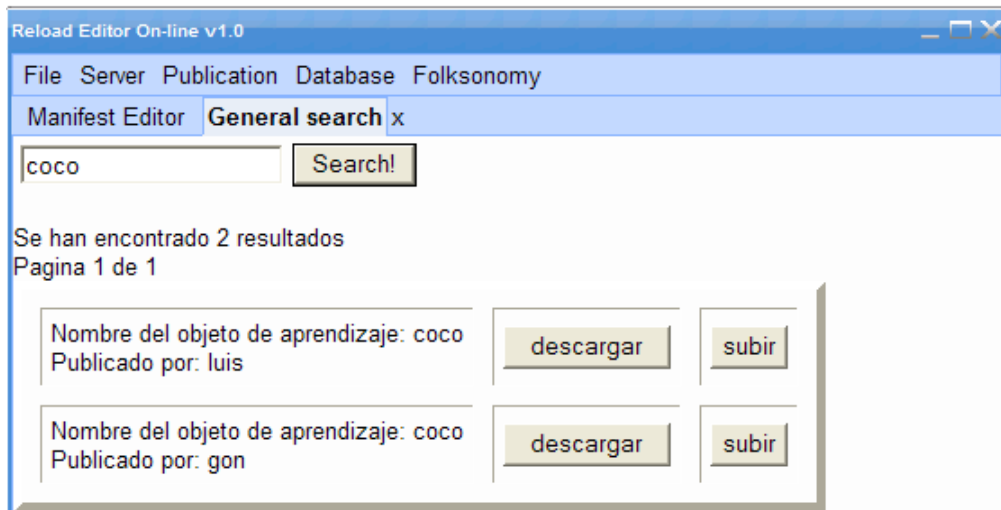


Figura 28: Ejemplo de resultados de búsqueda general

Por cada resultado encontrado, se nos muestra el nombre del objeto de aprendizaje, el nombre del usuario que lo ha publicado, y dos botones; uno para descargar localmente el archivo y otro para añadirlo directamente a nuestra carpeta personal.

Los resultados se muestran por páginas. Caben 5 resultados por página y en caso de que los resultados ocupen más de una página (como ocurre en el ejemplo de la figura 29) se nos proporcionan botones para avanzar por las distintas páginas de resultados. Existen botones para avanzar y retroceder y para visualizar la primera y la última página.

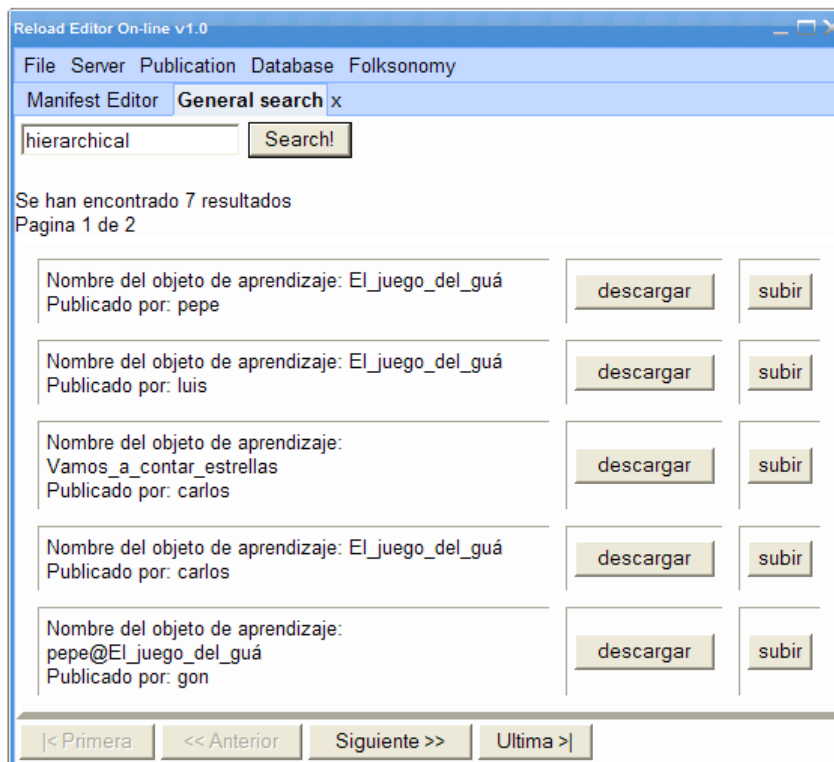


Figura 29: Ejemplo con más de una página de resultados

Si el buscador no encuentra ningún resultados para la búsqueda que hemos realizado, por supuesto no muestra ningún resultado, y a cambio muestra una mensaje informando de ello.

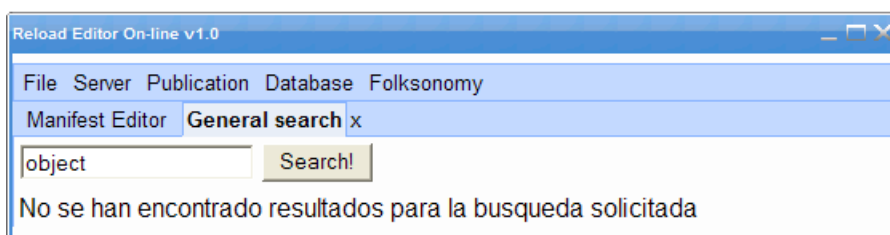


Figura 30: Ejemplo en el que no se han encontrado resultados

Búsqueda avanzada de un objeto de aprendizaje

La figura 31 nos muestra una vista general del buscador avanzado. Lo podemos obtener seleccionando la opción “Advanced” del menú Publication, desplegando la entrada “Search an LO”.

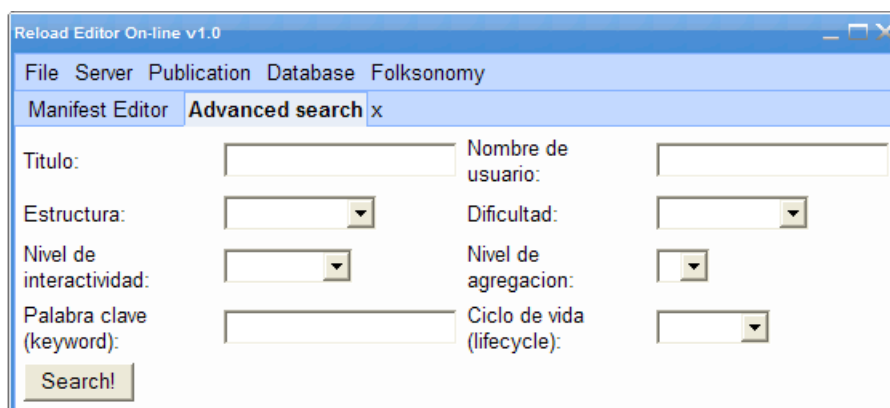


Figura 31: Vista general del buscador avanzado

Con la búsqueda avanzada, a diferencia de con la general, podemos elegir los campos por los que buscar. El buscador solo buscará coincidencias para los campos que queramos. Bastará con introducir texto o con seleccionar una de las opciones que nos ofrece, en cualquiera de los 8 campos de búsqueda. Podemos realizar búsquedas por el título del objeto de aprendizaje, por el usuario que lo ha publicado, por la estructura del objeto (linear, collection, atomic, networked, hierarchical), por el nivel de dificultad que entraña (very easy, easy, médium, difficult, very difficult), por el nivel de interactividad que dispone el OA (very low, low, médium, high, very high), por el nivel de agregación (1, 2, 3, 4), por alguna de las palabras clave que puede contener o por el ciclo de vida (draft, final, revised).

Los resultados para este tipo de búsqueda se muestran de la misma forma que ocurría en el caso de la búsqueda general. En cambio el orden de los resultados dependerá del número de coincidencias de cada objeto encontrado con los criterios introducidos. Es decir, se mostrará en primer lugar el objeto de aprendizaje que más coincidencias tenga con los criterios que requiere la búsqueda, y se mostrarán al final de la lista aquellos que menos se acerquen a la búsqueda solicitada.

Opciones del menú Database

El menú database nos ofrece la posibilidad de guardar los content preview generados por el editor (menú File) en la base de datos del servidor. Existe la posibilidad de guardar varias versiones de un content preview aunque hagan referencia al mismo objeto de aprendizaje. La única restricción es que en el momento de guardar el archivo en la base de datos, debemos tener una copia del archivo actual del editor, almacenado en nuestra carpeta personal. Por supuesto al igual que se pueden añadir archivos se pueden eliminar y descargar localmente.

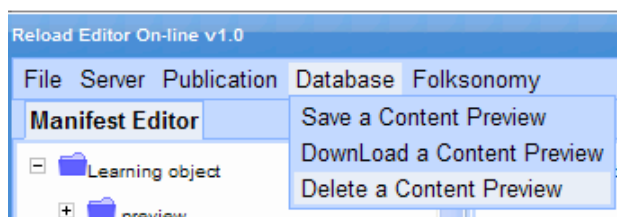


Figura 32: Menú Database

Guardar en la base de datos una versión de un content preview

Si queremos guardar una versión de un content preview del objeto que tenemos abierto en el editor, basta con seleccionar la opción “Save a Content Preview” del menú Database, como podemos observar en la figura 32. Se abrirá una ventana similar a la que teníamos en la figura 11 (importar un OA al editor). El proceso es el mismo, seleccionamos el archivo .zip que contiene el content preview, usando el explorador de archivos tras haber pulsado en “examinar”, y lo subimos al servidor pulsando en “upload”. El archivo se habrá subido correctamente una vez que el sistema nos deje pulsar en “ok” para cerrar la ventana.

Para poder guardar distintas versiones del content preview de un mismo objeto de aprendizaje basta con que tengan distinto nombre. Como vemos en la imagen 33, esto es posible, y además de guardar el nombre del content preview, se menciona el nombre del OA al que hace referencia.

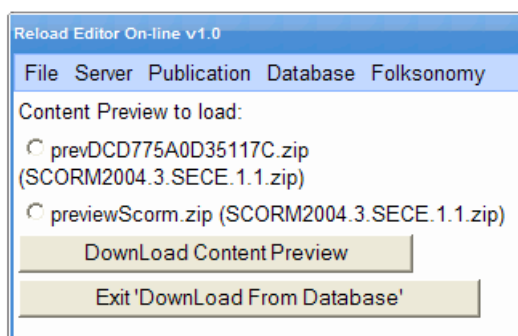


Figura 33: Lista de los content preview disponibles en la base de datos

En caso contrario el sistema nos dirá que ya existe un archivo con ese nombre y pide nuestra confirmación para sobrescribirlo por el nuevo (figura 34).

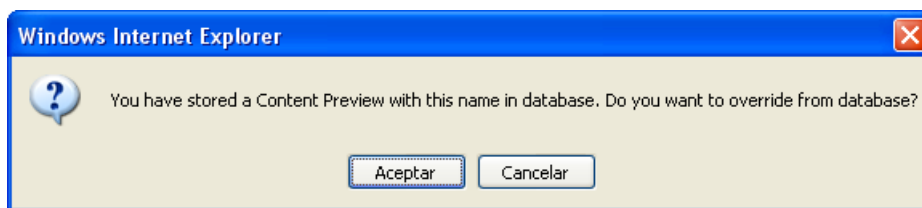


Figura 34: Mensaje de confirmación para sobrescribir un archivo en la base de datos

En caso de que no tengamos una copia en la carpeta personal, del objeto que actualmente tenemos abierto en el editor, el sistema nos muestra un mensaje de error como el que sigue



Figura 35: Mensaje de error al subir un content preview

Eliminar un content preview almacenado en la base de datos

La operación de eliminar un content preview de la base de datos es totalmente análoga a eliminar un objeto de aprendizaje del servidor. Los pasos son los mismos. Seleccionamos la opción “Delete a content preview” del menú Database y se nos lista todos lo tengamos almacenado en la base de datos. Seleccionamos el archivo que deseamos eliminar y pinchamos en el botón “Delete content preview”. Instantáneamente se habrá eliminado.

El error que nos muestra el sistema en caso de no haber seleccionado ningún archivo, es el de la figura 21.

Para salir sin eliminar ningún archivo, pinchamos en el botón “Exit Delete from database”.

Descargar un content preview almacenado en la base de datos

Para recuperar cualquier archivo que hayamos subido a la base de datos y descargarlo localmente, basta con seleccionar la opción “Download a content preview” del menú Database. Volvemos a tener la lista de archivos guardados en la base de datos, y tendremos que seleccionar uno para descargarlo. Pinchamos en el botón “Download Content Preview” y se abrirá una ventana de descarga con la que tendremos que decidir si abrir, guardar o cancelarla.



Figura 36: Descarga de un archivo almacenado en la base de datos

Si pulsamos en guardar tendremos que elegir nombre y destino para el archivo y pulsar en guardar. A partir de este momento disponemos de la copia del archivo guardada localmente en disco.

Si pulsamos en abrir, únicamente podremos ver el contenido del archivo y visualizar los archivos que contiene.

Si decidimos cancelar la descarga todo se queda como estaba.

En caso de no seleccionar ningún archivo y pulsar en “Download Content Preview” volveremos a obtener el mismo error de siempre (figura 21).

Para salir sin descargar ningún archivo pulsamos en “Exit Download from database”.

Opciones del menú Folksonomy

El menú Folksonomy centraliza las funcionalidades relacionadas con la socialización de objetos de aprendizaje. Nos permite etiquetar objetos, hacer búsquedas sobre las etiquetas, y además publicar por internet los objetos.

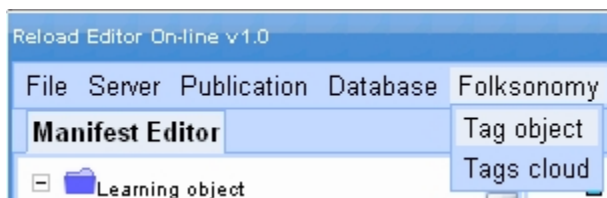


Figura 37: Detalle del menú Folksonomy

Etiquetar un objeto de aprendizaje

Seleccionando la opción “Tag object” del menú tenemos acceso a la siguiente pantalla:



Figura 38: Pantalla de etiquetado “Tag object”

Primero seleccionaremos el objeto que queremos etiquetar en el desplegable, y a continuación usaremos los botones + y – para añadir cuadros de texto, y escribiremos una etiqueta en cada uno. Cuando hayamos terminado, pulsamos el botón “Tag it!” para hacer efectivo el etiquetado.

Si el propietario del objeto lo desea, puede además consultar las etiquetas que ya tiene el objeto de aprendizaje usando el botón “Get object tags”. También se puede eliminar las etiquetas actuales del objeto mediante el checkbox “Reset object tags”. Estas dos opciones suelen usarse combinadas para eliminar etiquetas que el propietario de un objeto considera incorrectas, obteniendo primeramente las etiquetas, eliminando las incorrectas, marcando la opción de eliminar actuales y pulsando “Tag it!”

Publicar un objeto por internet

Desde esta pantalla tenemos también acceso a la funcionalidad de publicar un objeto de aprendizaje por internet. Seleccionamos primeramente el objeto en el desplegable, y después haremos click sobre el nombre de la web en la que deseamos publicar el objeto. A continuación se abrirá un nuevo navegador web. Es posible que al abrirse el nuevo navegador salga un mensaje de advertencia de que se está abandonando la aplicación y se procede a navegar por internet con los riesgos que ello conlleva, pulsaremos “Si” para continuar.

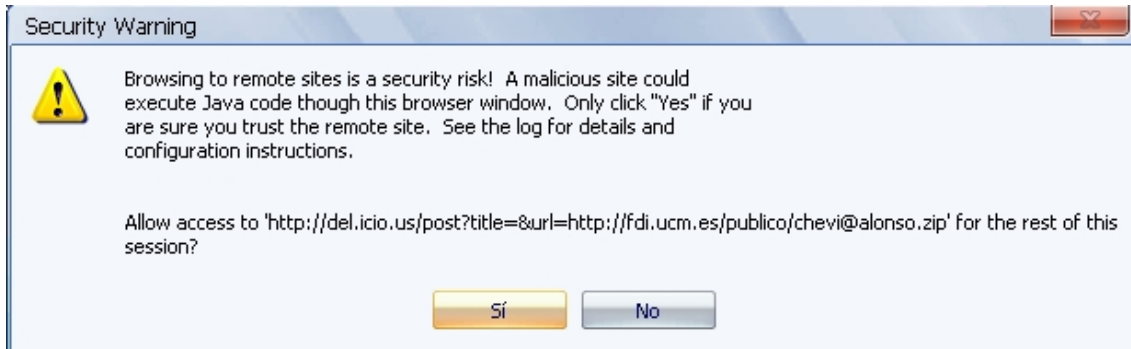


Figura 39: Mensaje de advertencia

Se nos pedirá que hagamos login en la web y tras ello procedemos a etiquetar la URL del objeto, tomamos como ejemplo del.icio.us

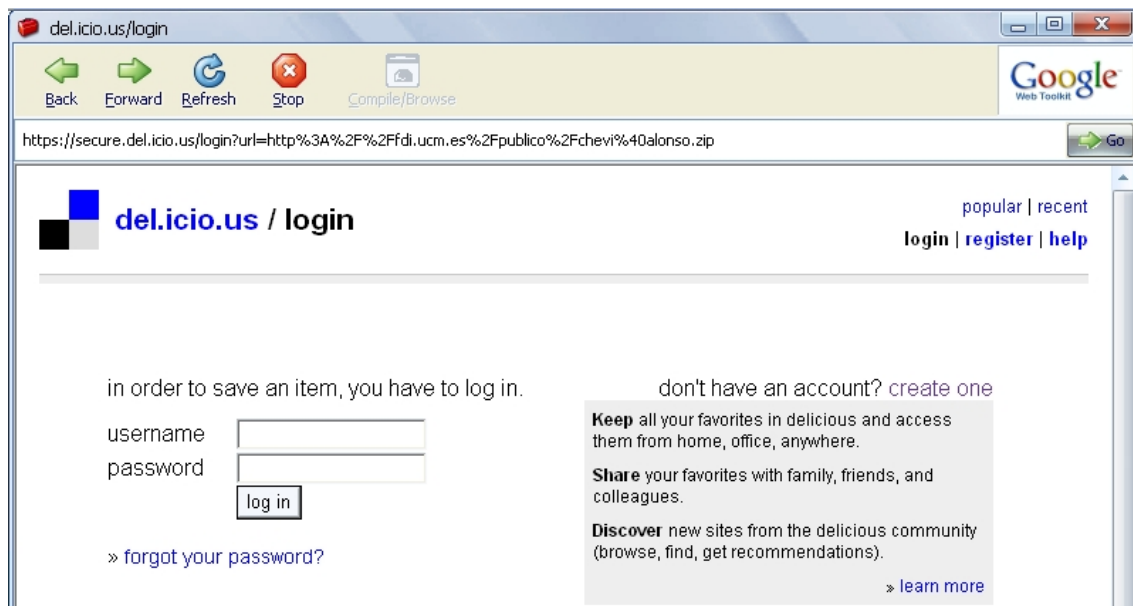


Figura 40: Login de usuario

Rellenamos los campos “description” y en el campo “tags” escribimos las etiquetas que caracterizan al objeto de aprendizaje separadas por espacios en blanco, y terminamos pulsando el botón “save”

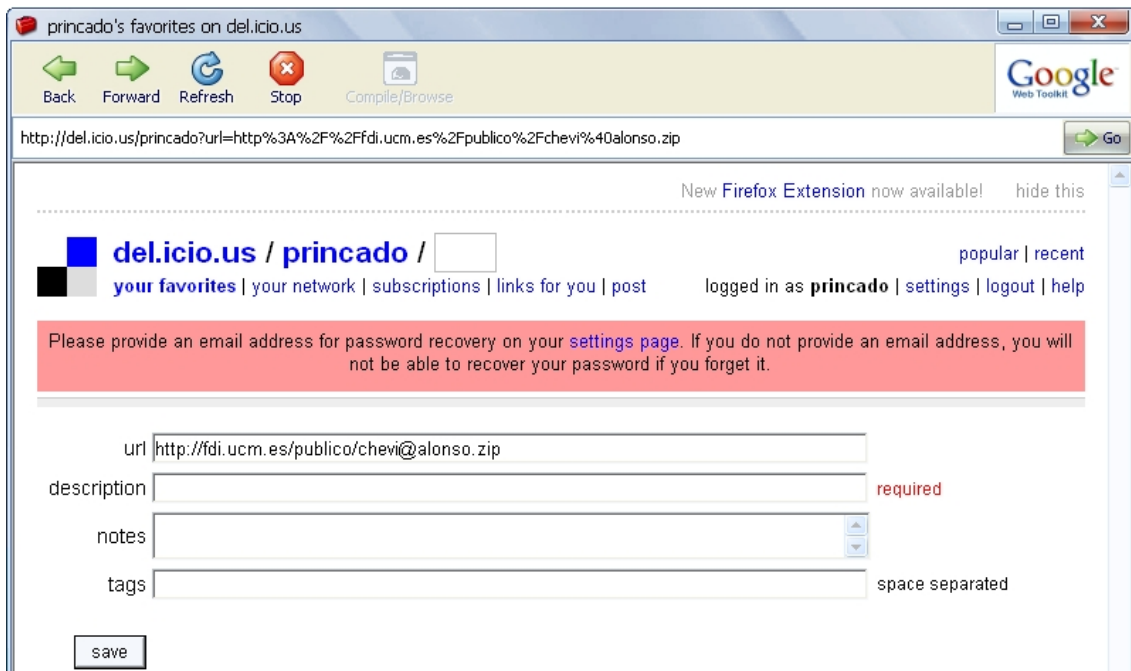


Figura 41: Etiquetado y publicación del objeto

Para cada web el proceso puede variar pero es basicamente similar.

Buscar objetos de aprendizaje mediante la nube de etiquetas

Seleccionando la opción “Tags cloud del menú tenemos acceso a la siguiente pantalla:



Figura 42: Nube de etiquetas

En la parte izquierda vemos una nube de etiquetas con formato. Estas son las 32 etiquetas más importantes que etiquetan a los objetos públicos y propios. Se muestran ordenadas alfabéticamente, y mediante un código de colores y tamaños por orden de importancia. Los colores van desde el rojo para las más importantes hasta el verde para las menos, pasando por naranja y amarillo. De esta forma encontramos fácilmente las más relevantes. Clicando sobre una etiqueta, restringimos la búsqueda a los objetos de aprendizaje que estén etiquetados con ella, y se refina la búsqueda generándose una nueva nube con las etiquetas de dichos objetos.

En la parte derecha vemos una lista de los objetos encontrados como resultado de la búsqueda, ordenados alfabéticamente y divididos en páginas de 5 objetos para mayor claridad de lectura. Para cada objeto se muestra su nombre y autor, y se ofrece la posibilidad de abrirlo en el editor o de etiquetarlo, abriéndose la pantalla vista en el epígrafe anterior, pero con algunas opciones (las reservadas al propietario de objeto) deshabilitadas.

En la parte inferior se van apilando las etiquetas que hemos clicado, de tal forma que los objetos tienen todas las etiquetas. En cualquier momento podemos volver atrás en la búsqueda clicando sobre cualquier etiqueta, o volver al principio de la búsqueda clicando sobre “Beginning”.

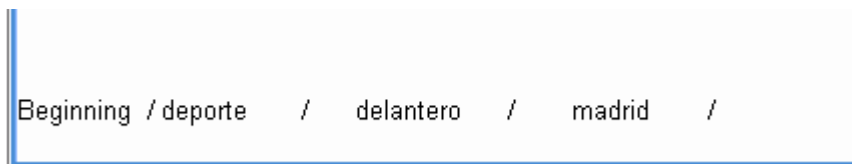


Figura 42: Etiquetas apiladas

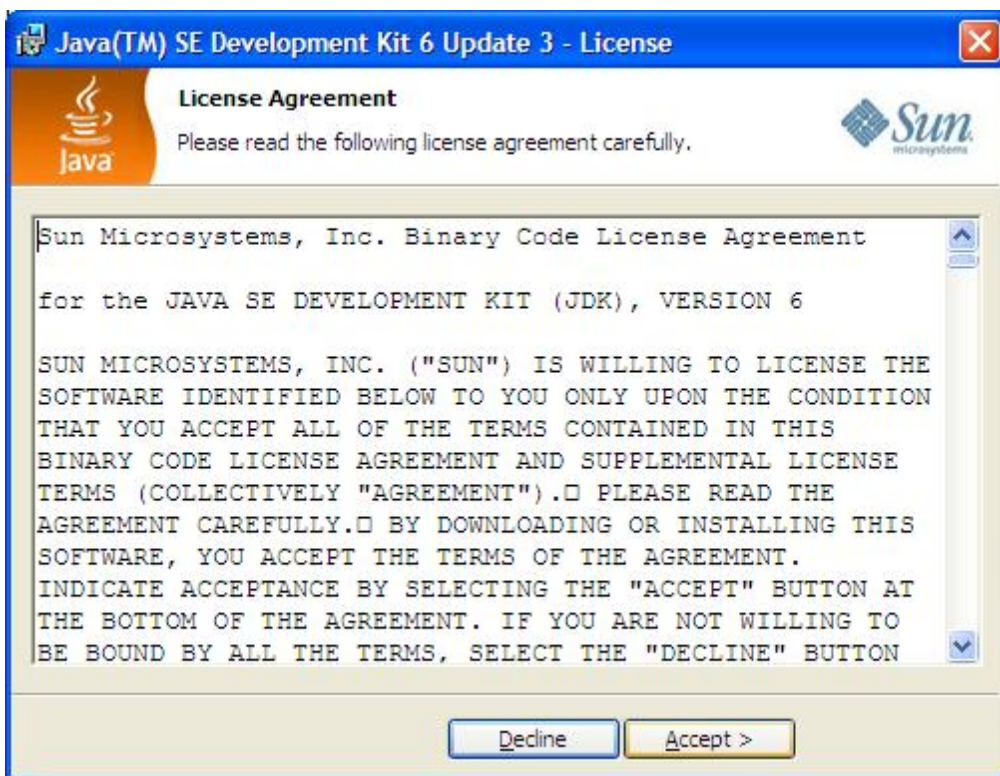
INSTALACIÓN DE LA APLICACIÓN GODACHEANT

Instalación de la máquina virtual de java

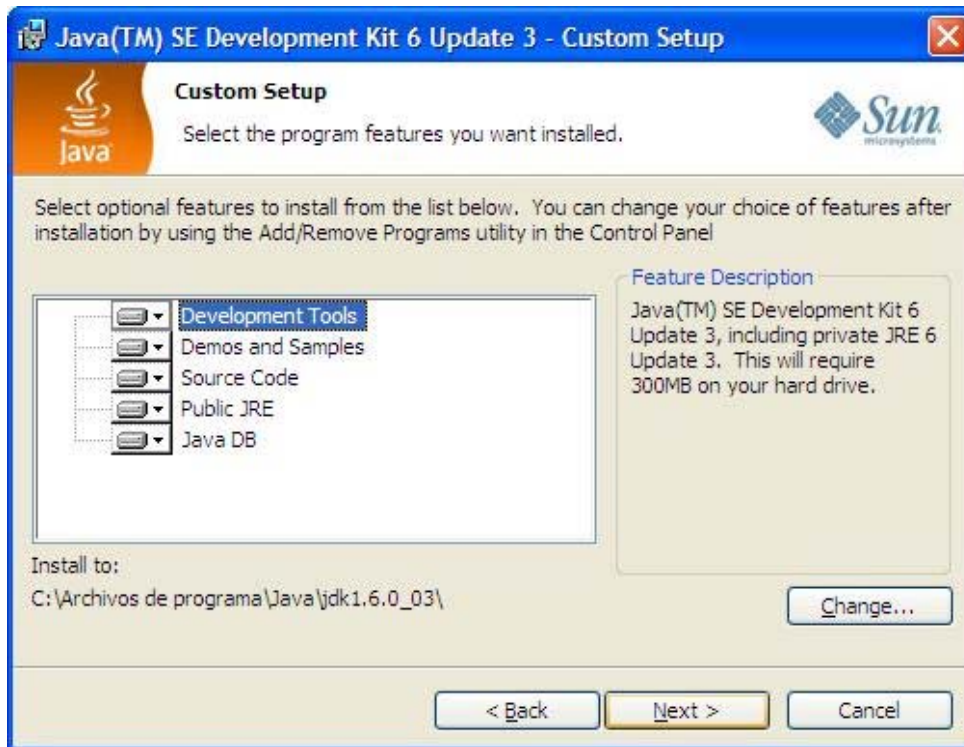
Pasos a seguir:

1º Hacemos doble click sobre el archivo “jdk-6u3-windows-i586-p” que está situado en la carpeta Instalación que hay en el CD que adjuntamos con el proyecto.

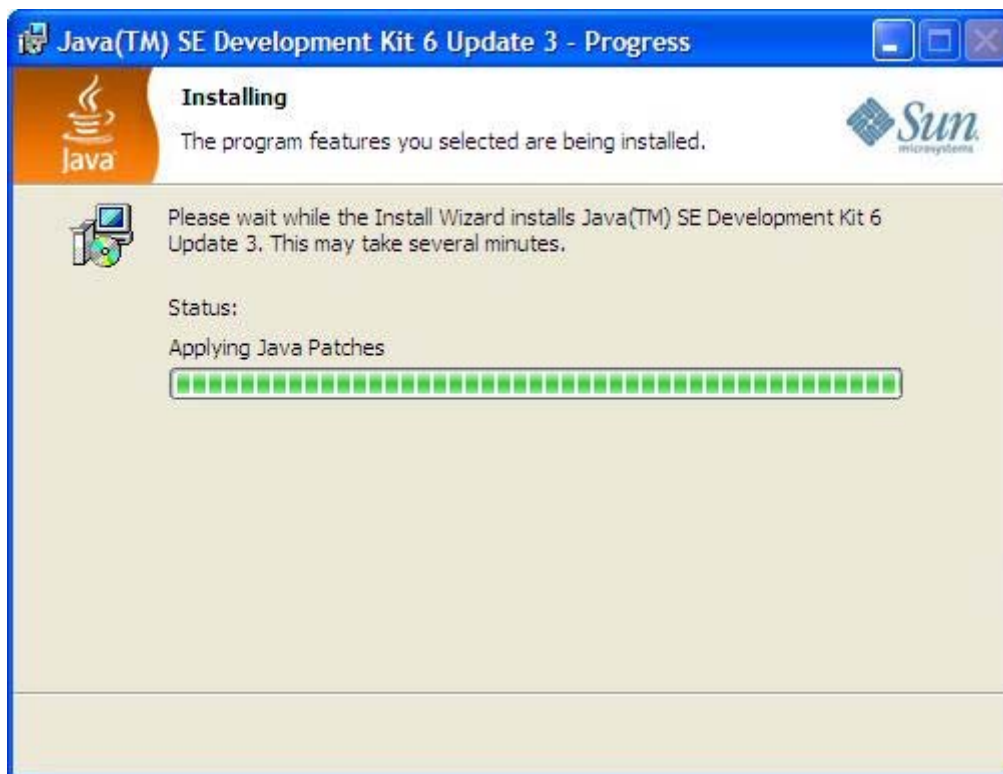
A continuación nos aparecerá la siguiente imagen:



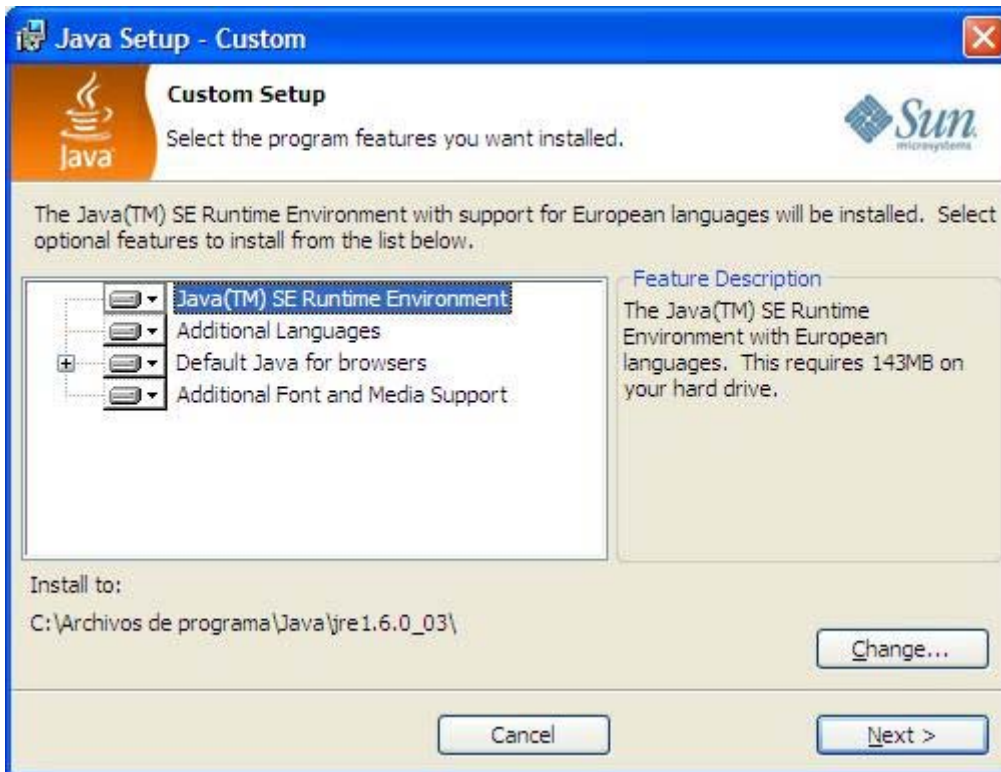
2º Aceptamos los términos de la licencia (botón Accept).
A continuación aparecerá la siguiente pantalla:



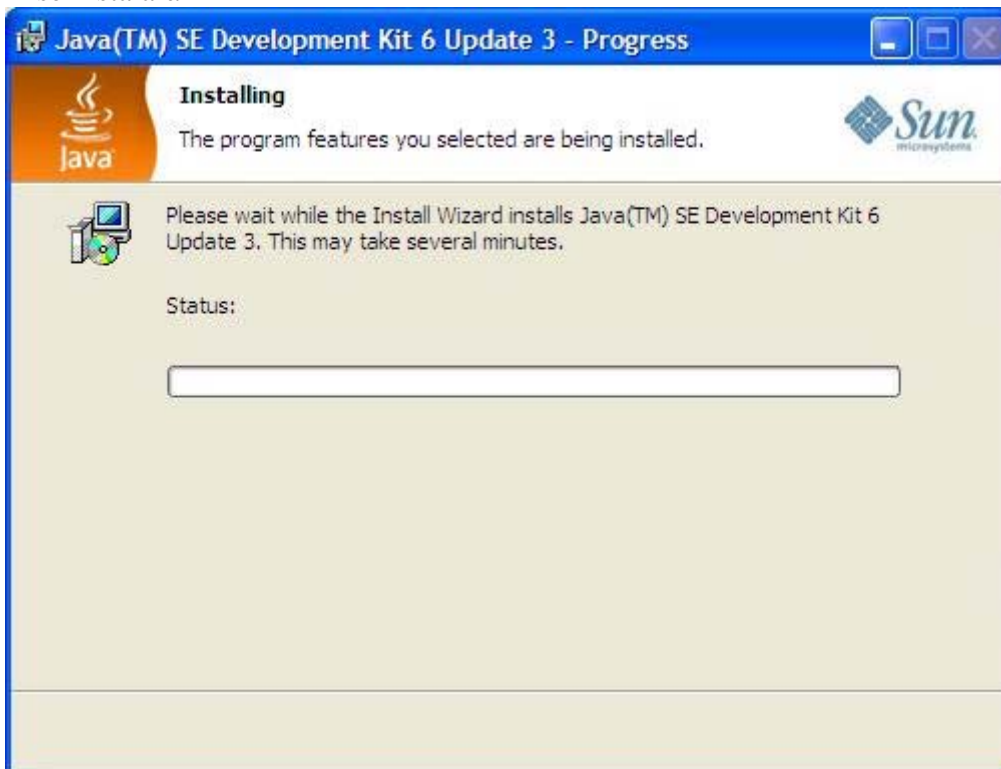
Y le damos a Next para que continúe la instalación copiando los archivos necesarios en C:/Archivos de programa/Java/jdk1.6.0_03 y se procederá a la instalación:



Después de este proceso se iniciará la instalación del jre
3° Aceptamos la instalación del jre dando al botón Next en la carpeta C:/Archivos de programa/Java/jre1.6.0_03



Y se instalará



Una vez concluida la instalación aparecerá una ventana para finalizar la instalación y si se quiere ver el archivo readme

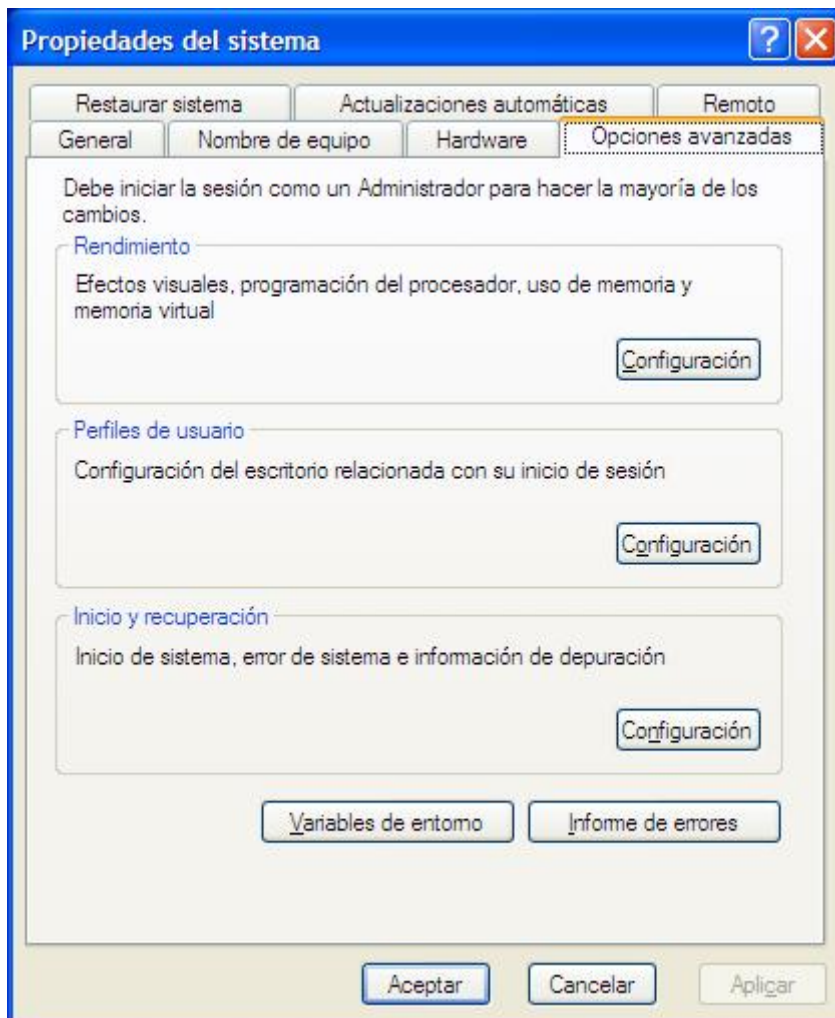


Una vez concluido todos estos pasos ya tenemos instalada todos los componentes necesarios de la máquina virtual de java versión 1.6.0_03.

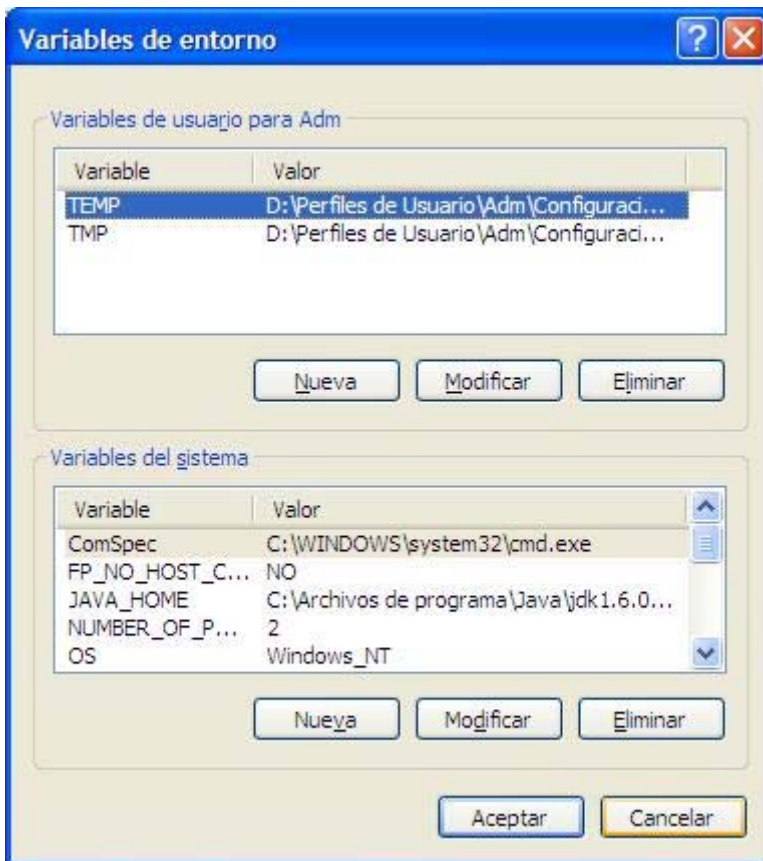
Ahora vamos a configurar sus variables de entorno:

1º Damos con el boton derecho en el escritorio al icono de Mi Pc y pulsamos propiedades

2º Nos colocamos en la pestaña de Opciones avanzadas y pulsamos sobre el botón Variables de entorno.



3° Pulsamos en el boton Nueva de abajo (Variables del sistema)



4° Creamos la variable de entorno JAVA_HOME:\Archivos de programa\Java\jdk1.6.0_03 (que es la ruta donde hemos instalado la maquina virtual de java)



Instalación y configuración de MySQL

Pasos a seguir:

1º Hacemos doble click sobre el setup.exe localizado en la carpeta “MySql\mysql-5.0.51a-win32” que está situado en la carpeta Instalación que hay en el CD que adjuntamos con el proyecto.

Aparecerá un asistente de instalación, pulsamos Next



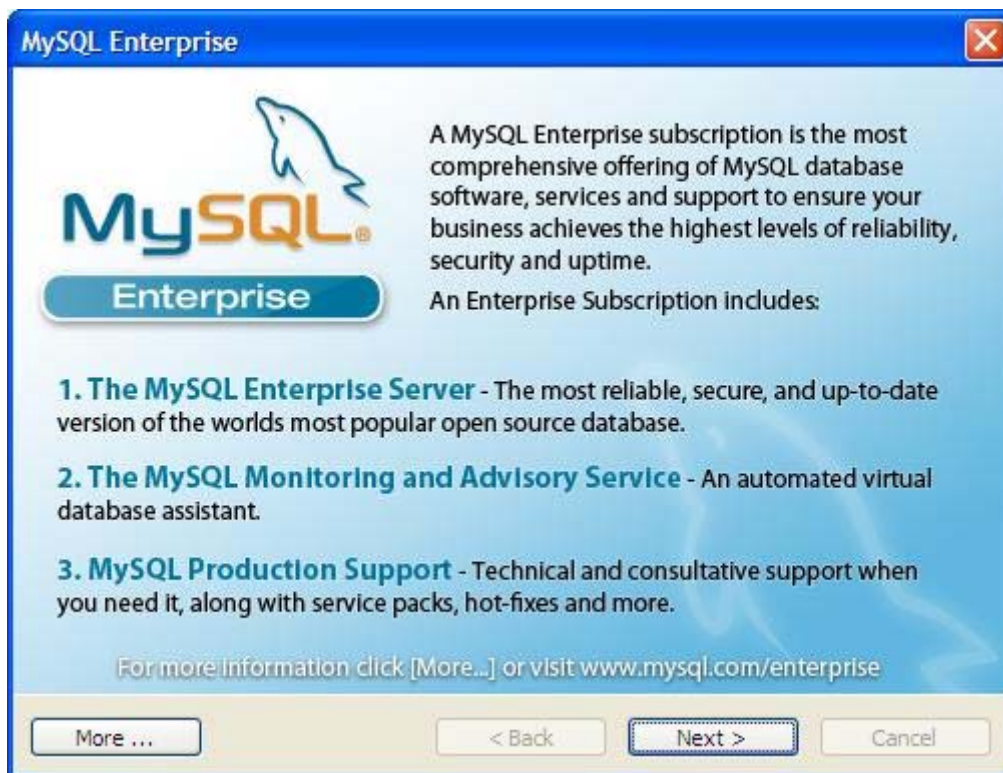
2º En el siguiente paso elegimos una instalación Typical y le damos a Next



3º Al ser la instalación típica se nos instalará en C:/Archivos de programa/MySL/MySQL Server 5.0 y le damos a Install



4° En este paso volvemos a darle a Next



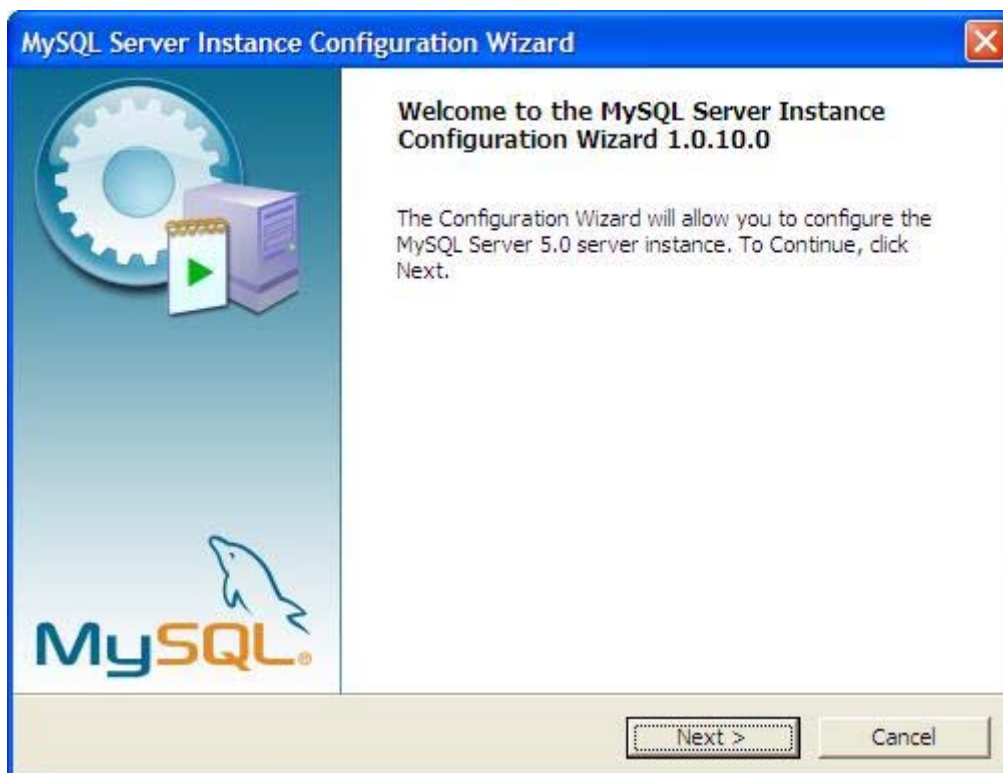
5° Pulsamos Next



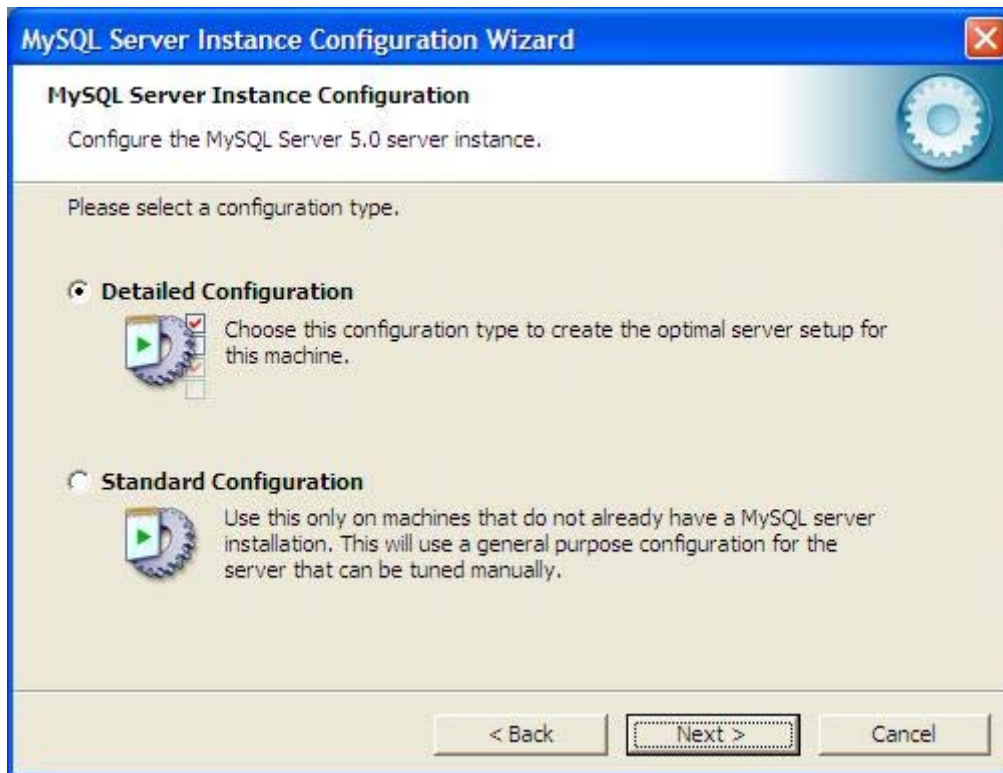
6° Nos aparece ahora la opción de configurar el MySQL Server y la dejamos marcada y le damos a Finish para terminar la instalación y empezar con la configuración.



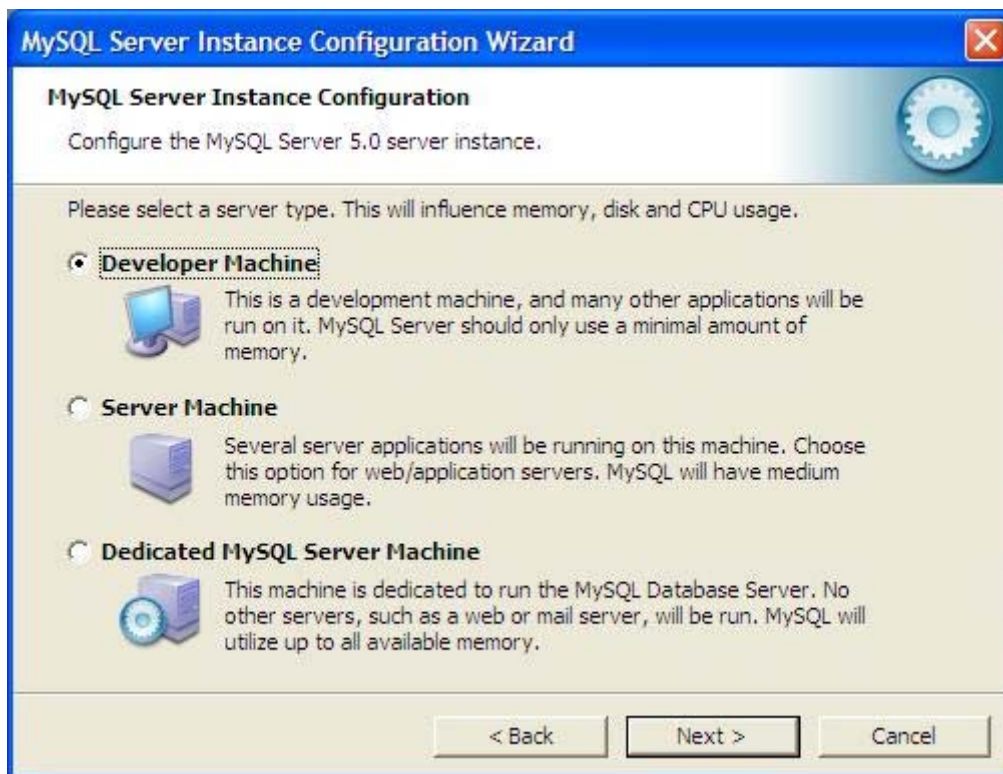
7° Pulsamos Next



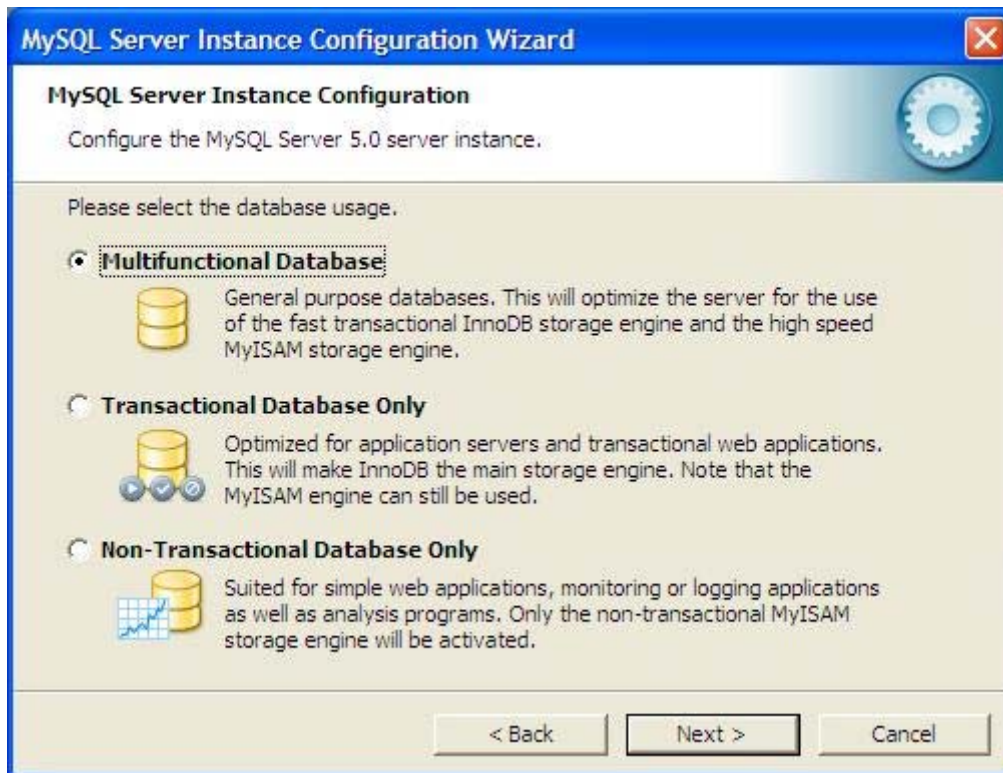
8º Elegimos la opcion Detailed Configuration y pulsamos Next



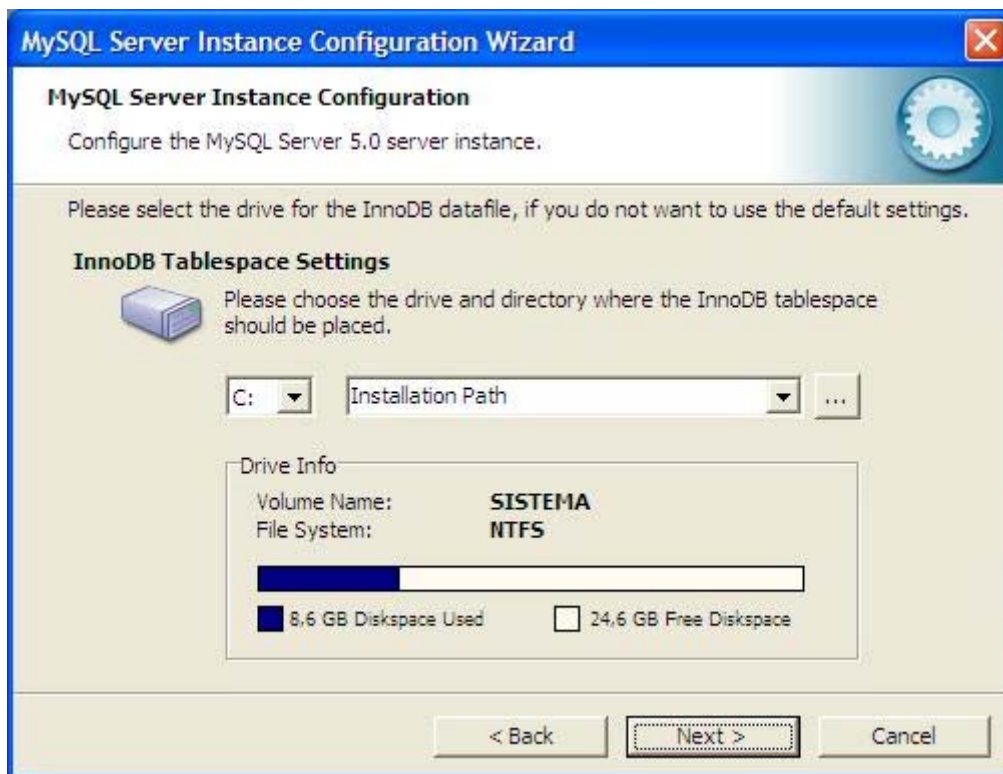
9º Elegimos Developer Machine y pulsamos Next



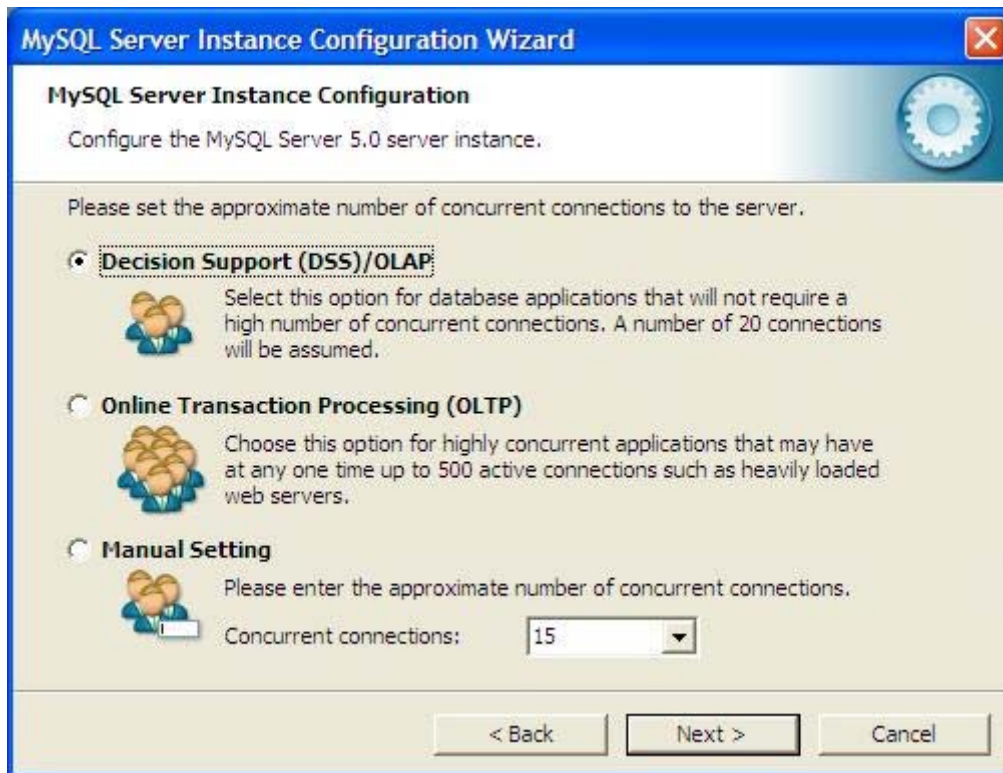
10° Elegimos Multifunctional Database y pulsamos Next



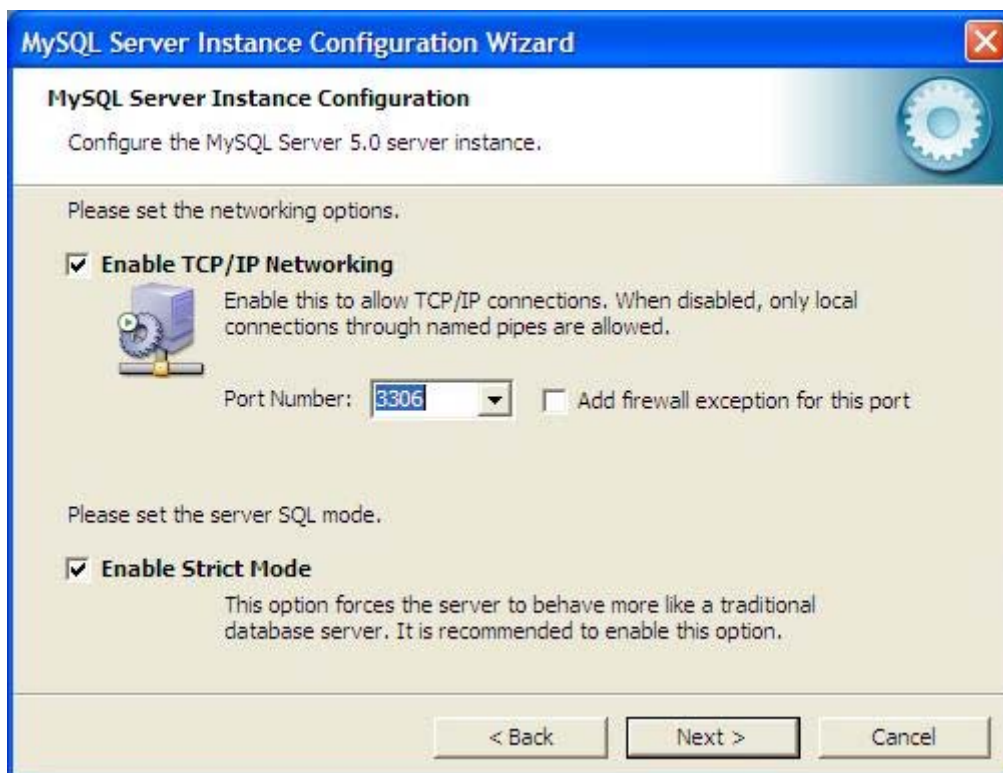
11° Lo dejamos tal y como esta y pulsamos Next



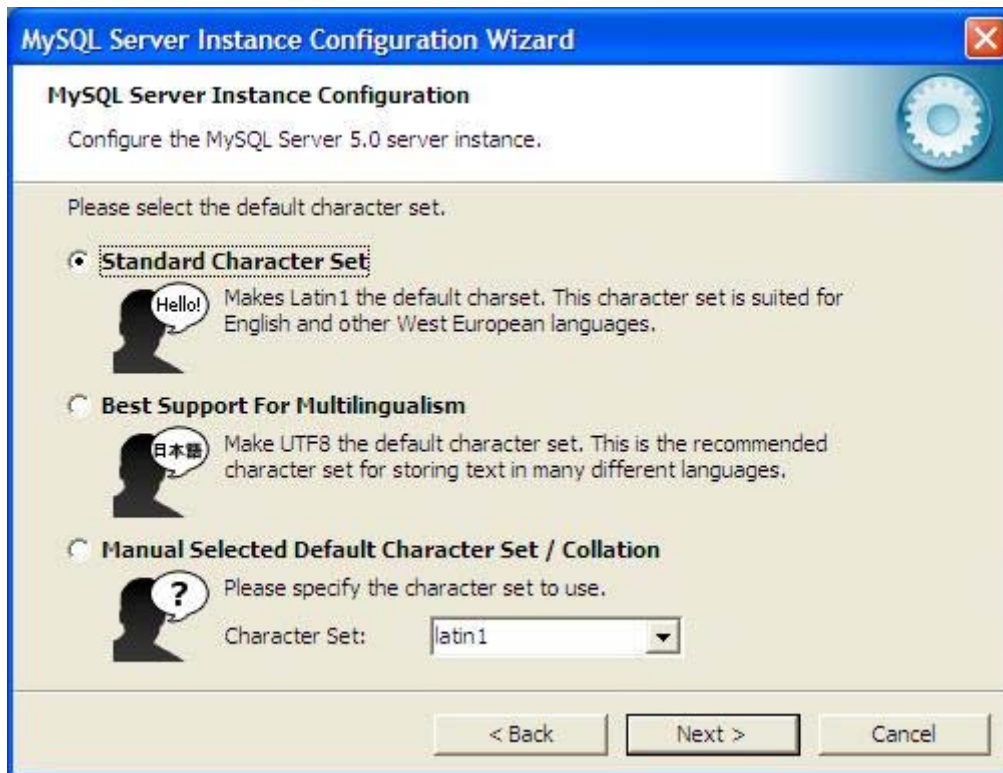
12° Seleccionamos Decision Support (DSS)/OLAP y pulsamos Next



13° Marcamos las opciones Enable TCP/IP Networking y Enable Strict Mode y como número de puerto el 3306 y pulsamos Next.



14° Elegimos Standard Character Set y pulsamos Next



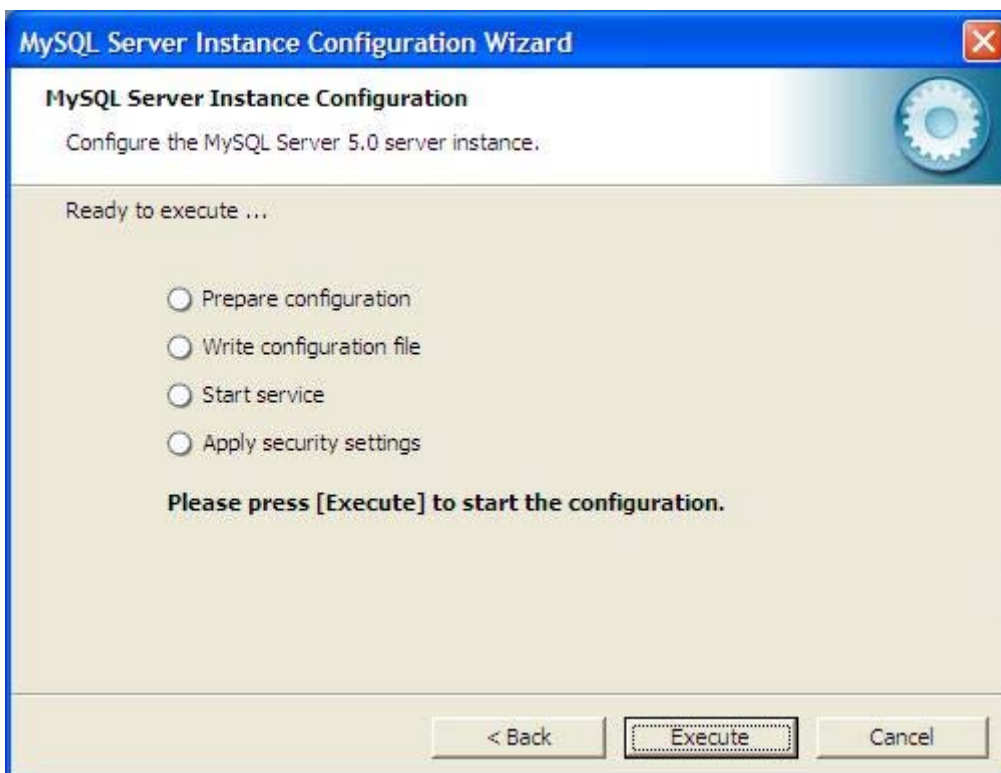
15° Lo instalamos como Install As Windows Service y marcamos Launch the MySQL Server automatically y pulsamos Next.



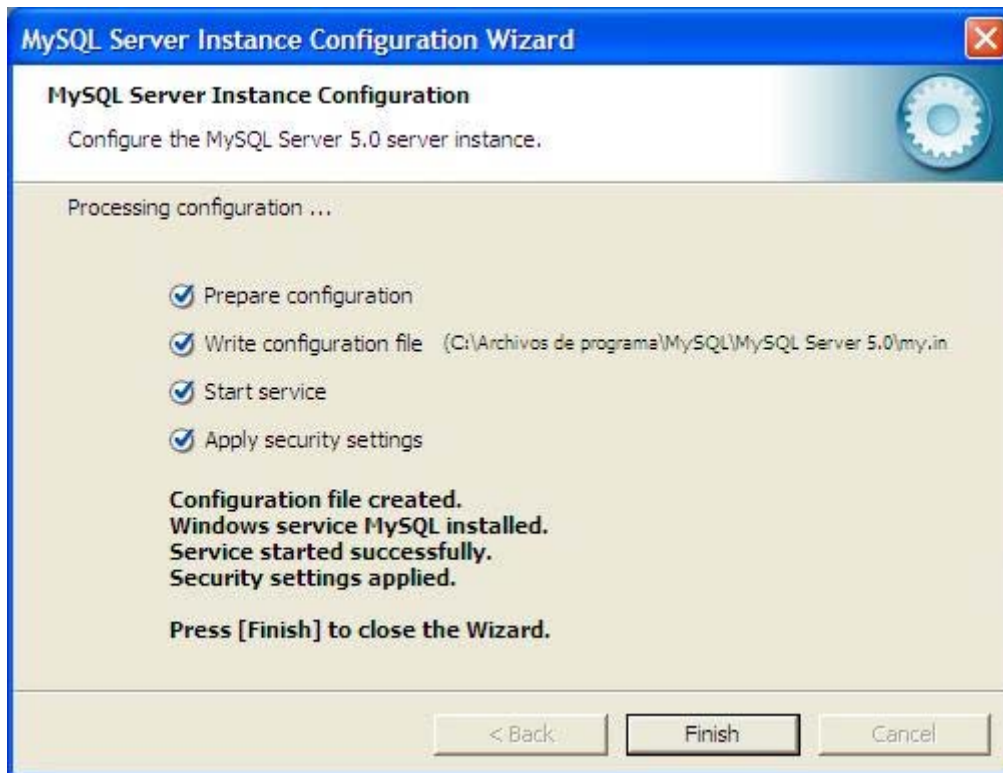
16º Escribimos en nuestro caso como contraseña de root “reload” y marcamos Enable root access from remote machines



17º Pulsamos execute para que empiece la configuración



18° Una vez terminada la configuración pulsamos Finish para terminarla



Una vez terminada la instalación nos vamos a “C:\Archivos de programa\MySQL\MySQL Server 5.0” y editamos el my.ini para terminar de configurarlo.

Ahora crearemos el parámetro *max_allowed_packet* y le asignaremos el valor de 16M, esto es debido a que sino por defecto solo permite subir archivos de 1M a la base de datos. Hay que localizar la sección [mysqld] y añadir una nueva línea tal y como se ve en la siguiente imagen:

```
67 [mysqld]
68
69 # The TCP/IP Port the MySQL Server will listen on
70 port=3306
71 max_allowed_packet=16M
72
--
```

IMPORTANTE: Para que surta efecto esta variable hay que reiniciar el ordenado. Si vuelves a usar el asistente de configuración del MySQL pierdes la línea añadida y tienes que volverlo a hacer.

Para tener las tablas de la base de datos para la aplicación ejecutamos el MySQL, introducimos la contraseña y después copiamos el contenido del archivo “base de datos.txt” que esta en la carpeta Instalacion del CD del proyecto.

```
create database reload;  
use reload;
```

```
create table usuario(  
    nombre varchar(12) not null primary key,  
    password varchar(10) not null);
```

```
create table archivosUsuario(  
    nombre          VARCHAR(12),  
    nombreArchivo VARCHAR(50),  
    publico bool,  
    size VARCHAR(10),  
    archivo LONGBLOB,  
    PRIMARY KEY(nombre,nombreArchivo),  
    foreign key(nombre)  
    references usuario(nombre)  
    ON DELETE CASCADE);
```

```
create table tag(  
    tag varchar(20) not null primary key,  
    apariciones SMALLINT);
```

```
create table etiquetado(  
    nombre varchar(12) not null,  
    nombreArchivo varchar(50) not null,  
    tag varchar(20) not null,  
    primary key(nombre, nombreArchivo, tag),  
    foreign key(nombre, nombreArchivo)  
    references archivosUsuario(nombre, nombreArchivo)  
    ON DELETE CASCADE,  
    foreign key(tag)  
    references tag(tag));
```

```
create table archivosContentPreview(  
    nombre          VARCHAR(12),  
    nombreArchivo VARCHAR(50),  
    nombreContentPreview VARCHAR(50),  
    size VARCHAR(10),  
    archivo LONGBLOB,  
    PRIMARY KEY(nombre,nombreArchivo, nombreContentPreview),  
    foreign key(nombre, nombreArchivo)  
    references          archivosUsuario(nombre,          nombreArchivo));
```

Una vez instalado el MySQL vamos a instalar el driver para luego configurar el acceso con el ODBC.

Pasos a seguir:

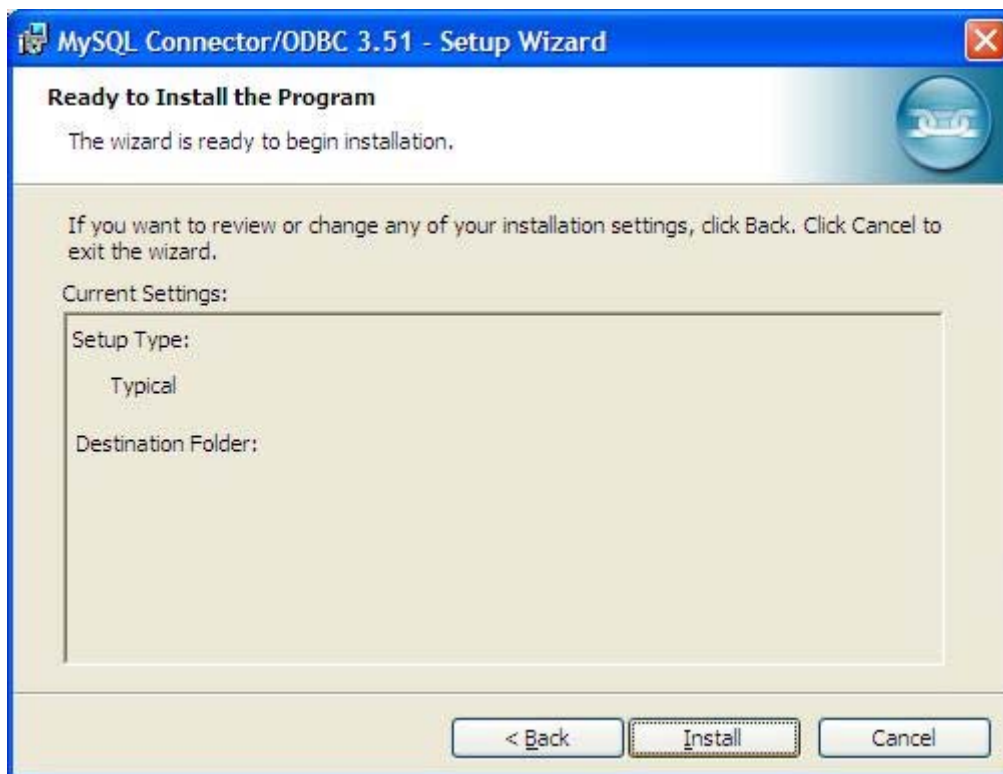
1° Hacemos doble click sobre el setup.exe localizado en la carpeta “MySQL\mysql-connector-odbc-3.51.23-win32” que está situado en la carpeta Instalación que hay en el CD que adjuntamos con el proyecto. Aparecerá la siguiente pantalla y pulsamos Next



2º Elegimos la instalación Typical y pulsamos Next



3º Pulsamos Install

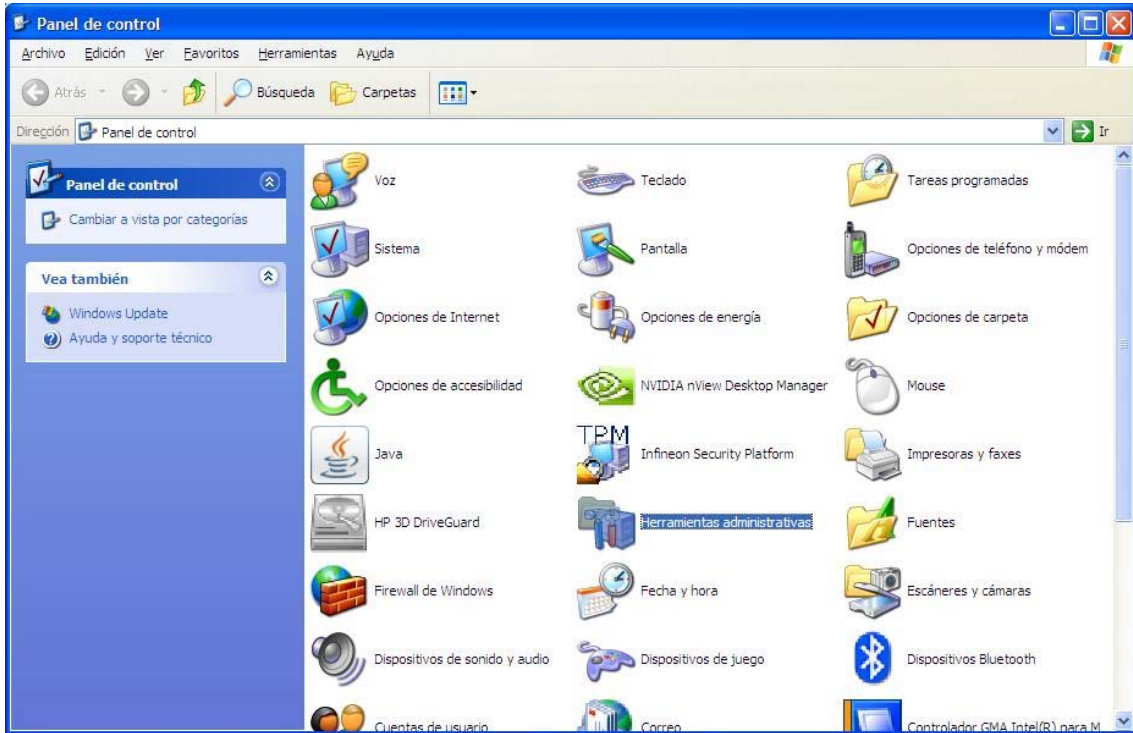


4º Y una vez terminado el proceso de instalación pulsamos Finish

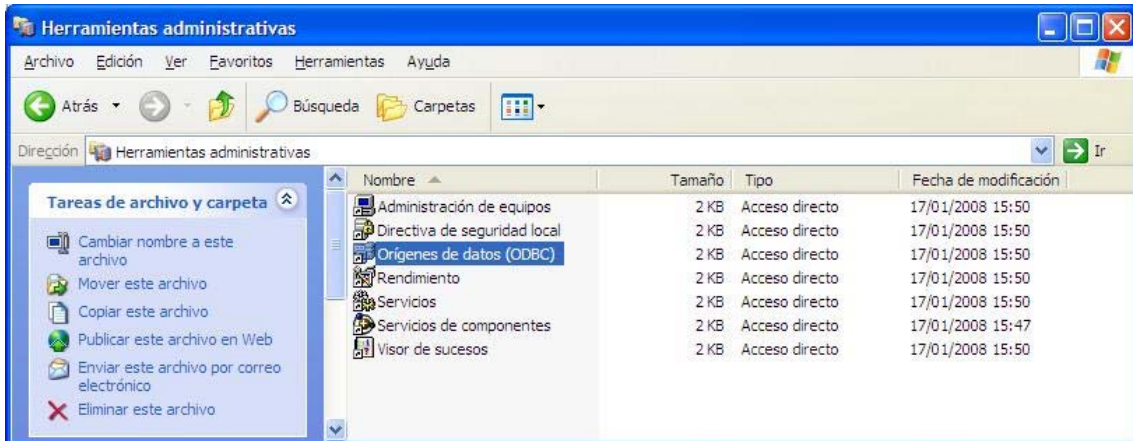


Una vez instalado el MySQL y su driver hay que configurar el ODBC para poder acceder desde la aplicación así que hacemos lo siguiente.

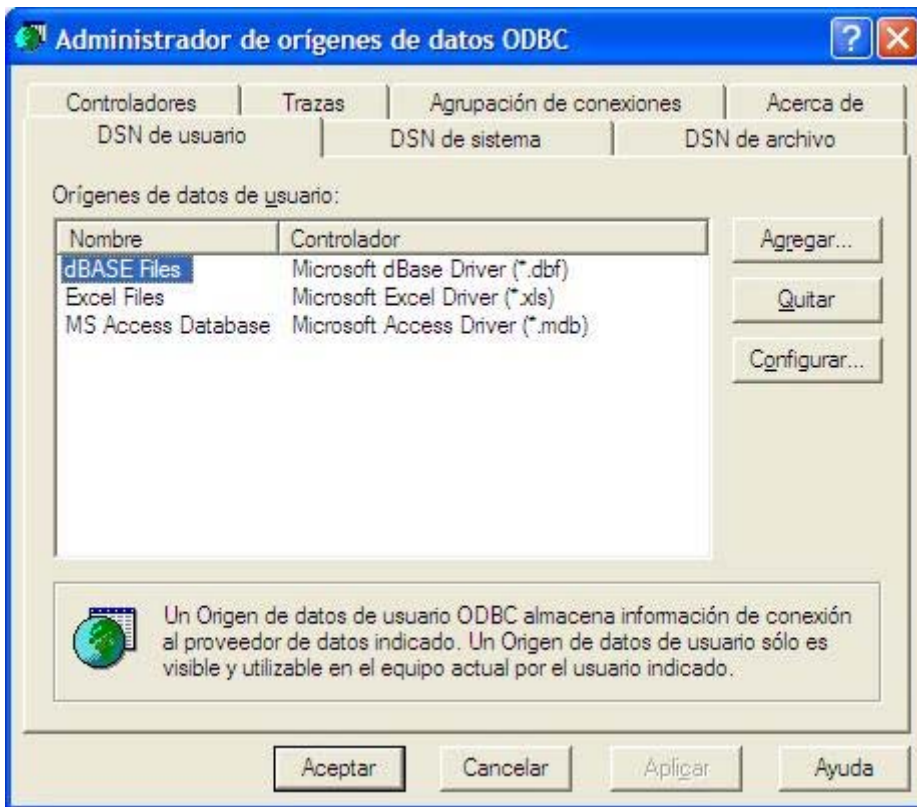
1º Inicio → Configuración → Panel de control y seleccionamos Herramientas administrativas



2º Elegimos Orígenes de datos (ODBC)



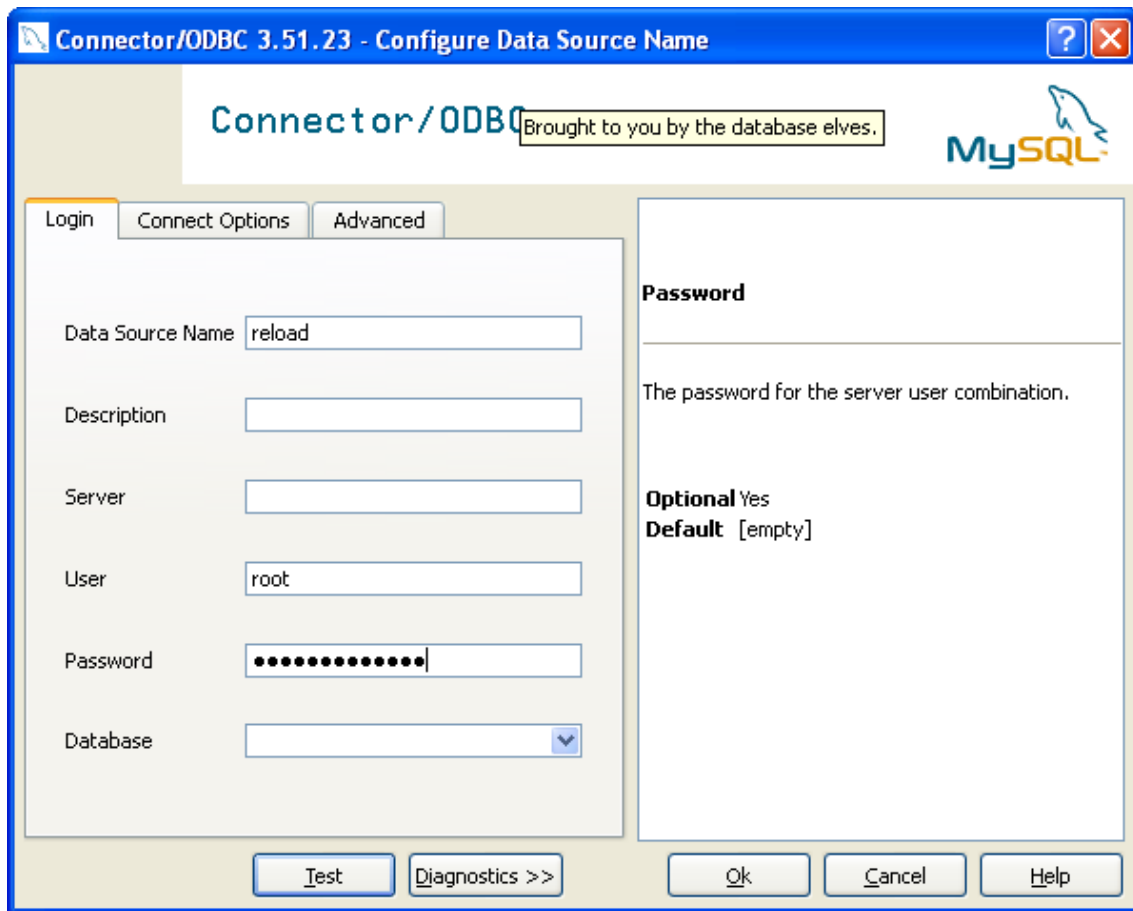
3º Damos a agregar uno nuevo



4º Seleccionamos el MySQL ODBC 3.51 Driver que es el driver que acabamos de instalar y pulsamos Finalizar



5° Configuramos el connector con los siguientes datos (Password en nuestro caso “reload”) y probamos la conexión pulsando Test y una vez que lo ha verificado pulsamos OK para terminar.



Y una vez realizado todos estos pasos ya tenemos configurada la base de datos para la aplicación.

Configuración del conexiones.properties

Si queremos modificar la localización de la base de datos o el password del usuario root de la base de datos lo podemos hacer sin problemas ya que con un simple cambio en el WEB-INF/classes/conexiones.properties del Godacheant.war permite hacerlo.

El contenido del conexiones.properties es el siguiente:

```
mySql.driver=com.mysql.jdbc.Driver  
mySql.url=jdbc:mysql://localhost/reload  
mySql.user=root  
mySql.password=reload
```

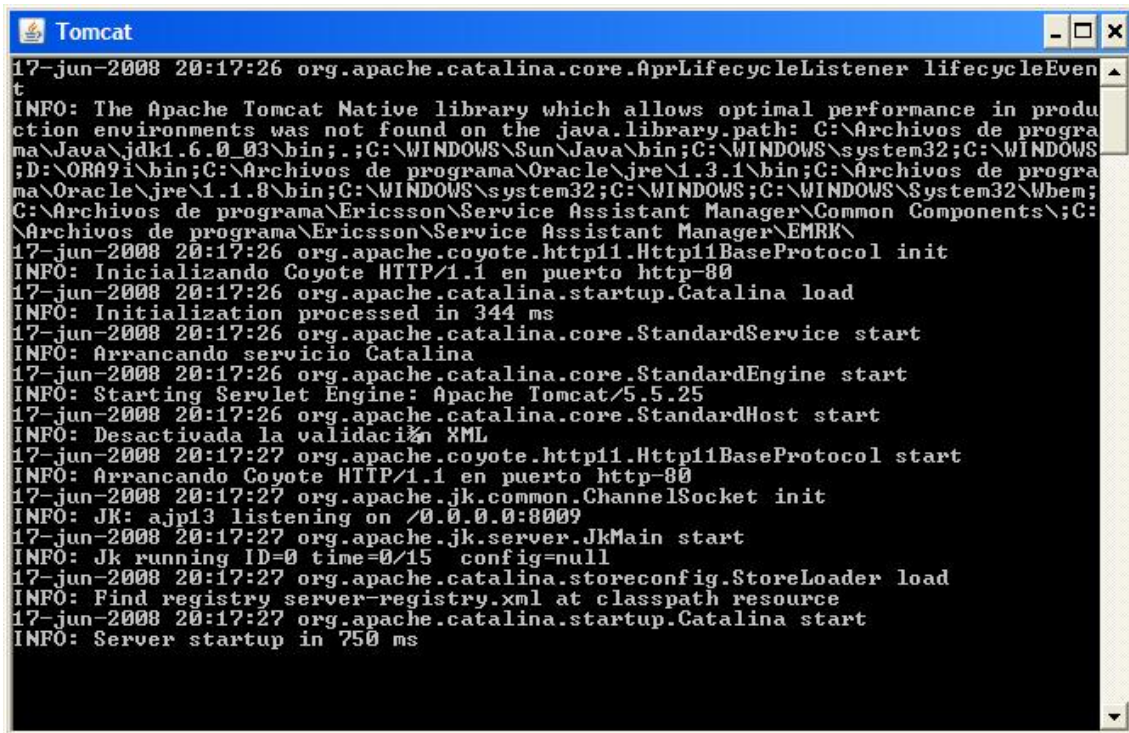
Si cambiamos la localización de la base de datos tendríamos que sustituir en mySql.url=jdbc:mysql://localhost/reload la palabra localhost por la dirección ip de la máquina donde este situada. Si por el contrario al registrar el driver queremos darle otro nombre distinto de reload se sustituiria la palabra reload por la que pusieramos.

Si ponemos otra contraseña distinta de reload, deberíamos sustituir mySql.password=reload poniendo la contraseña que elijamos.

Instalación Apache Tomcat

Proporcionamos un apache tomcat binario que no necesita instalación, simplemente hay que copiarlo en el directorio que quiera y ya funciona.

En su carpeta webapps es donde desplegaremos el .WAR y para utilizarlo daremos doble click sobre el startup.bat que esta en la carpeta /bin.

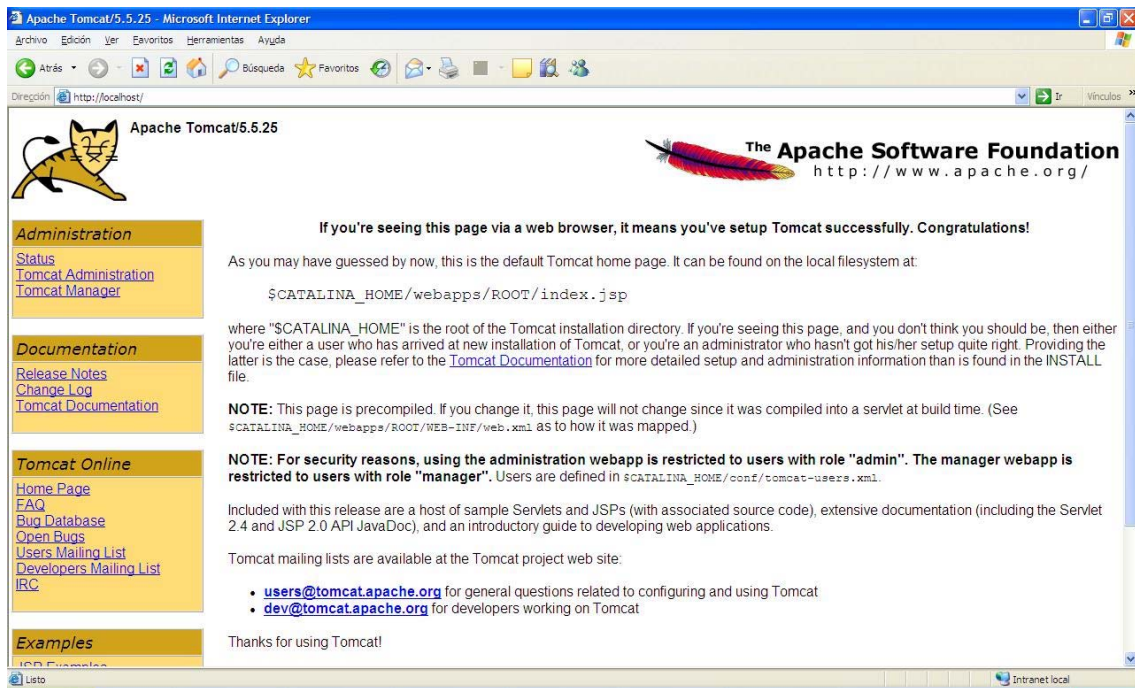
A screenshot of a Windows console window titled "Tomcat". The window displays a series of log messages from the Apache Tomcat startup process. The messages include timestamps, class names, and status reports such as "INFO: The Apache Tomcat Native library which allows optimal performance in production environments was not found...", "INFO: Inicializando Coyote HTTP/1.1 en puerto http-80", "INFO: Arrancando servicio Catalina", "INFO: Starting Servlet Engine: Apache Tomcat/5.5.25", "INFO: Desactivada la validación XML", "INFO: Arrancando Coyote HTTP/1.1 en puerto http-80", "INFO: JK: ajp13 listening on /0.0.0.0:8009", "INFO: Jk running ID=0 time=0/15 config=null", "INFO: Find registry server-registry.xml at classpath resource", and "INFO: Server startup in 750 ms". The console window has a standard Windows title bar with minimize, maximize, and close buttons.

Para evitar que descomprima el War en la carpeta conf en el archivo Server.xml hay que modificar lo siguiente:

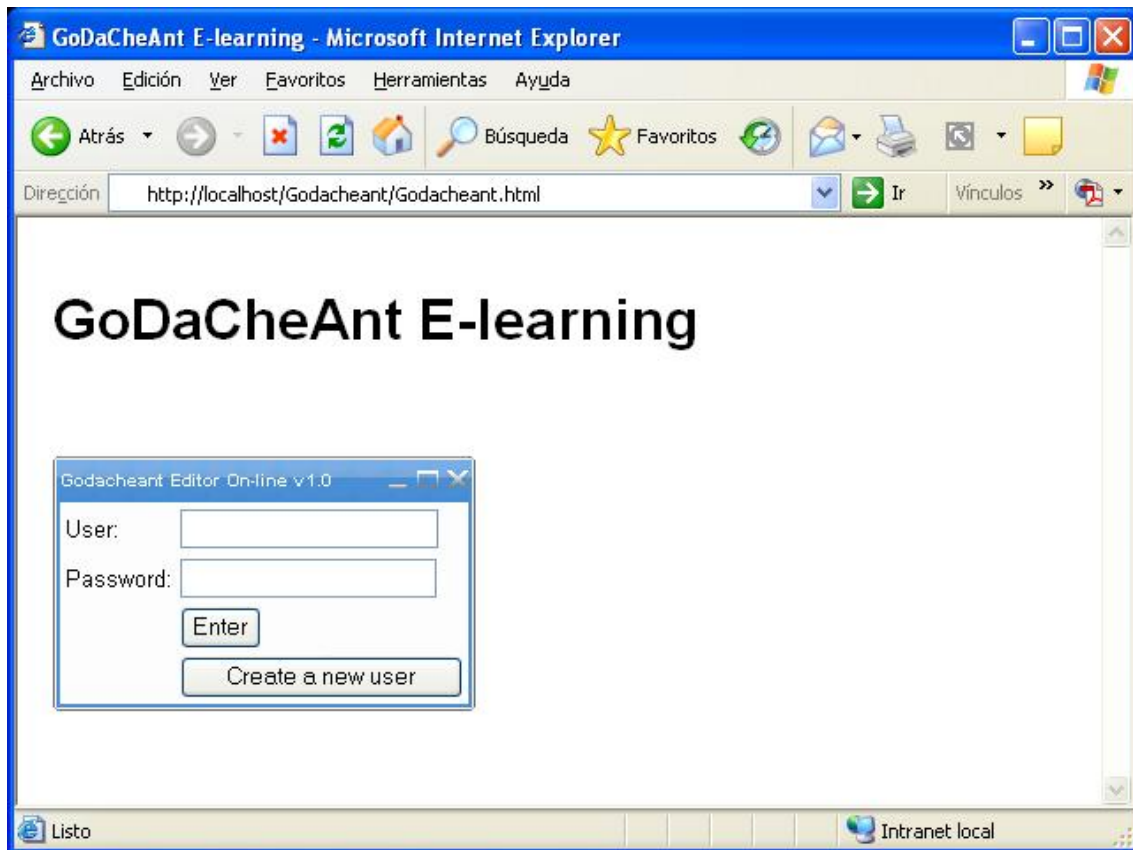
```
<Host name="localhost" appBase="webapps"
  unpackWARs="false" autoDeploy="true"
  xmlValidation="false" xmlNamespaceAware="false">
```

También está configurado el apache que hay en la carpeta Instalacion sobre el Puerto 80, con lo cual cuando queramos acceder al proyecto podemos poner el Puerto 80 o como es por defecto el que usan los exploradores podemos no ponerlo y se accede igualmente.

Para ver que funciona podemos abrir un explorador y escribir <http://localhost> o tambien <http://localhost:80> y para saber que funciona correctamente deberiamos ver la siguiente pantalla



Una vez realizados todos estos paso podemos copiar el Godacheant.war que esta en la carpeta de Instalacion que hay en el CD del proyecto y copiarlo en la carpeta webapps del tomcat, en nuestro caso en C:\Archivos de programa\apache-tomcat-5.5.25\webapps y una vez copiado ejecutamos el tomcat como hemos explicado anteriormente y una vez arrancado abrimos el Explorer y escribimos <http://localhost/Godacheant/Godacheant.html> y nos aparecera la aplicación (si se accede desde una máquina externa en vez de poner localhost hay que poner la ip de la máquina donde este instalado el programa).



Configuración del server.properties

En este archivo localizado en WEB-INF/classes/server.properties se debe escribir la ruta del servidor donde esta instalado el tomcat.

El contenido actual del archivo es:
server.url=www.fdi.ucm.es

Una vez que se instale la aplicación habría que ver cual es la dirección donde está instalado el tomcat y sobrescribir server.url=www.fdi.ucm.es con la nueva ruta.

OTRAS INSTALACIONES

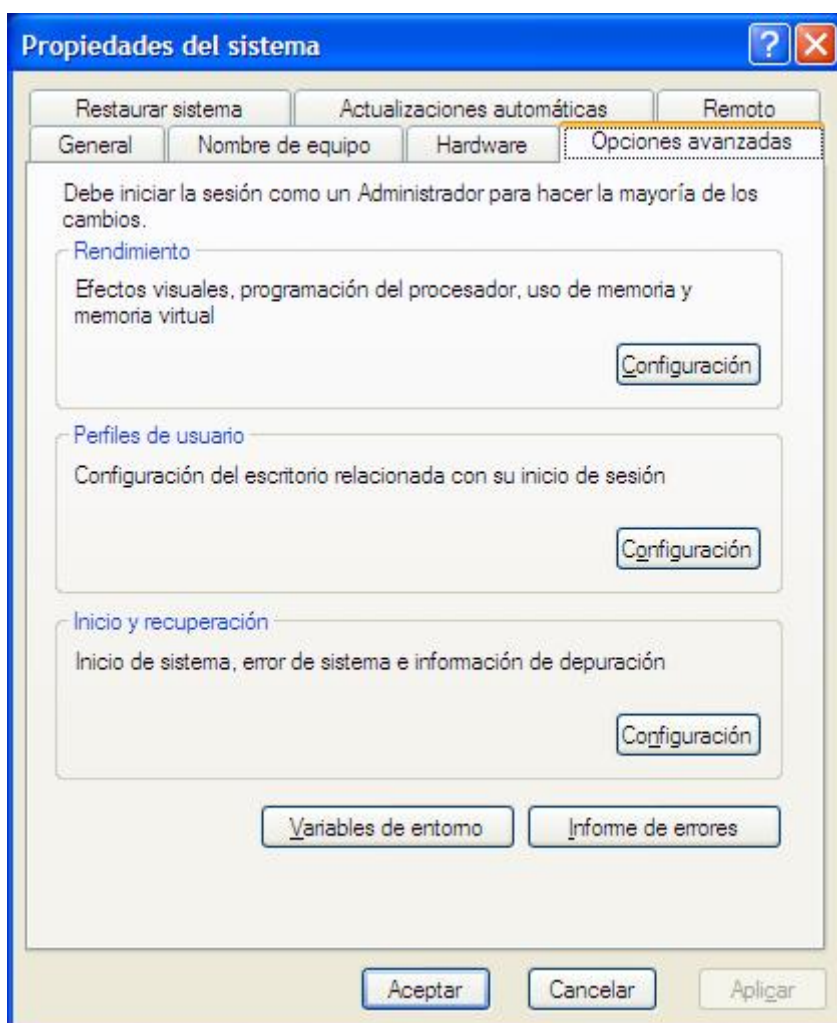
Instalación XmlBeans

Copiamos la carpeta xmlbeans-2.2.0 que hay en la carpeta Instalaciones que hay en el CD del proyecto por ejemplo en C\:

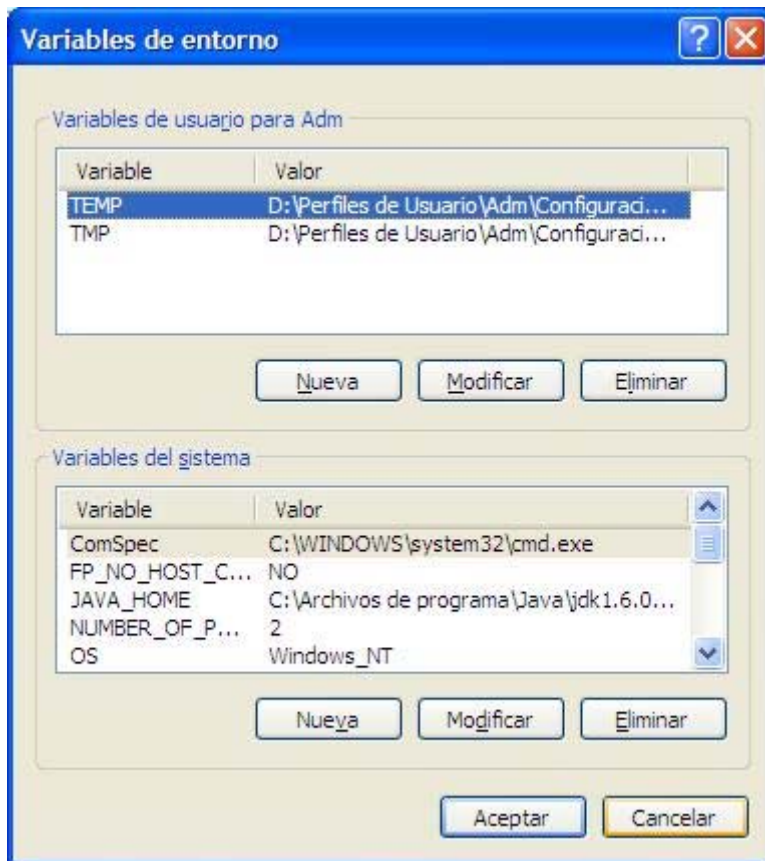
Configurar las variables de entorno:

1º Damos con el boton derecho en el escritorio al icono de Mi Pc y pulsamos propiedades

2º Nos colocamos en la pestaña de Opciones avanzadas y pulsamos sobre el botón Variables de entorno.



3º Pulsamos en el boton Nueva de abajo (Variables del sistema)



Añadir a classpath y a path los siguiente:

Classpath:

%XMLBEANS_HOME%\lib\xbean.jar;%XMLBEANS_HOME%\lib\jsr173_1.0_api.jar;%XMLBEANS_HOME%\lib\xmlpublic.jar

Path: %XMLBEANS_HOME%\bin

Crear XMLBEANS_HOME (ruta donde esta descomprimido el xmlBeans)

XMLBEANS_HOME C:\xmlbeans-2.2.0

Instalación del Apache-Ant-1.7.0

Copiamos la carpeta apache-ant-1.7.0 que hay en la carpeta Instalaciones que hay en el CD del proyecto en C.

Crear la variable de entorno ANT_HOME y si se ha extraido en C, el valor de la variable de entorno seria: C:\apache-ant-1.7.0

CREACIÓN DEL GODACHEANT.WAR Y DEL VALIDADOR

Introducción

Antes de explicar como crear el Godacheant.war y el validador de los objetos de aprendizaje vamos a explicar brevemente la estructura del proyecto y que es lo que contiene cada carpeta para saber antes que es lo que estamos haciendo realmente. También vamos a explicar la estructura interna del Godacheant.war y además mostraremos el listado de código que hemos cambiado y añadido a los esquemas sobre los originales para poder crear el validador sobre objetos de aprendizaje 2004 extendido.

Estructura del Proyecto:

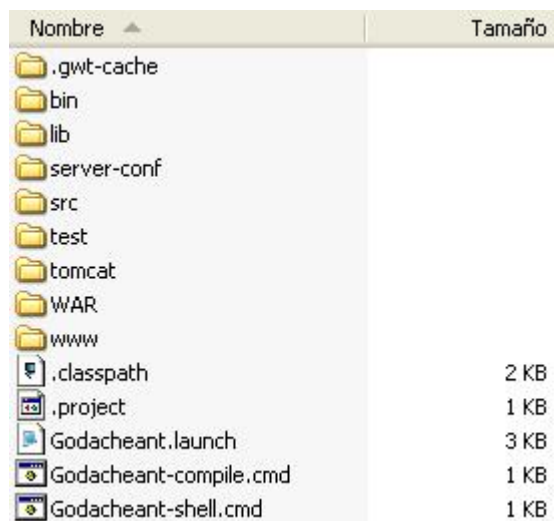
Godacheant

CARPETAS

- .gwt-cache
- bin
- lib
- server-conf
- src
- test
- tomcat
- WAR
- www

ARCHIVOS

- .classpath
- .project
- Godacheant.launch
- Godacheant-compile.cmd
- Godacheant-shell



Nombre	Tamaño
.gwt-cache	
bin	
lib	
server-conf	
src	
test	
tomcat	
WAR	
www	
.classpath	2 KB
.project	1 KB
Godacheant.launch	3 KB
Godacheant-compile.cmd	1 KB
Godacheant-shell.cmd	1 KB

bin

Ahí se localizan los *.class que se generan al compilar.

Para generar el WAR es necesario haber compilado para incluir los últimos *.class

lib

Están las librerías necesarias para poder ejecutar el proyecto durante el desarrollo en el eclipse. El desarrollo usa el 'gwt-dev-windows.jar' y 'gwt-user' **de la carpeta gwt-windows-1.4.60** y a la hora de hacer el .WAR se usan los que están en la carpeta lib junto al resto de librerías.

IMPORTANTE: los jar: 'gwt-dev-windows.jar' y 'gwt-user' que están en la carpeta lib se les ha extraído la carpeta 'javax' ya que es incompatible con el tomcat y sino entonces no se podría ejecutar en un tomcat externo. También se modifican los index.list que están en la carpeta META-INF y se quitan todos los directorios que están incluidos en la carpeta javax.

The screenshot shows the Eclipse IDE interface. On the left, the 'Project Explorer' shows a tree of projects and libraries. The 'lib' project is expanded, showing various JAR files and folders. On the right, the 'Properties' window is open for the selected file 'gwt-user', showing its type as 'HTML Document' and its size as 5,62 KB. The 'Content Assist' window is also visible, showing a list of files and folders with their sizes.

Nombre	Tamaño
gwt-windows-1.4.60	
commons-fileupload-1.2	52 KB
commons-io-1.3.1	82 KB
gwm	230 KB
gwt-dev-windows	9.414 KB
gwt-user	1.842 KB
gwt-widgets-0.1.5	247 KB
jdom	150 KB
jsr173_1.0_api	24 KB
mysql-connector-java-5.0.3-bin	482 KB
validarScormCam2003	1.381 KB
xbean	2.603 KB

Nombre	Tamaño
COPYING	13 KB
about	2 KB
gwt-benchmark-viewer	3.516 KB
gwt-dev-windows	9.815 KB
gwt-servlet	431 KB
gwt-user	1.937 KB
gwt-ll.dll	13 KB
swt-win32-3235.dll	352 KB
about	3 KB
COPYING	16 KB
index	6 KB
release_notes	53 KB
applicationCreator	
benchmarkViewer	
i18nCreator	
junitCreator	1 KB
projectCreator	1 KB
gwt-module	5 KB

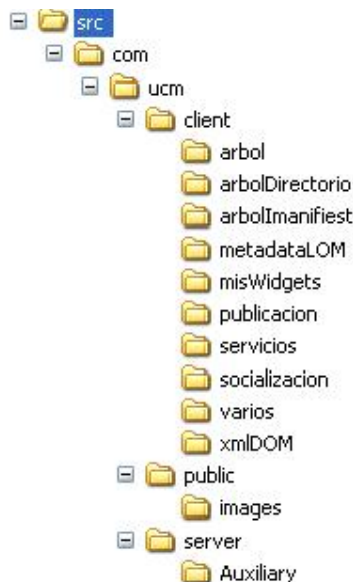
Tipo: HTML Document
Fecha de modificación: 2
Tamaño: 5,62 KB

server-conf

Contiene el web.xml necesario para que el Godacheant.war sepa donde estan localizados los servicios en el servidor.

En el web.xml hay que incluirle los servicios que utilice la aplicación. En caso de añadir más servicios habría que incluirlos en este archivo a la hora de generar el Godacheant.war ya que sino al ejecutarlo no se realizarían las llamadas a los servicios.

src



Contiene el código de la aplicación:

.\conexiones.properties: archivo de configuración para el acceso a la base de datos

.\com\ucm Tiene:

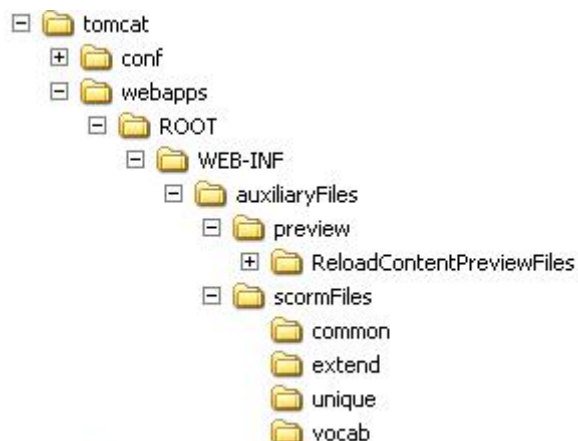
.\com\ucm\Godacheant.gwt.xml

.\com\ucm\client: posee la parte del cliente, no puede importar librerías porque falla, ya que para eso está la parte del servidor

.\com\ucm\public --> posee el `.\com\ucm\public\Godacheant.html`

.\com\ucm\server --> posee los servicios

tomcat



Subdirectorio importantes:

.\webapps\ROOT\WEB-INF ya que posee:

.\webapps\ROOT\WEB-INF\auxiliaryFiles

.\webapps\ROOT\WEB-INF\auxiliaryFiles\preview

.\webapps\ROOT\WEB-INF\auxiliaryFiles\scormFiles

para que así se cree estos subdirectos en el WAR.

WAR

En este directorio cuando realicemos los pasos que se verán en el punto 'Como crear el Godacheant.war' se colocara el Godacheant.war; siempre que creamos el Godacheant.war sera conveniente eliminar el Godacheant.war anterior.

Es el archivo que vamos a desplegar en el tomcat para ejecutarlo en un servidor externo y evitar los problemas de tener el tomcat embebido.

www

Posee los archivos compilados de 'html' y javaScript de nuestro codigo java.

Siempre que queramos un Godacheant.war nuevo con la ultima version hay que ejecutar el Godacheant y darle a la opcion 'Compile/Browse' para asi crear los archivos compilados de la ultima versión.

Como crear el Godacheant.war

1º Compilar el proyecto para generar los *.class de la ultima version, que se almacenan en la carpeta bin.

2º Ejecutar el proyecto y darle a la opcion 'Compile/Browse' para generar los nuevos archivos compilados de las clases java

3º Comprobamos que el .\server-conf\web.xml posee todos los servicios añadidos desde la ultima version.

4º Abrimos el simbolo de sistema, nos colocamos en la carpeta del Godacheant y usamos la siguiente linea de comandos:

```
.\lib\gwt-windows-1.4.60\projectCreator -ant Godacheant
```

Con esa linea de comandos obtenemos el .\Godacheant.ant.xml

5º Abrimos el .\Godacheant.ant.xml y le añadimos los siguiente después de algun target(no se le pueden añadir acentos ya que con algunos editores surgen problemas y luego no funciona):

```
//*****  
    <target name="compilewww" description="Compila el codigo cliente">  
        <exec executable="cmd.exe">  
            <arg line="/c Godacheant-compile.cmd" />  
        </exec>  
    </target>  
  
    <target name="war" description="Genera el war con el codigo servidor" depends="compile, compilewww">  
        <war destfile="./WAR/Godacheant.war" webxml="server-conf/web.xml">  
            <lib dir="lib">  
                <include name="**/*.jar" />  
            </lib>  
            <!-- Ubicacion de las clases ejecutables -->  
            <classes dir="bin" />  
            <!-- Recursos no-java que se incluiran en el fichero war -->  
            <fileset dir="www/com.ucm.Godacheant">  
                <include name="**/*.*" />  
            </fileset>  
            <fileset dir="tomcat/webapps/ROOT">  
                <include name="**/*.*" />  
            </fileset>  
        </war>  
    </target>  
//*****
```

y guardas el archivo.

6º Instalar el apache-ant (Ver 'instalacion del apache-ant-1.7.0')

7º Estando en la carpeta del proyecto en el simbolo de sistema, utilizar la linea de comandos (suponemos que ha instalado el apache-ant en el directorio que aparece en el manual de instalacion)

```
C:\apache-ant-1.7.0\bin\Ant -f Godacheant.ant.xml war
```

8º Se ha creado el Godacheant.war en la carpeta WAR

9º Copiar el Godacheant.war en el tomcat en la carpeta 'webapps' del tomcat sobre el que vayamos a desplegar

10º Ejecutar el tomcat (carpeta bin\startup (archivo por lotes))

11º Abrir explorador y escribir --> <http://localhost:8080/Godacheant/Godacheant.html>
(el 8080 depende de en que puerto este instalado el tomcat, en caso de tenerlo configurado al puerto 80 que es nuestro caso, no hace falta poner ':8080' y quedaría así: <http://localhost/Godacheant/Godacheant.html>).

Desplegar la aplicación en el Tomcat de forma manual sin tener que crear el War

Creamos la carpeta Godacheant en la carpeta del tomcat webapps.

Dentro de Godacheant copiamos los archivos de la carpeta www\com.ucm.Godacheant de la aplicación. También creamos la carpeta WEB-INF.

Dentro de WEB-INF creamos el web.xml, y las siguientes 3 carpetas

.\WEB-INF\auxiliaryFiles: necesario para crear un Nuevo objeto de aprendizaje

.\WEB-INF\classes: archivos java compilados (carpeta bin del proyecto)

.\WEB-INF\lib: librerías utilizadas en el proyecto con la particularidad de:

'gwt-dev-windows.jar' y 'gwt-user' se les ha extraído la carpeta 'javax' ya que es incompatible con el tomcat y sino entonces no se podría ejecutar en un tomcat externo. También se modifican los index.list que están en la carpeta META-INF y se quitan todos los directorios que están incluidos en la carpeta javax.

Estructura del Godacheant.war

.\archivos compilados de la carpeta www\com.ucm.Godacheant del proyecto

.\WEB-INF

dentro del .\WEB-INF

.\WEB-INF\classes: posee los archivos java compilados en html y JavaScript (carpeta bin del proyecto)

.\WEB-INF\lib: posee las librerías del proyecto pero 'gwt-dev-windows.jar' y 'gwt-user' se les ha extraído la carpeta 'javax' ya que es incompatible con el tomcat y sino entonces no se podría ejecutar en un tomcat externo. También se modifican los index.list que están en la carpeta META-INF y se quitan todos los directorios que están incluidos en la carpeta javax.

.\WEB-INF\auxiliaryFiles: posee los archivos para la creación del Nuevo LO

.\WEB-INF\web.xml: se ponen los servicios que posee la aplicación

Creación del validador a través del XmlBeans

Comando para crear el jar que sirve para validar el imsmanifest.xml en función del Scorm2004 extendido es el siguiente:

```
scomp -out nombreJar.jar imscp_v1p1.xsd lom.xsd adlnav_v1p3.xsd adlseq_v1p3.xsd  
adlcp_v1p3.xsd imsss_v1p0.xsd
```

Para que el comando funcione tienes que estar con el simbolo de sistemas en la carpeta \bin del xmlBeans y dentro de esta carpeta tener todos los archivos xsd que estan en la carpeta “Instalacion\xmlbeans-2.2.0\bin\xsd para validar modificados” que hay en el CD del proyecto que sirven para validar el imsmanifest, otra opción es poner las rutas de todos y no tiene que estar entonces eso en la carpeta \bin, eso es a elección del que se lo instale.

Como no se podía crear un jar que validara el Scorm2004 extendido con los xsd que se nos proporcionaron hubo que realizar unos cambios en algunos archivos que se mostrarán a continuación.

Los archivos modificados fueron los siguientes que estan localizados en “Instalacion\xmlbeans-2.2.0\bin\xsd para validar modificados”:

- Common/elementNamens.xsd
- Common/elementTypes.xsd
- Common/vocabTypes.xsd
- Common/vocabValues.xsd
- Vocab/strict.xsd
- Vocab/custom.xsd
- Unique/strict.xsd

Listado de la parte de código de los esquemas que se ha modificado

Common/elementNamens.xsd

```
<!-- 5.12 Cognitive Process -->
  <xs:group name="cognitiveProcess">
    <xs:sequence>
      <xs:element name="cognitiveProcess" type="cognitiveProcess"/>
    </xs:sequence>
  </xs:group>
<!-- 5.12 Cognitive Process -->
<!--xs:group name="cognitiveProcess">
  <xs:sequence>
    <xs:element name="cognitiveProcess" type="cognitiveProcess">
      <xs:unique name="cognitiveProcessUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group-->

<!-- 6 Rights -->
<xs:group name="rights">
  <xs:sequence>
    <xs:element name="rights" type="rights">
      <xs:unique name="rightsUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group>

<!-- 6.1 Cost -->
<xs:group name="cost">
  <xs:sequence>
    <xs:element name="cost" type="cost">
      <xs:unique name="costUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:group name="copyrightAndOtherRestrictions">
  <xs:sequence>
    <xs:element name="copyrightAndOtherRestrictions" type="copyrightAndOtherRestrictions">
      <xs:unique name="copyrightAndOtherRestrictionsUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group>

<!-- 6.3 Description -->
<xs:group name="description">
  <xs:sequence>
    <xs:element name="description" type="description"/>
  </xs:sequence>
</xs:group>
```

```

<!-- 6.4 Access -->
<xs:group name="access">
  <xs:sequence>
    <xs:element name="access" type="access">
      <xs:unique name="accessUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group>
<!-- 6.4.1 Access Type-->
<xs:group name="accessType">
  <xs:sequence>
    <xs:element name="accessType" type="accessType">
      <xs:unique name="accessTypeUnique">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group>
<!-- 6.4.2 Description
<xs:group name="description">
  <xs:sequence>
    <xs:element name="description" type="description">
      <xs:unique name="description">
        <xs:selector xpath="*/>
        <xs:field xpath="@uniqueElementName"/>
      </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:group-->

```

Common/elementTypes.xsd

```
<!-- 5 Educational -->
<xs:complexType name="educational">
  <xs:sequence>
    <xs:group ref="interactivityType" minOccurs="0"/>
    <xs:group ref="learningResourceType" maxOccurs="unbounded"/>
    <xs:group ref="interactivityLevel" minOccurs="0"/>
    <xs:group ref="semanticDensity" minOccurs="0"/>
    <xs:group ref="intendedEndUserRole" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="context" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="typicalAgeRange" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="difficulty" minOccurs="0"/>
    <xs:group ref="typicalLearningTime" minOccurs="0"/>
    <xs:group ref="descriptionUnbounded" minOccurs="0" maxOccurs="unbounded"/>
    <xs:group ref="language" maxOccurs="unbounded"/>
    <xs:group ref="cognitiveProcess" minOccurs="0" maxOccurs="unbounded"/>
    <!--xs:group ref="ex:customElements"/-->
  </xs:sequence>
  <xs:attributeGroup ref="ag:educational"/>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<!-- 5.1 Interactivity Type -->
<xs:complexType name="interactivityType">
  <xs:complexContent>
    <xs:extension base="interactivityTypeVocab">
      <xs:attributeGroup ref="ag:interactivityType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.2 Learning Resource Type -->
<xs:complexType name="learningResourceType">
  <xs:complexContent>
    <xs:extension base="learningResourceTypeVocab">
      <xs:attributeGroup ref="ag:learningResourceType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.3 Interactivity Level -->
<xs:complexType name="interactivityLevel">
  <xs:complexContent>
    <xs:extension base="interactivityLevelVocab">
      <xs:attributeGroup ref="ag:interactivityLevel"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.4 Semantic Density -->
<xs:complexType name="semanticDensity">
  <xs:complexContent>
    <xs:extension base="semanticDensityVocab">
      <xs:attributeGroup ref="ag:semanticDensity"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.5 Intended End User Role -->
<xs:complexType name="intendedEndUserRole">
  <xs:complexContent>
    <xs:extension base="intendedEndUserRoleVocab">
      <xs:attributeGroup ref="ag:intendedEndUserRole"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

<!-- 5.6 Context -->
<xs:complexType name="context">
  <xs:complexContent>
    <xs:extension base="contextVocab">
      <xs:attributeGroup ref="ag:context"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.7 Typical Age Range -->
<xs:complexType name="typicalAgeRange">
  <xs:complexContent>
    <xs:extension base="LanguageString">
      <xs:attributeGroup ref="ag:typicalAgeRange"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.8 Difficulty -->
<xs:complexType name="difficulty">
  <xs:complexContent>
    <xs:extension base="difficultyVocab">
      <xs:attributeGroup ref="ag:difficulty"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.9 Typical Learning Time -->
<xs:complexType name="typicalLearningTime">
  <xs:complexContent>
    <xs:extension base="Duration">
      <xs:attributeGroup ref="ag:typicalLearningTime"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 5.10 Description
<xs:complexType name="description">
  <xs:complexContent>
    <xs:extension base="LanguageString">
      <xs:attributeGroup ref="ag:description"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType> -->

<!-- 5.11 Language
<xs:complexType name="language">
  <xs:simpleContent>
    <xs:extension base="LanguageId">
      <xs:attributeGroup ref="ag:language"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType> -->

  <!-- 5.12 Cognitive Process -->
  <xs:complexType name="cognitiveProcess">
    <xs:complexContent>
      <xs:extension base="cognitiveProcessVocab">
        <xs:attributeGroup ref="ag:cognitiveProcess"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

<!-- 6 Rights -->
<xs:complexType name="rights">
  <xs:sequence>
    <xs:group ref="cost" minOccurs="0"/>
    <xs:group ref="copyrightAndOtherRestrictions"/>
    <xs:group ref="description" minOccurs="0"/>
    <xs:group ref="access"/>
    <!--xs:group ref="ex:customElements"/-->
  </xs:sequence>
  <xs:attributeGroup ref="ag:rights"/>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<!-- 6.1 Cost -->
<xs:complexType name="cost">
  <xs:complexContent>
    <xs:extension base="costVocab">
      <xs:attributeGroup ref="ag:cost"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:complexType name="copyrightAndOtherRestrictions">
  <xs:complexContent>
    <xs:extension base="copyrightAndOtherRestrictionsVocab">
      <xs:attributeGroup ref="ag:copyrightAndOtherRestrictions"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 6.3 Description -->
<xs:complexType name="description">
  <xs:complexContent>
    <xs:extension base="LanguageString">
      <xs:attributeGroup ref="ag:description"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 6.4 Access -->
<xs:complexType name="access">
  <xs:sequence>
    <xs:group ref="accessType"/>
    <xs:group ref="description"/>
    <!--xs:group ref="ex:customElements"/-->
  </xs:sequence>
  <xs:attributeGroup ref="ag:access"/>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<!-- 6.4.1 Access Type -->
<xs:complexType name="accessType">
  <xs:complexContent>
    <xs:extension base="accessTypeVocab">
      <xs:attributeGroup ref="ag:accessType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- 6.4.2 Description -->
<xs:complexType name="description">
  <xs:complexContent>
    <xs:extension base="LanguageString">
      <xs:attributeGroup ref="ag:description"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType-->

```

Common/vocabTypes.xsd

```
<!-- 5.1 Interactivity Type -->
<xs:complexType name="interactivityTypeVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="interactivityTypeValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="interactivityTypeValue">
  <xs:simpleContent>
    <xs:extension base="voc:interactivityType">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.2 Learning Resource Type -->
<xs:complexType name="learningResourceTypeVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="learningResourceTypeValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="learningResourceTypeValue">
  <xs:simpleContent>
    <xs:extension base="voc:learningResourceType">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.3 Interactivity Level -->
<xs:complexType name="interactivityLevelVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="interactivityLevelValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="interactivityLevelValue">
  <xs:simpleContent>
    <xs:extension base="voc:interactivityLevel">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.4 Semantic Density -->
<xs:complexType name="semanticDensityVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="semanticDensityValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="semanticDensityValue">
  <xs:simpleContent>
    <xs:extension base="voc:semanticDensity">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

```

    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.5 Intended End User Role -->
<xs:complexType name="intendedEndUserRoleVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="intendedEndUserRoleValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="intendedEndUserRoleValue">
  <xs:simpleContent>
    <xs:extension base="voc:intendedEndUserRole">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.6 Context -->
<xs:complexType name="contextVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="contextValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="contextValue">
  <xs:simpleContent>
    <xs:extension base="voc:context">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 5.8 Difficulty -->
<xs:complexType name="difficultyVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="difficultyValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="difficultyValue">
  <xs:simpleContent>
    <xs:extension base="voc:difficulty">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

<!-- 5.12 Cognitive Process -->
<xs:complexType name="cognitiveProcessVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="cognitiveProcessValue"/>
    <!--xs:group ref="ex:customElements"/-->
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>
<xs:complexType name="cognitiveProcessValue">
  <xs:simpleContent>
    <xs:extension base="voc:cognitiveProcess">
      <xs:attributeGroup ref="ag:value"/>
      <!--xs:attributeGroup ref="ex:customAttributes"/-->
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<!-- 6.1 Cost -->
<xs:complexType name="costVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="costValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="costValue">
  <xs:simpleContent>
    <xs:extension base="voc:cost">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:complexType name="copyrightAndOtherRestrictionsVocab">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="source" type="sourceValue"/>
    <xs:element name="value" type="copyrightAndOtherRestrictionsValue"/>
    <xs:group ref="ex:customElements"/>
  </xs:choice>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>

<xs:complexType name="copyrightAndOtherRestrictionsValue">
  <xs:simpleContent>
    <xs:extension base="voc:copyrightAndOtherRestrictions">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<!-- 6.4.1 Access Type -->
<xs:complexType name="accessTypeVocab">
  <xs:sequence>
    <xs:element name="source" type="sourceValue" minOccurs="1" maxOccurs="1"/>
    <xs:element name="value" type="accessTypeValue" minOccurs="1" maxOccurs="1"/>
    <!--xs:group ref="ex:customElements"/-->
  </xs:sequence>
  <xs:attributeGroup ref="ex:customAttributes"/>
</xs:complexType>
<xs:complexType name="accessTypeValue">
  <xs:simpleContent>
    <xs:extension base="voc:accessType">
      <xs:attributeGroup ref="ag:value"/>
      <xs:attributeGroup ref="ex:customAttributes"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

Common/vocabValues.xsd

En este archivo había que cambiar prácticamente todo ya que había vocabulario nuevo en casi todas las estructuras, así que listo todo el archivo.

```
<xs:schema targetNamespace="http://ltsc.ieee.org/xsd/LOM"
  xmlns="http://ltsc.ieee.org/xsd/LOM"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="IEEE LTSC LOM XML 1.0">

  <xs:annotation>
    <xs:documentation>
      This work is licensed under the Creative Commons Attribution-ShareAlike
      License. To view a copy of this license, see the file license.txt,
      visit http://creativecommons.org/licenses/by-sa/2.0 or send a letter to
      Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
    </xs:documentation>

    <xs:documentation>
      This component schema definition provides global type declarations for the standard
      LOMv1.0 vocabulary tokens for those LOM data elements whose values are taken from
      a Vocabulary data type.

      This component schema definition defines the standard vocabulary value
      declarations as defined in the LOMv1.0 base schema. These vocabulary
      value declarations are used in conjunction with both vocab/custom.xsd and
      vocab/strict.xsd.
    </xs:documentation>
  </xs:annotation>

  <!-- LOMv1.0 base schema vocabulary source and value declarations -->

  <!-- Source -->
  <xs:simpleType name="sourceValues">
    <xs:restriction base="xs:token">
      <xs:enumeration value="LOM-ESv1.0"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- 1.7 Structure -->
  <xs:simpleType name="structureValues">
    <xs:restriction base="xs:token">
      <xs:enumeration value="atomic"/>
      <xs:enumeration value="collection"/>
      <xs:enumeration value="networked"/>
      <xs:enumeration value="hierarchical"/>
      <xs:enumeration value="linear"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- 1.8 Aggregation Level -->
  <xs:simpleType name="aggregationLevelValues">
    <xs:restriction base="xs:token">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
      <xs:enumeration value="3"/>
      <xs:enumeration value="4"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- 2.2 Status -->
  <xs:simpleType name="statusValues">
    <xs:restriction base="xs:token">
      <xs:enumeration value="draft"/>
      <xs:enumeration value="final"/>
      <xs:enumeration value="revised"/>
      <xs:enumeration value="unavailable"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- 2.3.1 Role -->
```

```

<xs:simpleType name="roleValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="author"/>
    <xs:enumeration value="publisher"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="initiator"/>
    <xs:enumeration value="terminator"/>
    <xs:enumeration value="validator"/>
    <xs:enumeration value="editor"/>
    <xs:enumeration value="graphical designer"/>
    <xs:enumeration value="technical implementer"/>
    <xs:enumeration value="content provider"/>
    <xs:enumeration value="technical validator"/>
    <xs:enumeration value="educational validator"/>
    <xs:enumeration value="script writer"/>
    <xs:enumeration value="instructional designer"/>
    <xs:enumeration value="subject matter expert"/>
  </xs:restriction>
</xs:simpleType>

<!-- 3.2.1 Role -->
<xs:simpleType name="roleMetaValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="creator"/>
    <xs:enumeration value="validator"/>
  </xs:restriction>
</xs:simpleType>

<!-- 4.4.1.1 Type -->
<xs:simpleType name="typeValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="operating system"/>
    <xs:enumeration value="browser"/>
  </xs:restriction>
</xs:simpleType>

<!-- 4.4.1.2 Name -->
<xs:simpleType name="nameValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="pc-dos"/>
    <xs:enumeration value="ms-windows"/>
    <xs:enumeration value="macos"/>
    <xs:enumeration value="linux"/>
    <xs:enumeration value="unix"/>
    <xs:enumeration value="multi-so"/>
    <xs:enumeration value="multi-os"/>
    <xs:enumeration value="none"/>
    <xs:enumeration value="any"/>
    <xs:enumeration value="mozilla firefox"/>
    <xs:enumeration value="netscape communicator"/>
    <xs:enumeration value="ms-internet explorer"/>
    <xs:enumeration value="opera"/>
    <xs:enumeration value="amaya"/>
  </xs:restriction>
</xs:simpleType>

<!-- 5.1 Interactivity Type -->
<xs:simpleType name="interactivityTypeValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="active"/>
    <xs:enumeration value="expositive"/>
    <xs:enumeration value="mixed"/>
  </xs:restriction>
</xs:simpleType>

```

```

<!-- 5.2 Learning Resource Type -->
<xs:simpleType name="learningResourceTypeValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="photograph"/>
    <xs:enumeration value="illustration"/>
    <xs:enumeration value="video"/>
    <xs:enumeration value="animation"/>
    <xs:enumeration value="music"/>
    <xs:enumeration value="sound effect"/>
    <xs:enumeration value="voice-over"/>
    <xs:enumeration value="compound audio"/>
    <xs:enumeration value="narrative text"/>
    <xs:enumeration value="hypertext"/>
    <xs:enumeration value="computer graphics"/>
    <xs:enumeration value="integrated media"/>
    <xs:enumeration value="database"/>
    <xs:enumeration value="table"/>
    <xs:enumeration value="graph"/>
    <xs:enumeration value="conceptual map"/>
    <xs:enumeration value="navigation map"/>
    <xs:enumeration value="multimedia presentation"/>
    <xs:enumeration value="tutorial"/>
    <xs:enumeration value="digital dictionary"/>
    <xs:enumeration value="digital encyclopaedia"/>
    <xs:enumeration value="digital periodical publication"/>
    <xs:enumeration value="thematic or corporate webs/web portals"/>
    <xs:enumeration value="wiki"/>
    <xs:enumeration value="weblog"/>
    <xs:enumeration value="multimedia creation/edition tool"/>
    <xs:enumeration value="web design tool"/>
    <xs:enumeration value="office tool"/>
    <xs:enumeration value="programming tool"/>
    <xs:enumeration value="information/knowledgeanalysis/organization tool"/>
    <xs:enumeration value="process/procedure supporting tools"/>
    <xs:enumeration value="individual/cooperative/collaborative learning/working management
tool"/>
    <xs:enumeration value="multimedia creation/edition service"/>
    <xs:enumeration value="web design service"/>
    <xs:enumeration value="office service"/>
    <xs:enumeration value="programming service"/>
    <xs:enumeration value="information/knowledgeanalysis/organization service"/>
    <xs:enumeration value="process/procedure supporting service"/>
    <xs:enumeration value="individual/cooperative/collaborative learning/working management
service"/>
    <xs:enumeration value="guided reading"/>
    <xs:enumeration value="master class"/>
    <xs:enumeration value="textual-image analysis"/>
    <xs:enumeration value="discussion activity"/>
    <xs:enumeration value="closed exercise or problem"/>
    <xs:enumeration value="contextualized case problem"/>
    <xs:enumeration value="open problem"/>
    <xs:enumeration value="real or virtual learning environment"/>
    <xs:enumeration value="didactic game"/>
    <xs:enumeration value="webquest"/>
    <xs:enumeration value="experiment"/>
    <xs:enumeration value="real project"/>
    <xs:enumeration value="simulation"/>
    <xs:enumeration value="questionnaire"/>
    <xs:enumeration value="exam"/>
    <xs:enumeration value="self assessment"/>
  </xs:restriction>
</xs:simpleType>

<!-- 5.3 Interactivity Level -->
<xs:simpleType name="interactivityLevelValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="very low"/>
    <xs:enumeration value="low"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="high"/>
    <xs:enumeration value="very high"/>
  </xs:restriction>
</xs:simpleType>

```

```

<!-- 5.4 Semantic Density -->
<xs:simpleType name="semanticDensityValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="very low"/>
    <xs:enumeration value="low"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="high"/>
    <xs:enumeration value="very high"/>
  </xs:restriction>
</xs:simpleType>

<!-- 5.5 Intended End User Role -->
<xs:simpleType name="intendedEndUserRoleValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="learner"/>
    <xs:enumeration value="special needs learner"/>
    <xs:enumeration value="gifted learner"/>
    <xs:enumeration value="learners late integration into the education system"/>
    <xs:enumeration value="learner with other specific educational support needs"/>
    <xs:enumeration value="general public"/>
    <xs:enumeration value="individual"/>
    <xs:enumeration value="group"/>
    <xs:enumeration value="teacher"/>
    <xs:enumeration value="tutor"/>
    <xs:enumeration value="family"/>
    <xs:enumeration value="information scientist"/>
    <xs:enumeration value="computer scientist"/>
    <xs:enumeration value="manager"/>
    <xs:enumeration value="Education expert"/>
    <xs:enumeration value="subject matter expert"/>
  </xs:restriction>
</xs:simpleType>

<!-- 5.6 Context -->
<xs:simpleType name="contextValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="classroom"/>
    <xs:enumeration value="laboratory"/>
    <xs:enumeration value="real environment"/>
    <xs:enumeration value="home"/>
    <xs:enumeration value="mixed"/>
    <xs:enumeration value="teacher"/>
    <xs:enumeration value="family"/>
    <xs:enumeration value="tutor"/>
    <xs:enumeration value="schoolmate"/>
    <xs:enumeration value="independent"/>
    <xs:enumeration value="blended"/>
    <xs:enumeration value="presencial"/>
    <xs:enumeration value="face to face"/>
    <xs:enumeration value="distance"/>
  </xs:restriction>
</xs:simpleType>

<!-- 5.8 Difficulty -->
<xs:simpleType name="difficultyValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="very easy"/>
    <xs:enumeration value="easy"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="difficult"/>
    <xs:enumeration value="very difficult"/>
  </xs:restriction>
</xs:simpleType>

```

```

<!-- 5.12 Cognitive Process -->
<xs:simpleType name="cognitiveProcessValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="analyse"/>
    <xs:enumeration value="implement"/>
    <xs:enumeration value="collaborate"/>
    <xs:enumeration value="compare"/>
    <xs:enumeration value="share"/>
    <xs:enumeration value="compete"/>
    <xs:enumeration value="understand"/>
    <xs:enumeration value="prove"/>
    <xs:enumeration value="communicate"/>
    <xs:enumeration value="contextualize"/>
    <xs:enumeration value="control"/>
    <xs:enumeration value="cooperate"/>
    <xs:enumeration value="create"/>
    <xs:enumeration value="decide"/>
    <xs:enumeration value="define"/>
    <xs:enumeration value="describe"/>
    <xs:enumeration value="discuss"/>
    <xs:enumeration value="design"/>
    <xs:enumeration value="self assessment"/>
    <xs:enumeration value="explain"/>
    <xs:enumeration value="extrapolate"/>
    <xs:enumeration value="innovate"/>
    <xs:enumeration value="investigate"/>
    <xs:enumeration value="judge"/>
    <xs:enumeration value="motivate"/>
    <xs:enumeration value="observe"/>
    <xs:enumeration value="organize"/>
    <xs:enumeration value="organize oneself"/>
    <xs:enumeration value="plan"/>
    <xs:enumeration value="practise"/>
    <xs:enumeration value="produce"/>
    <xs:enumeration value="recognize"/>
    <xs:enumeration value="remember"/>
    <xs:enumeration value="write up"/>
    <xs:enumeration value="consider"/>
    <xs:enumeration value="connect"/>
    <xs:enumeration value="represent"/>
    <xs:enumeration value="solve"/>
    <xs:enumeration value="simulate"/>
    <xs:enumeration value="summarize"/>
    <xs:enumeration value="value"/>
  </xs:restriction>
</xs:simpleType>

<!-- 6.1 Cost -->
<xs:simpleType name="costValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="yes"/>
    <xs:enumeration value="no"/>
  </xs:restriction>
</xs:simpleType>

```

```

<!-- 6.2 Copyright and Other Restrictions -->
<xs:simpleType name="copyrightAndOtherRestrictionsValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="proprietary license"/>
    <xs:enumeration value="free software license EUPL"/>
    <xs:enumeration value="free software license GPL"/>
    <xs:enumeration value="dual free content license GPL and EUPL"/>
    <xs:enumeration value="other free software licenses"/>
    <xs:enumeration value="public domain"/>
    <xs:enumeration value="not appropriate"/>
    <xs:enumeration value="intellectual property license"/>
    <xs:enumeration value="creative commons: attribution"/>
    <xs:enumeration value="creative commons: attribution - non derived work"/>
    <xs:enumeration value="creative commons: attribution - non derived work - non commercial"/>
    <xs:enumeration value="creative commons: attribution - non commercial"/>
    <xs:enumeration value="creative commons: attribution - non commercial - share alike"/>
    <xs:enumeration value="creative commons: attribution - share alike"/>
    <xs:enumeration value="license GFDL"/>
  </xs:restriction>
</xs:simpleType>

  <!--6.4.1 Access Type Values-->
  <xs:simpleType name="accessTypeValues">
    <xs:restriction base="xs:token">
      <xs:enumeration value="universal"/>
      <xs:enumeration value="non-universal"/>
    </xs:restriction>
  </xs:simpleType>

<!-- 7.1 Kind -->
<xs:simpleType name="kindValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="ispartof"/>
    <xs:enumeration value="haspart"/>
    <xs:enumeration value="isversionof"/>
    <xs:enumeration value="hasversion"/>
    <xs:enumeration value="isformatof"/>
    <xs:enumeration value="hasformat"/>
    <xs:enumeration value="references"/>
    <xs:enumeration value="isreferencedby"/>
    <xs:enumeration value="isbasedon"/>
    <xs:enumeration value="isbasisfor"/>
    <xs:enumeration value="requires"/>
    <xs:enumeration value="isrequiredby"/>
  </xs:restriction>
</xs:simpleType>

<!-- 9.1 Purpose -->
<xs:simpleType name="purposeValues">
  <xs:restriction base="xs:token">
    <xs:enumeration value="discipline"/>
    <xs:enumeration value="idea"/>
    <xs:enumeration value="prerequisite"/>
    <xs:enumeration value="educational objective"/>
    <xs:enumeration value="accessibility restrictions"/>
    <xs:enumeration value="educational level"/>
    <xs:enumeration value="skill level"/>
    <xs:enumeration value="security level"/>
    <xs:enumeration value="competency"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Vocab/strict.xsd

```
<!-- 5.1 Interactivity Type -->
<xs:simpleType name="interactivityType">
  <xs:union memberTypes="lom:interactivityTypeValues"/>
</xs:simpleType>

<!-- 5.2 Learning Resource Type -->
<xs:simpleType name="learningResourceType">
  <xs:union memberTypes="lom:learningResourceTypeValues"/>
</xs:simpleType>

<!-- 5.3 Interactivity Level -->
<xs:simpleType name="interactivityLevel">
  <xs:union memberTypes="lom:interactivityLevelValues"/>
</xs:simpleType>

<!-- 5.4 Semantic Density -->
<xs:simpleType name="semanticDensity">
  <xs:union memberTypes="lom:semanticDensityValues"/>
</xs:simpleType>

<!-- 5.5 Intended End User Role -->
<xs:simpleType name="intendedEndUserRole">
  <xs:union memberTypes="lom:intendedEndUserRoleValues"/>
</xs:simpleType>

<!-- 5.6 Context -->
<xs:simpleType name="context">
  <xs:union memberTypes="lom:contextValues"/>
</xs:simpleType>

<!-- 5.8 Difficulty -->
<xs:simpleType name="difficulty">
  <xs:union memberTypes="lom:difficultyValues"/>
</xs:simpleType>

<!-- 5.12 Cognitive Process -->
<xs:simpleType name="cognitiveProcess">
  <xs:union memberTypes="lom:cognitiveProcessValues"/>
</xs:simpleType>

<!-- 6.1 Cost -->
<xs:simpleType name="cost">
  <xs:union memberTypes="lom:costValues"/>
</xs:simpleType>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:simpleType name="copyrightAndOtherRestrictions">
  <xs:union memberTypes="lom:copyrightAndOtherRestrictionsValues"/>
</xs:simpleType>

<!-- 6.4.1 Access Type -->
<xs:simpleType name="accessType">
  <xs:union memberTypes="lom:accessTypeValues"/>
</xs:simpleType>
```

Vocab/custom.xsd

```
<!-- 5.1 Interactivity Type -->
<xs:simpleType name="interactivityType">
  <xs:union memberTypes="lom:interactivityTypeValues lx:interactivityTypeValues"/>
</xs:simpleType>

<!-- 5.2 Learning Resource Type -->
<xs:simpleType name="learningResourceType">
  <xs:union memberTypes="lom:learningResourceTypeValues lx:learningResourceTypeValues"/>
</xs:simpleType>

<!-- 5.3 Interactivity Level -->
<xs:simpleType name="interactivityLevel">
  <xs:union memberTypes="lom:interactivityLevelValues lx:interactivityLevelValues"/>
</xs:simpleType>

<!-- 5.4 Semantic Density -->
<xs:simpleType name="semanticDensity">
  <xs:union memberTypes="lom:semanticDensityValues lx:semanticDensityValues"/>
</xs:simpleType>

<!-- 5.5 Intended End User Role -->
<xs:simpleType name="intendedEndUserRole">
  <xs:union memberTypes="lom:intendedEndUserRoleValues lx:intendedEndUserRoleValues"/>
</xs:simpleType>

<!-- 5.6 Context -->
<xs:simpleType name="context">
  <xs:union memberTypes="lom:contextValues lx:contextValues"/>
</xs:simpleType>

<!-- 5.8 Difficulty -->
<xs:simpleType name="difficulty">
  <xs:union memberTypes="lom:difficultyValues lx:difficultyValues"/>
</xs:simpleType>

<!-- 5.12 Cognitive Process -->
  <xs:simpleType name="cognitiveProcess">
    <xs:union memberTypes="lom:cognitiveProcessValues lx:cognitiveProcessValues"/>
  </xs:simpleType>

<!-- 6.1 Cost -->
<xs:simpleType name="cost">
  <xs:union memberTypes="lom:costValues lx:costValues"/>
</xs:simpleType>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:simpleType name="copyrightAndOtherRestrictions">
  <xs:union memberTypes="lom:copyrightAndOtherRestrictionsValues lx:copyrightAndOtherRestrictionsValues"/>
</xs:simpleType>
```

Unique/strict.xsd

```
<!-- 5 Educational -->
<xs:attributeGroup name="educational"/>

<!-- 5.1 Interactivity Type -->
<xs:attributeGroup name="interactivityType">
  <xs:attribute name="uniqueElementName" fixed="interactivityType"/>
</xs:attributeGroup>

<!-- 5.2 Learning Resource Type -->
<xs:attributeGroup name="learningResourceType"/>

<!-- 5.3 Interactivity Level -->
<xs:attributeGroup name="interactivityLevel">
  <xs:attribute name="uniqueElementName" fixed="interactivityLevel"/>
</xs:attributeGroup>

<!-- 5.4 Semantic Density -->
<xs:attributeGroup name="semanticDensity">
  <xs:attribute name="uniqueElementName" fixed="semanticDensity"/>
</xs:attributeGroup>

<!-- 5.5 Intended End User Role -->
<xs:attributeGroup name="intendedEndUserRole"/>

<!-- 5.6 Context -->
<xs:attributeGroup name="context"/>

<!-- 5.7 Typical Age Range -->
<xs:attributeGroup name="typicalAgeRange"/>

<!-- 5.8 Difficulty -->
<xs:attributeGroup name="difficulty">
  <xs:attribute name="uniqueElementName" fixed="difficulty"/>
</xs:attributeGroup>

<!-- 5.9 Typical Learning Time -->
<xs:attributeGroup name="typicalLearningTime">
  <xs:attribute name="uniqueElementName" fixed="typicalLearningTime"/>
</xs:attributeGroup>

<!-- 5.10 Description
<xs:attributeGroup name="description"/> -->

<!-- 5.11 Language
<xs:attributeGroup name="language"/> -->
<!-- 5.1 Interactivity Type -->

<!-- 5.12 Cognitive Process -->
<xs:attributeGroup name="cognitiveProcess"/>

<!--xs:attributeGroup name="cognitiveProcess">
  <xs:attribute name="uniqueElementName" fixed="cognitiveProcess"/>
</xs:attributeGroup-->

<!-- 6 Rights -->
<xs:attributeGroup name="rights">
  <xs:attribute name="uniqueElementName" fixed="rights"/>
</xs:attributeGroup>

<!-- 6.1 Cost -->
<xs:attributeGroup name="cost">
  <xs:attribute name="uniqueElementName" fixed="cost"/>
</xs:attributeGroup>

<!-- 6.2 Copyright and Other Restrictions -->
<xs:attributeGroup name="copyrightAndOtherRestrictions">
  <xs:attribute name="uniqueElementName" fixed="copyrightAndOtherRestrictions"/>
</xs:attributeGroup>
```

```
<!-- 6.3 Description -->
<xs:attributeGroup name="description">
  <xs:attribute name="uniqueElementName" fixed="description"/>
</xs:attributeGroup>
```

```
<!-- 6.4 Access -->
<xs:attributeGroup name="access">
  <xs:attribute name="uniqueElementName" fixed="access" type="xs:string"/>
</xs:attributeGroup>
```

```
<!-- 6.4.1 Access Type-->
<xs:attributeGroup name="accessType">
  <xs:attribute name="uniqueElementName" fixed="accessType" type="xs:string"/>
</xs:attributeGroup>
```

```
<!-- 6.4.2 Description
<xs:attributeGroup name="description">
  <xs:attribute name="uniqueElementName" fixed="description"/>
</xs:attributeGroup-->
```

BIBLIOGRAFÍA

Manuales de consulta

- SCORM 2004 3rd Edition DocSuite de ADL
 - SCORM 2004 3rd Edition Overview
 - SCORM 2004 3rd Edition Content Aggregation Model (CAM)
 - SCORM 2004 3rd Edition Run-Time Environment (RTE)
 - SCORM 2004 3rd Edition Sequencing and Navigation (SN)
- GWT Java AJAX Programming - Prabhakar Chaganti
- GWT in action – Robert Hanson y Adam Tacy
- Web 2.0 -- Advanced AJAX Applications with the Google Web Toolkit
- Google Web Toolkit, Taking the pain out of Ajax – Ed Burnette

Enlaces consultados

- GWT: <http://code.google.com/webtoolkit/>
- GWT User Interface Documentation:
<http://code.google.com/webtoolkit/documentation/com.google.gwt.doc.DeveloperGuide.UserInterface.html>
- MySql: <http://dev.mysql.com/doc/refman/5.0/es/index.html>.
<http://www.mysql.com/>
- XmlBeans: <http://xmlbeans.apache.org/>

AUTORIZACIÓN DE DIFUSIÓN

Los alumnos abajo firmantes Alberto Álvarez Toledano con DNI 50546596-W, David Andrés Ferreras con DNI 47222791-L y Gonzalo Sánchez de Ocaña Rengifo con DNI 50756086-P, autorizan a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Madrid, a 4 de julio de 2008

Firmas