

20/06/2009

# OPENCMS

**Realizado por Luis Barrio de Alarcón**  
**Alumno de 2º Administración de Sistemas Informáticos**  
**Instituto Gonzalo Nazareno Dos Hermanas**

## INDICE

---

<b>1. INTRODUCCIÓN A OPENCMS .....</b>	<b>6</b>
1.1. CONCEPTO .....	6
1.2. FUNCIONAMIENTO DE OPENCMS .....	6
1.3. CARACTERÍSTICAS GENERALES .....	7
1.4. CARACTERÍSTICAS TÉCNICAS Y FUNCIONALES .....	8
<b>2. INSTALACIÓN DE OPENCMS .....</b>	<b>8</b>
2.1. INSTALACIÓN DEL JDK .....	8
2.2. INSTALACIÓN DE TOMCAT .....	9
2.3. INSTALACIÓN DE MYSQL.....	9
2.4. DESPLIEGUE DE OPENCMS .....	10
2.5. CONSIDERACIONES DE SEGURIDAD.....	11
2.5.1. CAMBIO DE LA CONTRASEÑA DE OPENCMS.....	11
2.5.2. CAMBIO DE LA CONTRASEÑA DE ACCESO A MYSQL.....	11
2.6. EJECUCIÓN DEL ASISTENTE .....	11
2.6.1. LICENCIA .....	11
2.6.2. COMPONENTES DE OPENCMS.....	12
2.6.3. CONEXIÓN CON BASE DE DATOS.....	13
2.7. VISTA PREVIA DE OPENCMS.....	19
<b>3. INTERFAZ DE OPENCMS .....</b>	<b>21</b>
3.1. ELEMENTOS COMUNES .....	21
3.1.1. PROYECTO.....	21
3.1.2. PUBLICACIÓN PROYECTO.....	22
3.1.3. VISTA ACTUAL .....	22
3.1.4. RECARGAR .....	22
3.1.5. PREFERENCIAS .....	22
3.1.5.1. WORKPLACE .....	2
3.1.5.2. EXPLORADOR.....	23
3.1.5.3. DIALOGOS.....	23
3.1.5.4. EDITORES.....	24
3.1.5.5. DATOS USUARIO.....	25

---

---

3.1.6. SALIR .....	25
3.2. VISTA ADMINISTRACION.....	26
3.2.1. ADMINISTRACIÓN PROYECTOS.....	26
3.2.1.1. NUEVO PROYECTO.....	26
3.2.1.2. PROYECTOS ACTUALES.....	26
3.2.1.3. HISTORIAL DEL PROYECTO.....	26
3.2.1.4. GESTIONAR PROPIEDADES DEL HISTORIAL.....	27
3.3. USUARIOS Y GRUPOS.....	27
3.2.1.GESTIÓN DE USUARIOS.....	27
3.3.2. GESTION DE GRUPOS.....	27
3.4. GESTION DE BASE DE DATOS... ..	2
3.5. GESTION DE MODULOS .....	2
3.6. HISTORIAL DE FICHEROS .....	2
3.7. VALIDACION DE ENLACES .....	2
3.8. GALERIAS .....	2
3.8.1. GALERIA DE DESCARGAS .....	30
3.8.2. GALERIA HTML.....	30
3.8.3. GALERIA DE IMAGENES .....	30
3.8.4. GALERIA DE ENLACES EXTERNOS.....	30
3.9. GESTION DE BUSQUEDAS .....	30
3.10. TAREAS PROGRAMADAS.....	30
3.11. ADMINISTRACION DE FLEX CACHE .....	31
3.12. HERRAMIENTA WORKSPACE.....	31
<b>4. VISTA DE EXPLORADOR.....</b>	<b>32</b>
4.1. DETALLES DE LA VISTA DEL EXPLORADOR .....	32
4.2. OPERACIONES SOBRE LA CARPETA ACTUAL.....	23
4.2.1. VOLVER.....	33
4.2.2. ARRIBA.....	33
4.2.3. BUSCAR .....	33
4.2.4. NUEVO.....	33
4.3. OPERACIONES SOBRE LOS RECURSOS DEL VFS .....	34

---

---

<b>5. WEB DE EJEMPLO .....</b>	<b>35</b>
5.1. ESTRUCTURA DE LA WEB .....	36
5.2. ESTRUCTURA DEL MÓDULO .....	36
5.3. PLANTILLA XML.....	36
5.4. PLANTILLA JSP .....	37
5.5. PÁGINA WEB .....	37
<b>6. USO DE JSP EN OPENCMS .....</b>	<b>38</b>
6.1. PROCESAMIENTO DE JSP EN OPENCMS .....	2
6.2. ESTILOS DE SINTAXIS EN JSP .....	2
<b>7. LIBRERÍA DE ETIQUETAS DE OPENCMS.....</b>	<b>39</b>
7.1. INTRODUCCIÓN .....	2
7.2. DIRECTIVA DE LA LIBRERÍA DE ETIQUETAS .....	2
7.3. ETIQUETA CMS::USER .....	2
7.4. ETIQUETA CMS::PROPERTY .....	2
7.5. ETIQUETA CMS::LINK.....	2
7.6. ETIQUETA CMS::LABEL .....	2
7.7. ETIQUETA CMS::INCLUDE .....	40
7.8. ETIQUETA CMS::TEMPLATE.....	40
7.9. ETIQUETA CMS::INFO.....	40
<b>8. API DE OPENCMS .....</b>	<b>40</b>
8.1. CLASE CMSJSPACTIONELEMENT .....	40
8.1.1. USER.....	40
8.1.2. PROPERTY.....	41
8.1.3. LINK .....	41
8.1.4. LABEL .....	41
8.1.5. INCLUDE.....	41
8.1.6. TEMPLATE .....	41
8.1.7. INFO .....	41
8.1.8. GETNAVIGATION .....	41
8.1.9. GETCMSOBJECT .....	41
8.2. CLASE CMSJSPNAVELEMENT .....	42

---

---

8.2.1. GETDESCRIPTION.....	42
8.2.2. GETFILENAME.....	42
8.2.3. GETINFO.....	42
8.2.4. GETNAVTEXT.....	42
8.2.5. GETNAVTREELEVEL.....	42
8.2.6. GETPROPERTY .....	42
8.2.7. GETTITLE .....	42
8.2.8. ISFOLDERLINK .....	43
8.2.9. ISINNAVIGATION .....	43
<b>9. IMPORTACIÓN DEL MODULO DE ESPAÑOL.....</b>	<b>43</b>
<b>10. CREACIÓN DE UNA CARPETA PROYECTO .....</b>	<b>44</b>
10.1. CREACIÓN DE UNA CAJA(TTLISTBOX)¡ERROR! MARCADOR NO DEFINIDO.	46
10.2. CONFIGURACION ESTILO DE LA PLANTILLA .....	48
10.3. CREACIÓN DE TU PROPIO MENÚ.....	49
10.4. MODULO DE TEMPLATE TWO.....	50
10.5. AÑADIENDO GOOGLES MAPS AL PORTAL.....	54
<b>11. CONCLUSIÓN.....</b>	<b>57</b>

---

# 1. INTRODUCCION A OPENCMS

## 1.1 Concepto

OpenCMS es un gestor de contenido web de libre distribución dirigido a la construcción de intranets y de sitios web.

## 1.2 Funcionamiento de OpenCms

OpenCms sigue el paradigma cliente-servidor, funciona sobre un servidor web capaz de trabajar con páginas JSP y Servlets.

Como cualquier otra aplicación web, ofrece una serie de recursos que pueden ser solicitados por los clientes del servicio web.

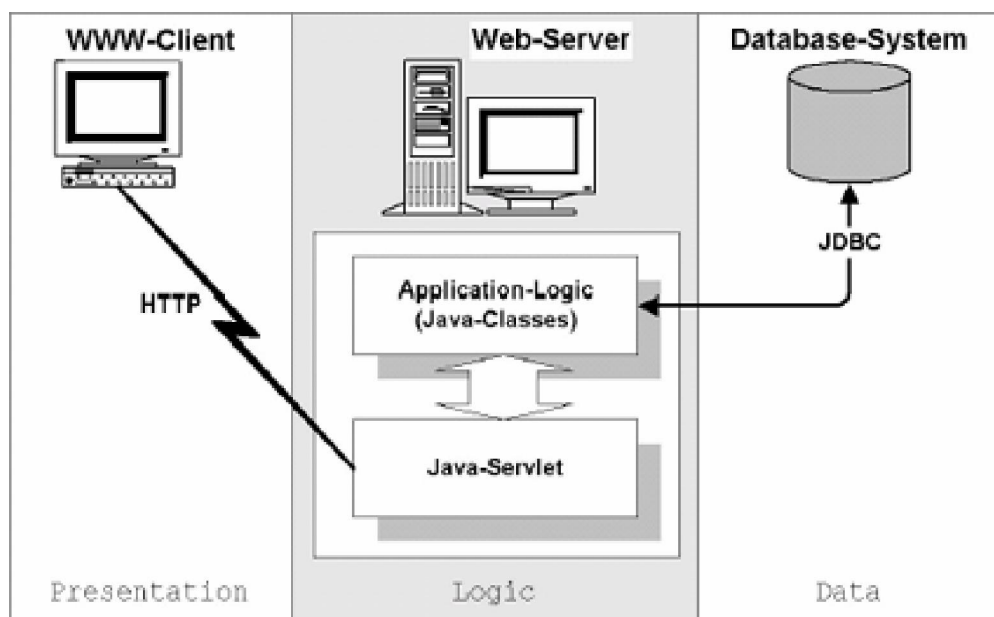
OpenCMS guarda todo sus recursos en base de datos, incluso la estructura de directorios y páginas que lo almacena en su sistema de ficheros virtual de ficheros (VFS).

OpenCMS utiliza una versión online de las páginas html que son las que puede visualizar el usuario, y que son independientes de todos los cambios que se estén produciendo en la gestión del sitio web o de la Intranet.

El proceso de funcionamiento básico de OpenCMS es:

1. El usuario final hace una petición de una página.
2. Esta petición la recibe Tomcat (o el servidor de aplicaciones que esté instalado), y la redirige al servlet de OpenCMS,
3. Este servlet busca la versión Online de dicha página en el VFS, mira que plantilla se utiliza para el diseño de dicha página y su contenido y genera el código html correspondiente a esa página y la devuelve.

Si la página pedida se ha podido exportar estáticamente no hay que generar la página, sino que se devuelve la propia página html almacenada en el sistema de ficheros real del servidor. Esta técnica hace que OpenCMS no tenga que acceder a la base de datos ni generar la página html en tiempo de ejecución.



### 1.3 Características generales

- Pensado para sitios web corporativos

- Idóneo para sitios web de medio y gran tamaño
- Estable (Versión 7.0.5)
- Existe actualmente bastante experiencia en su uso
- Open Source
- Plataformas Linux y Windows

## 1.4 Características técnicas y funcionales

- Desarrollado por completo en Java
- Gestión y desarrollo vía web
- Gestión de usuarios
- Soporte para flujos de trabajo
- Control de versiones
- Soporte internacional
- Exportación estática
- Cacheado de páginas
- API para la extensión mediante módulos

## 2. INSTALACIÓN DE OPENCMS

### 2.1 Instalación del JDK

Obtención del JDK 1.6.0\_11-b03 o superior de [java.sun.com](http://java.sun.com). Es necesario el entorno de desarrollo, no basta con la máquina virtual (JRE).

Lo instalamos en el sistema y lo configuramos de la siguiente manera.

En propiedades del sistema, vamos a la pestaña opciones avanzadas y le damos al botón variables de entorno.



En la variable de entorno JAVA\_HOME se guarda el directorio en el que se instala.

## 2.2 Instalación de Tomcat

Obtención de Tomcat de [jakarta.apache.org](http://jakarta.apache.org)

En la variable de entorno CATALINA\_HOME se guarda el directorio en el que se ha instalado.

OpenCMS utiliza el juego de caracteres ISO-8859-1. Para asegurar su correcto funcionamiento, hay que arrancar Tomcat con el siguiente parámetro

`-Dfile.encoding=ISO-859-1`

Se puede poner en la línea de comando que lanza el Tomcat o bien mediante la variable de entorno CATALINA\_OPTS.

Una vez iniciado el Tomcat se comprueba que funciona correctamente accediendo con el navegador a la máquina en el puerto 8080 (es en el que corre Tomcat por defecto).

### 2.2.1 Arranque y Apagado de Tomcat

Esta instalación no arranca en modo de servicios si no en forma manual. Para arrancarlo vamos al directorio bin del directorio tomcat, y hay dos auto ejecutables .bat: startup.bat y shutdown.bat

## 2.3 Instalación de MySQL

Obtener MySQL de <http://www.mysql.com>.

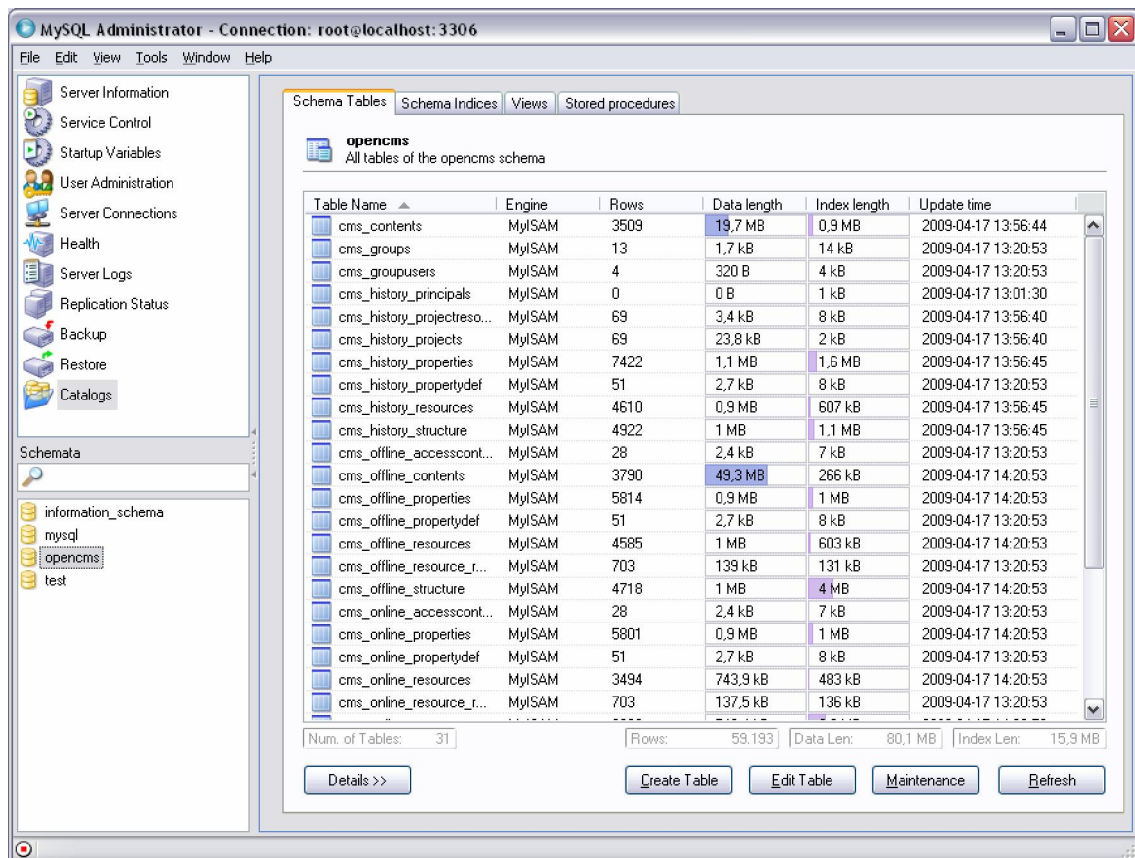
Si se instala en Windows lo hacemos en la unidad de disco que queramos:

Comprobar que funciona correctamente lanzando la consola de MySQL mediante el comando 'mysql'.

Con la herramienta de Mysql Administrador creamos un esquema llamado "opencms" para guardar la base de datos de opencms.



Visualización de todas las tablas del opencms:



A continuación creamos un usuario, para acceder a la base datos de OpenCMS. Se llamará

Usuario: usercms

Password: usercms

## 2.4 Despliegue de OpenCMS

Obtener OpenCms de <http://www.opencms.org>

Copiar el fichero opencms.war de la distribución en el directorio webapps de Tomcat.

Reiniciar Tomcat.

Comprobar que se ha desplegado correctamente mirando que en el directorio webapps se ha creado la carpeta opencms.

## **2.5 Consideraciones de seguridad**

### **2.5.1 Cambio de la contraseña de OpenCMS**

Acceder a la vista de administración y en las preferencias del usuario cambiar esta contraseña, pinchando en la pestaña “Datos de usuario”

### **2.5.2 Cambio de la contraseña de acceso a MySQL**

Se da de alta en MySQL, un usuario con acceso únicamente a la base de datos de OpenCms pero con todos los permisos sobre ésta.

En OpenCms modificar el fichero ‘opencms.properties’ para que se use este usuario y contraseña cuando se acceda a la base de datos desde OpenCms.

Datos del usuario:

User: usercms

Password: usercms

## **2.6 Ejecución del asistente**

Ejecutar el asistente para configurar OpenCms. Para ello abrir con el navegador la página:

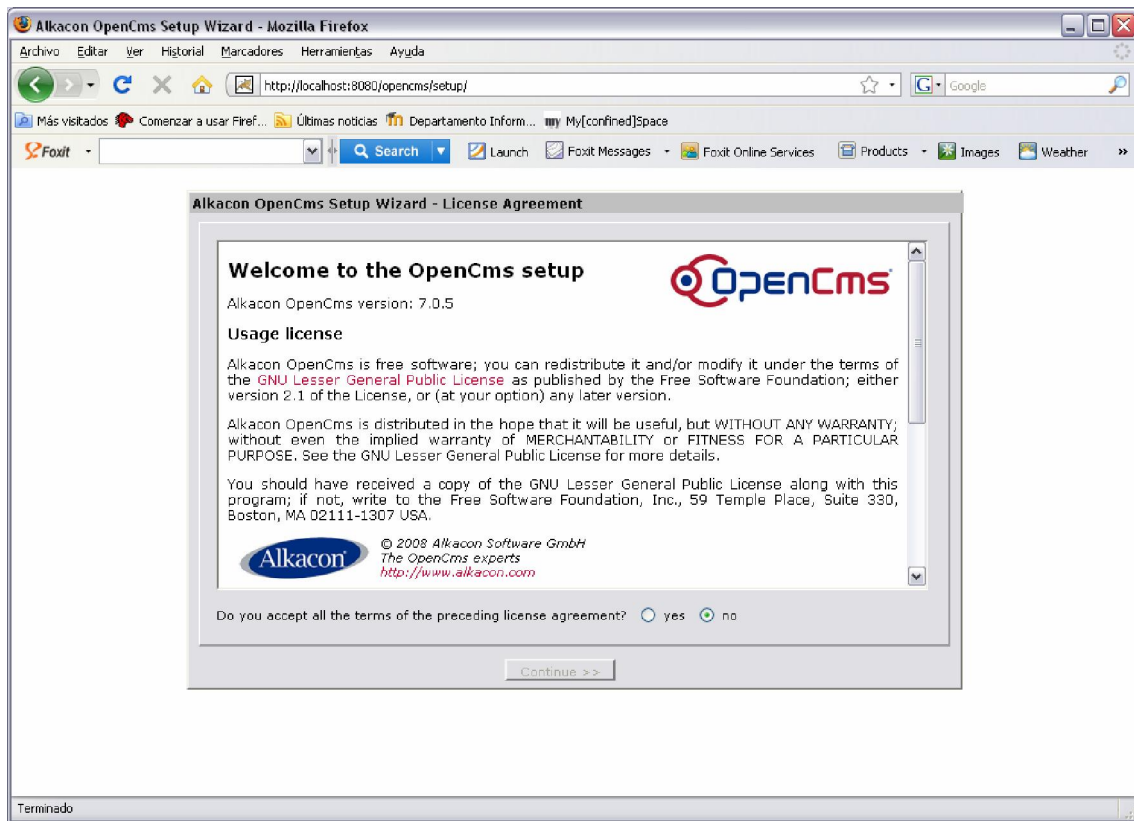
<http://localhost:8080/opencms/setup>

En localhost habrá que poner la máquina en la que se ha instalado Tomcat.

A partir de aquí sólo hay que seguir las instrucciones.

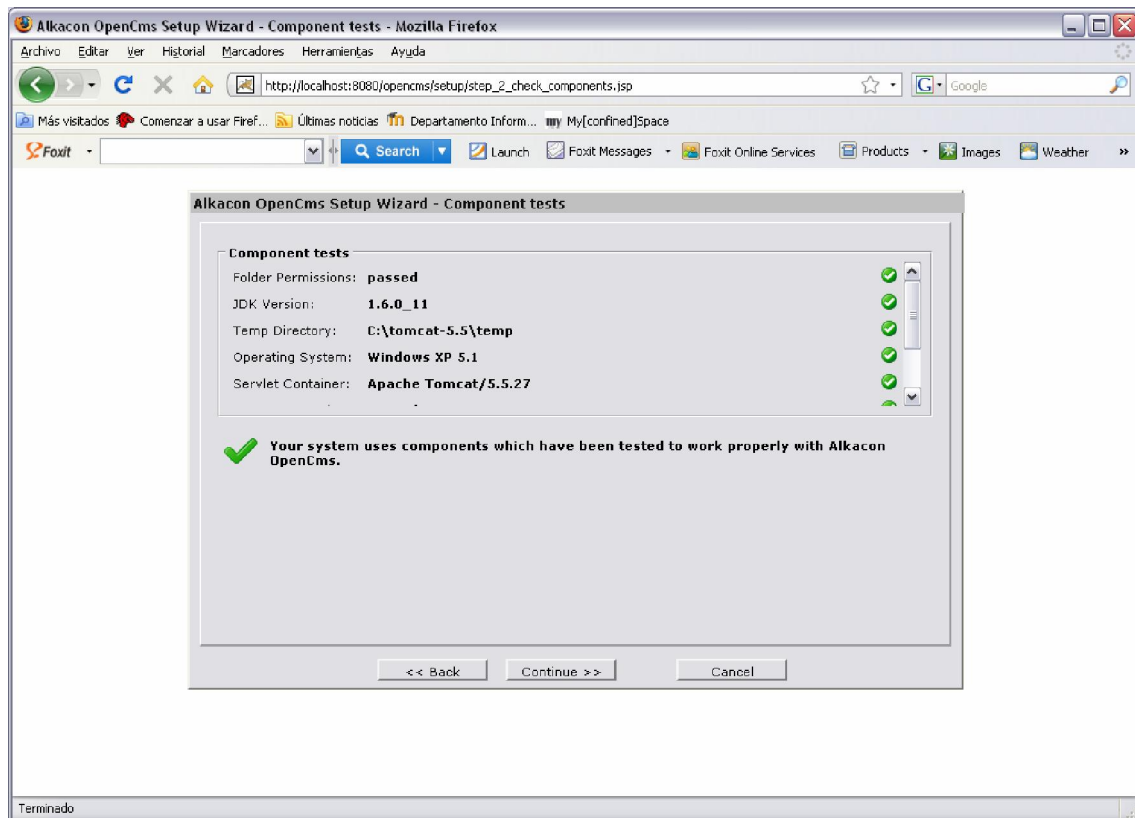
### **2.6.1 Licencia**

Pantalla de Bienvenida que muestra la versión y la licencia de opencms.



## 2.6.2 Componentes del OpenCMS

Muestra la versión del JDK directorio temporal del tomcat, plataforma donde esta montado el sistema operativo y la versión del apache tomcat.



### 2.6.3 Conexión con Base Datos OPENCMS

Database: Se selecciona el mysql instalado en la maquina.

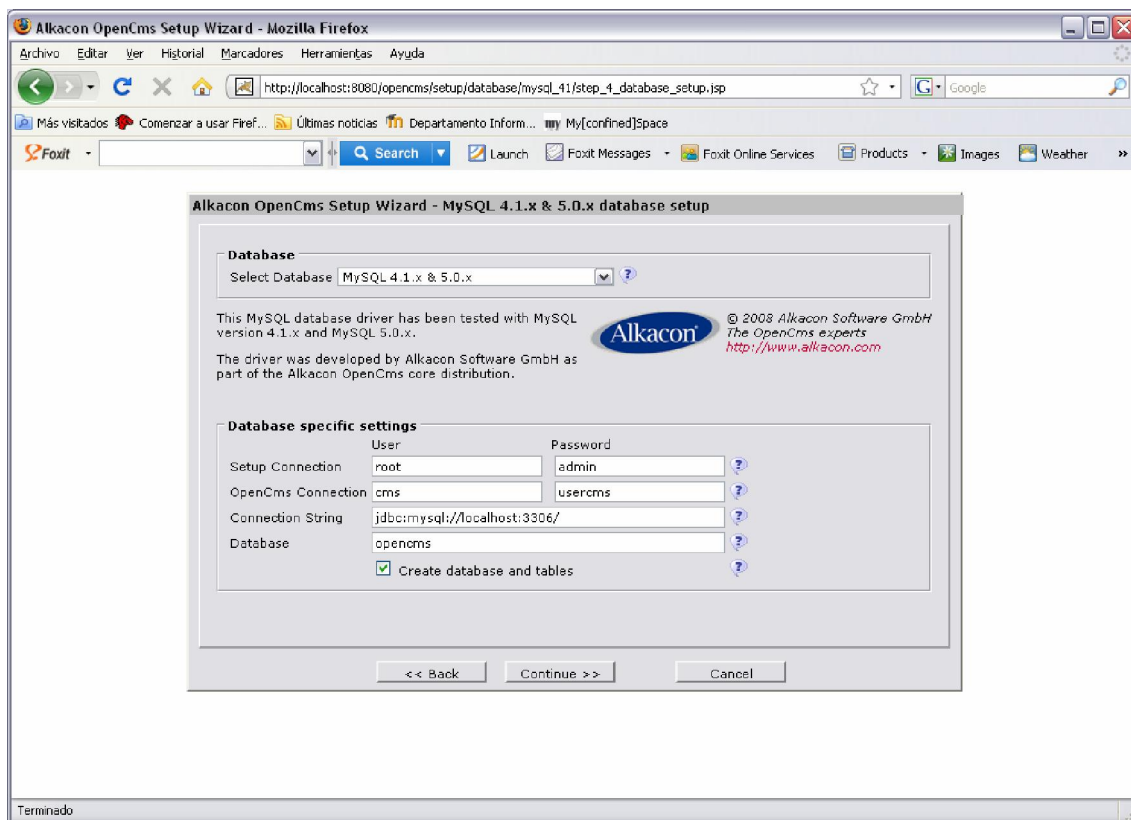
Se especifica los usuarios que tienen autoridad para acceder a la base datos del opencms.

Poniendo el usuario de la conexión será “root “y la contraseña “admin.”. Por defecto del mysql.

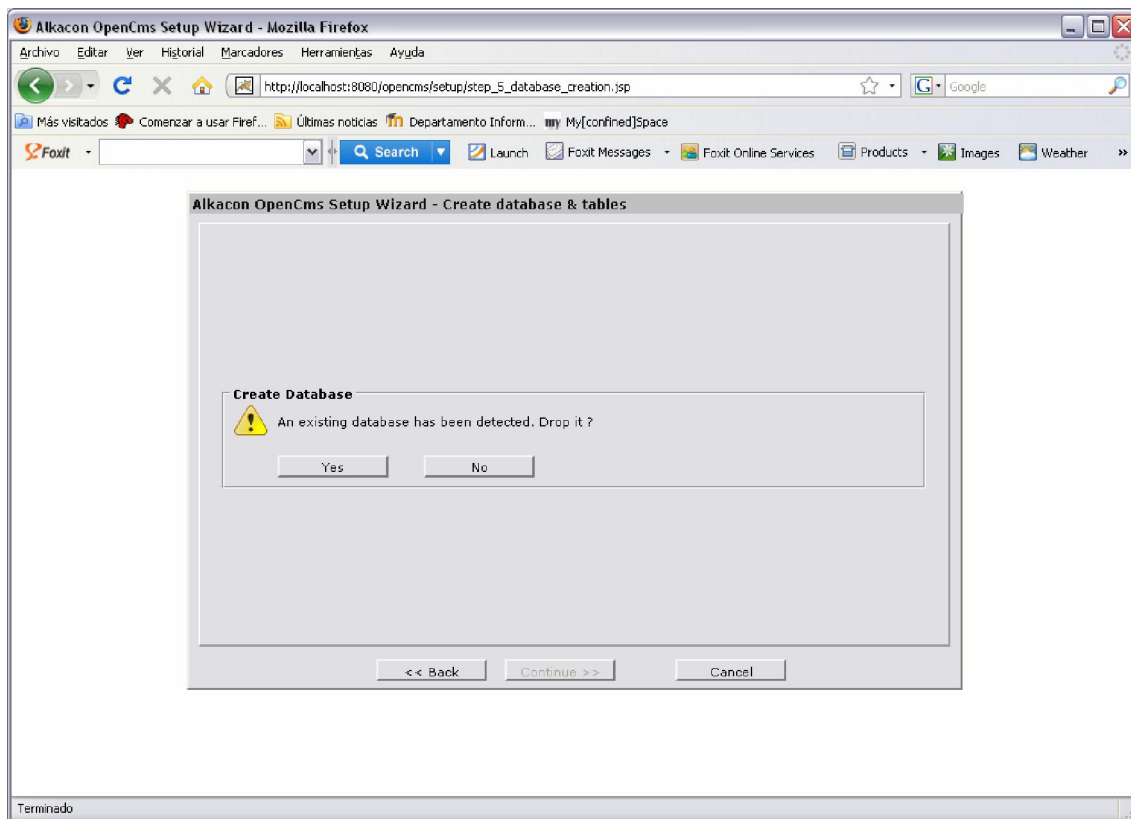
Y el usuario del opencms de la conexión es “cms” y el password “usercms”.

Ponemos la cadena de conexión y el nombre de la base de datos que le llamamos opencms.

Marcamos la casilla de abajo para que cree la base datos y las tablas.



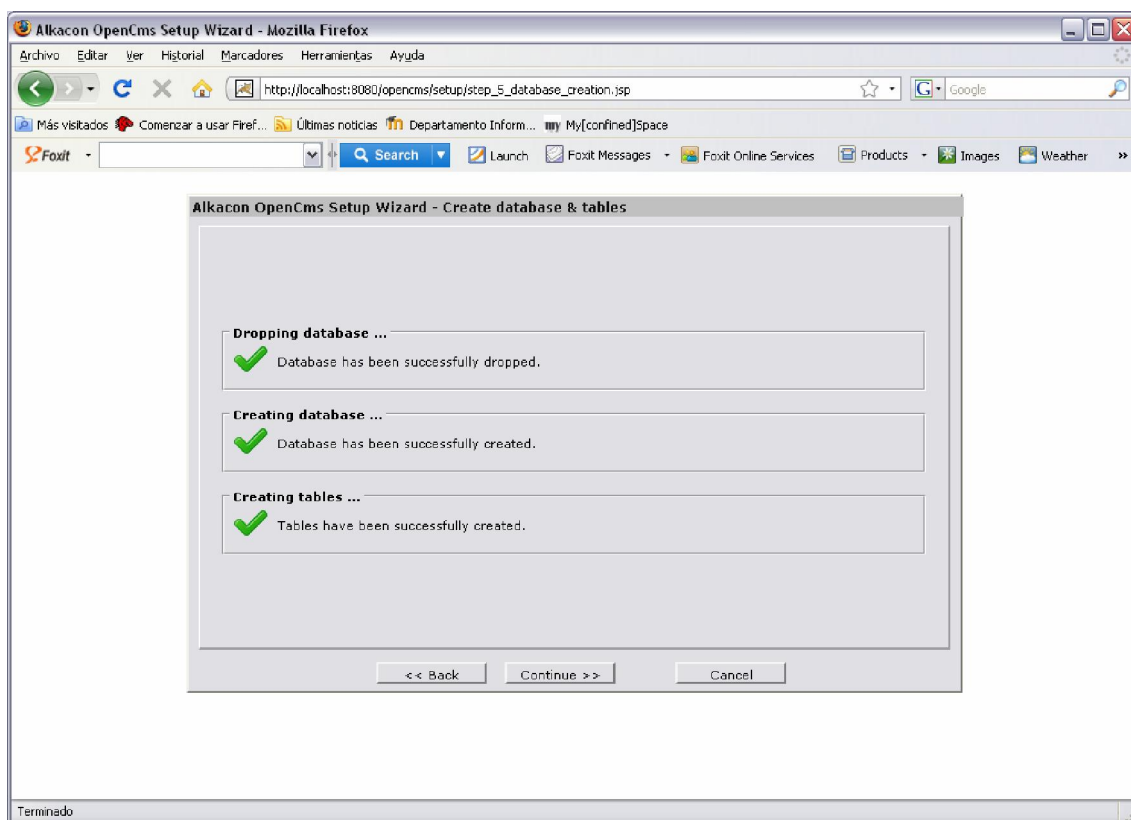
Había una base de datos existente que creamos llamada opencms y pide borrarla. Decimos que si.



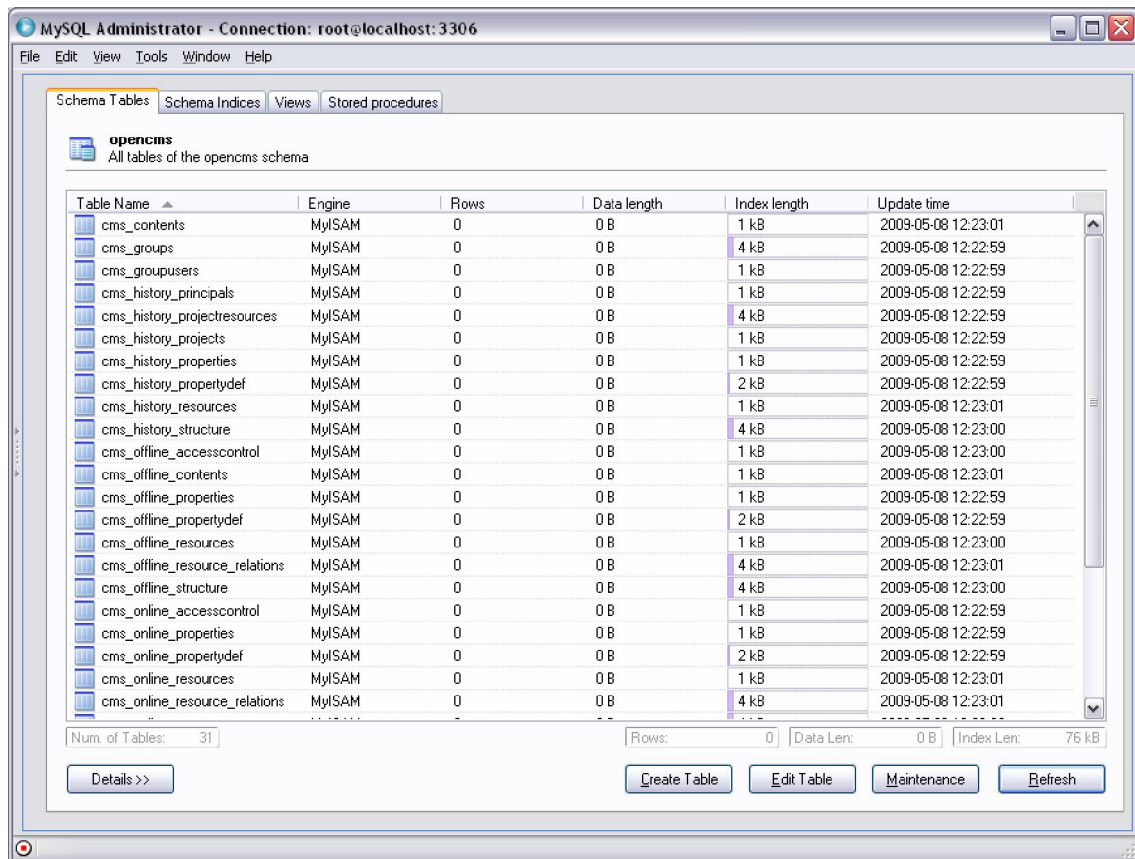
En el primer mensaje indica que la base de datos ha sido borrada.

En el segundo muestra que se creó la base de datos.

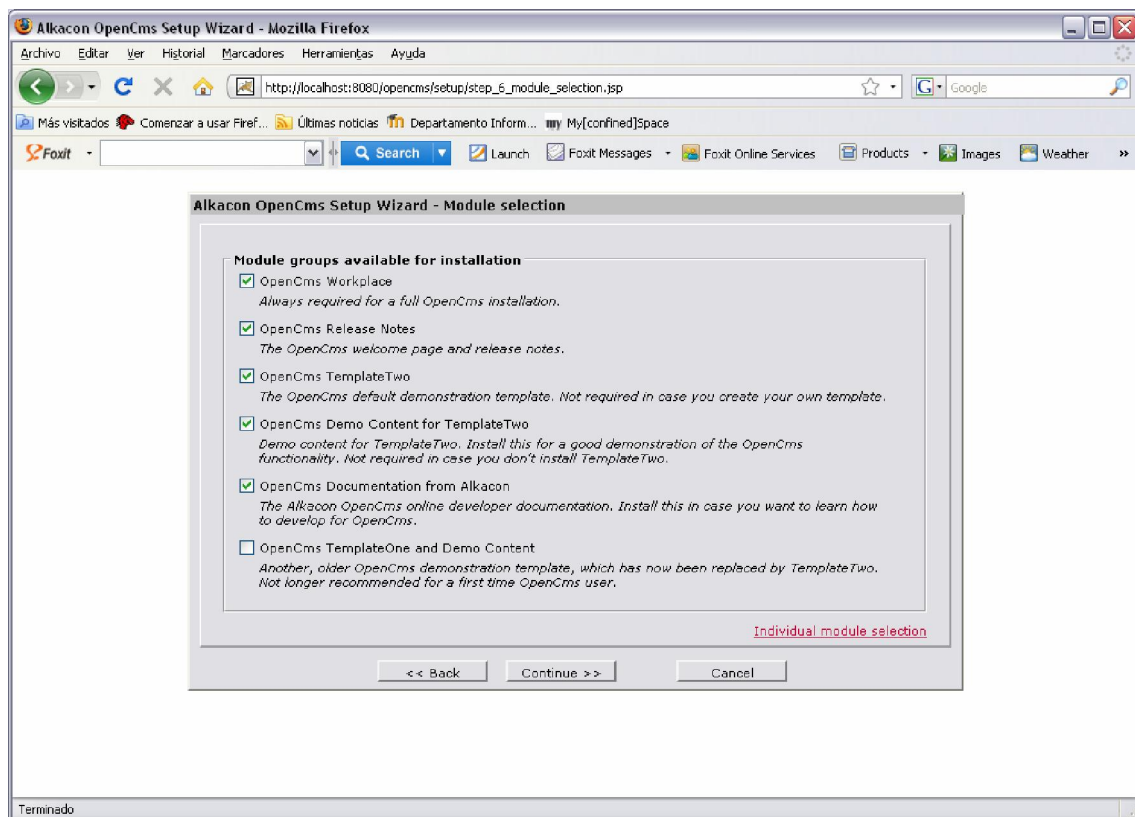
En el último mensaje muestra las tablas de la base de datos fueron creadas



Con el Mysql Administrador vemos que se crearon todas las tablas

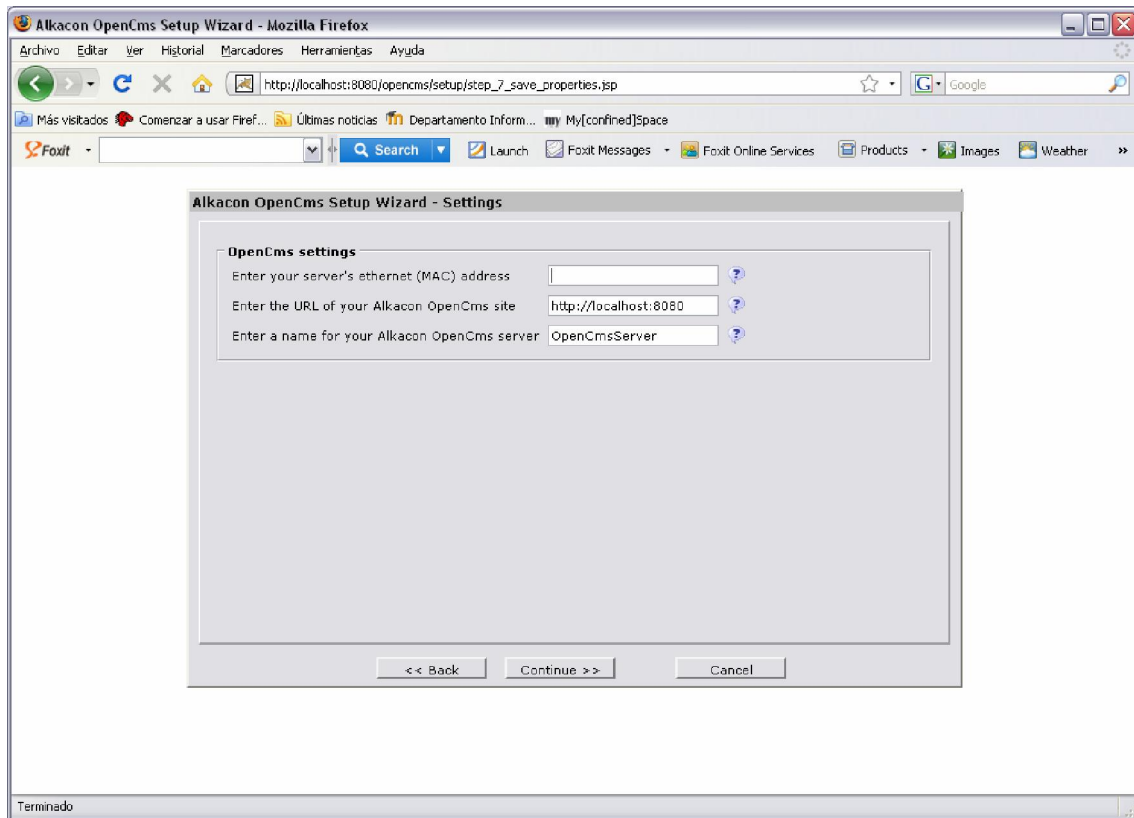


Aquí vemos que se te pide los módulos necesarios para cargar el OpenCms, también abajo puedes cargar personalmente tu propio modulo.

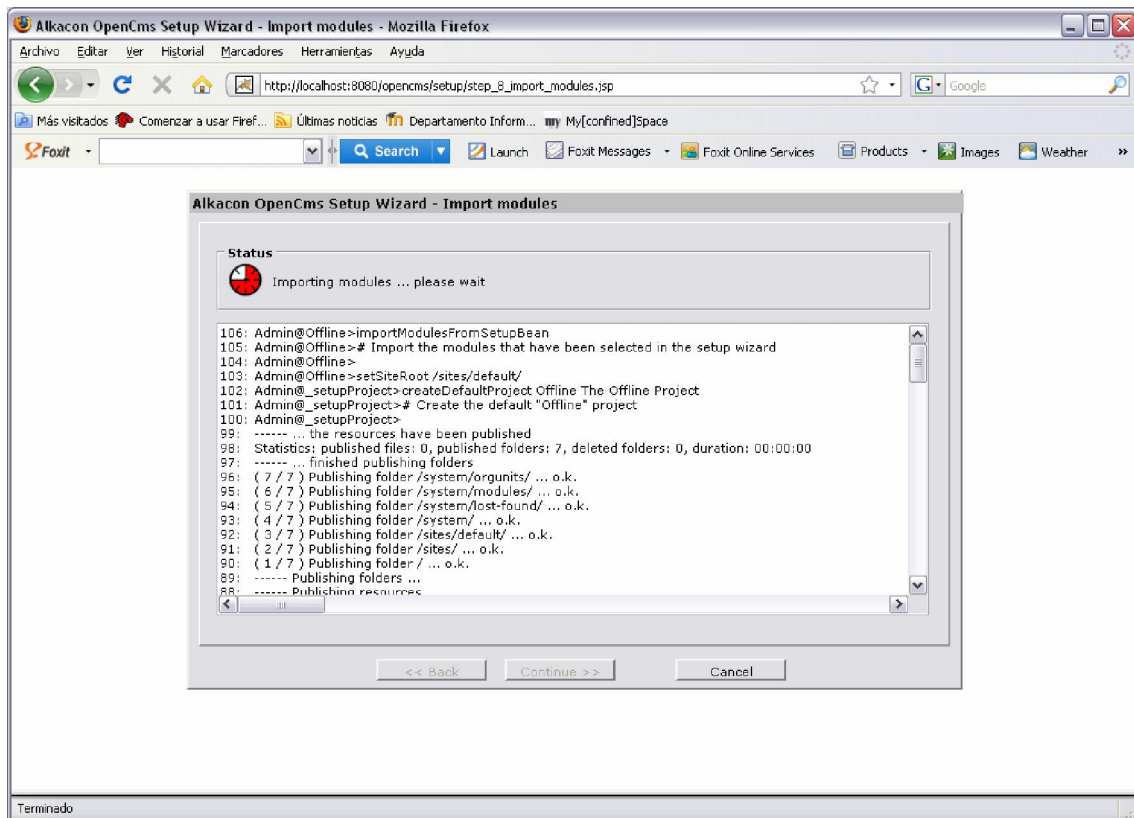




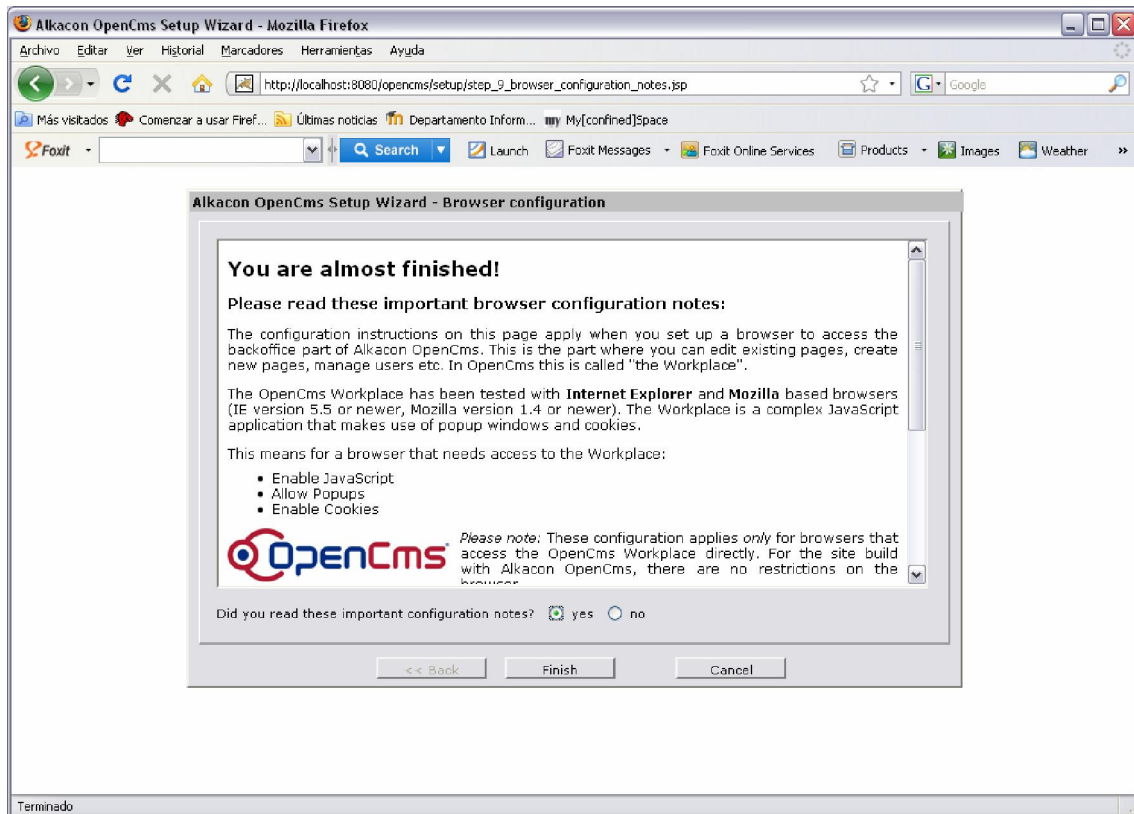
En las opciones de instalación nos pide la dirección url donde queremos ver al OpenCMS y debajo le ponemos un nombre al Servidor de OpenCms.



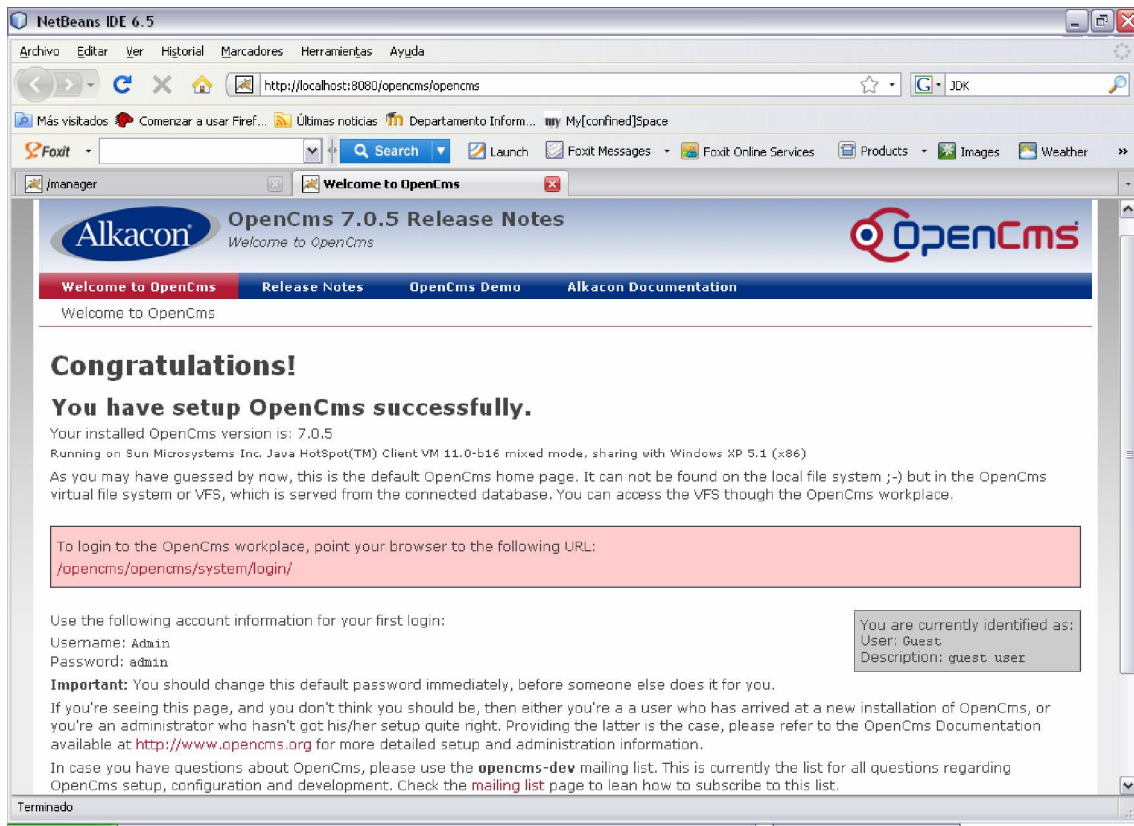
Aquí vemos como se cargan los módulos, tarda un tiempo



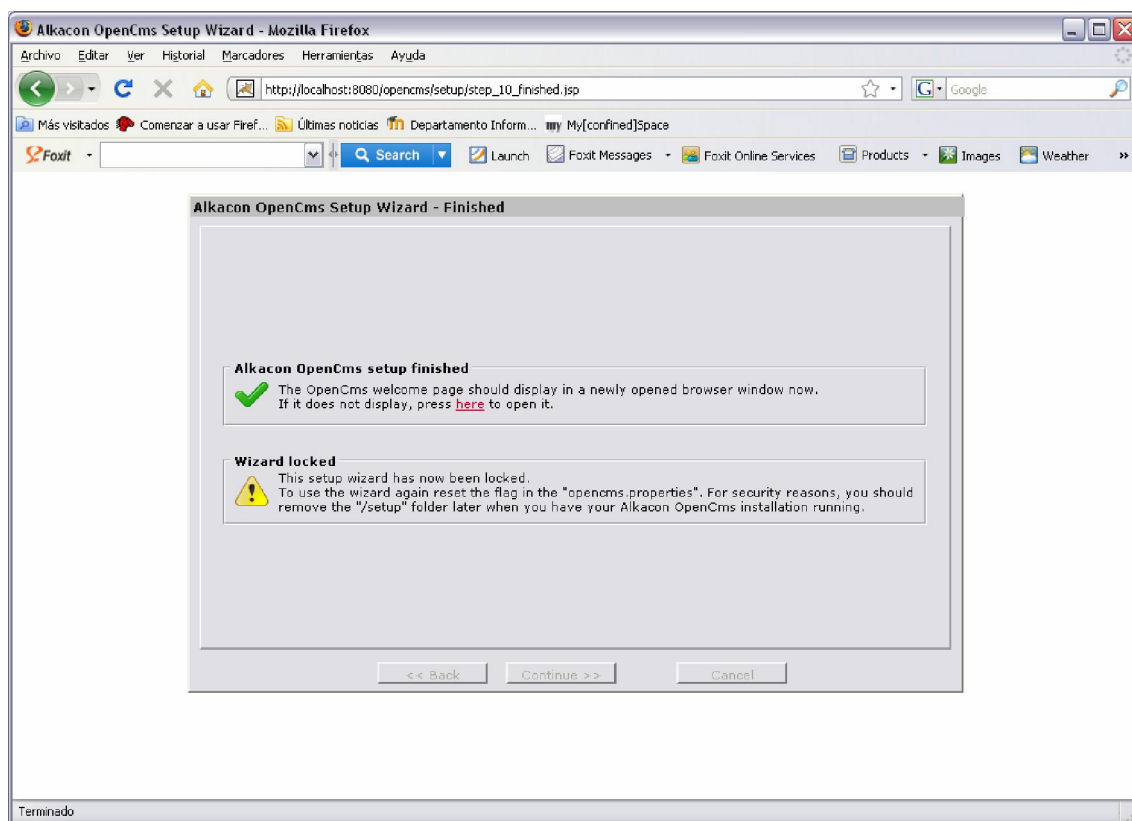
Final de la instalación



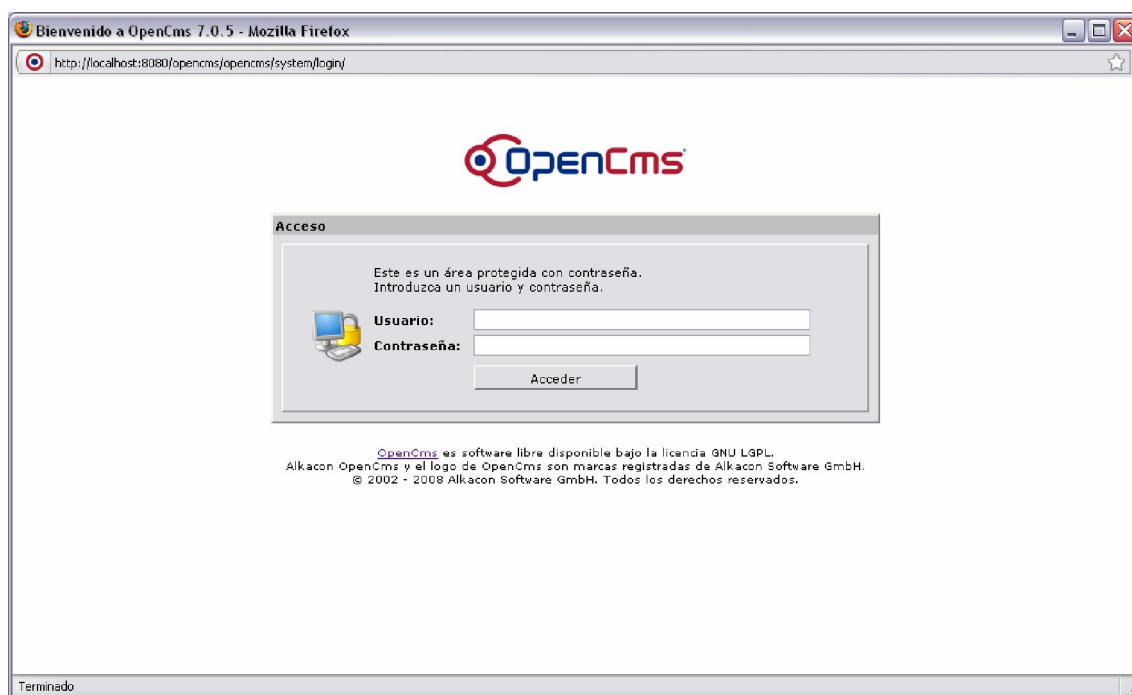
## 2.7 Vista Previa de OpenCMS



Antes de continuar hay que borrar el directorio setup de la carpeta opencms.



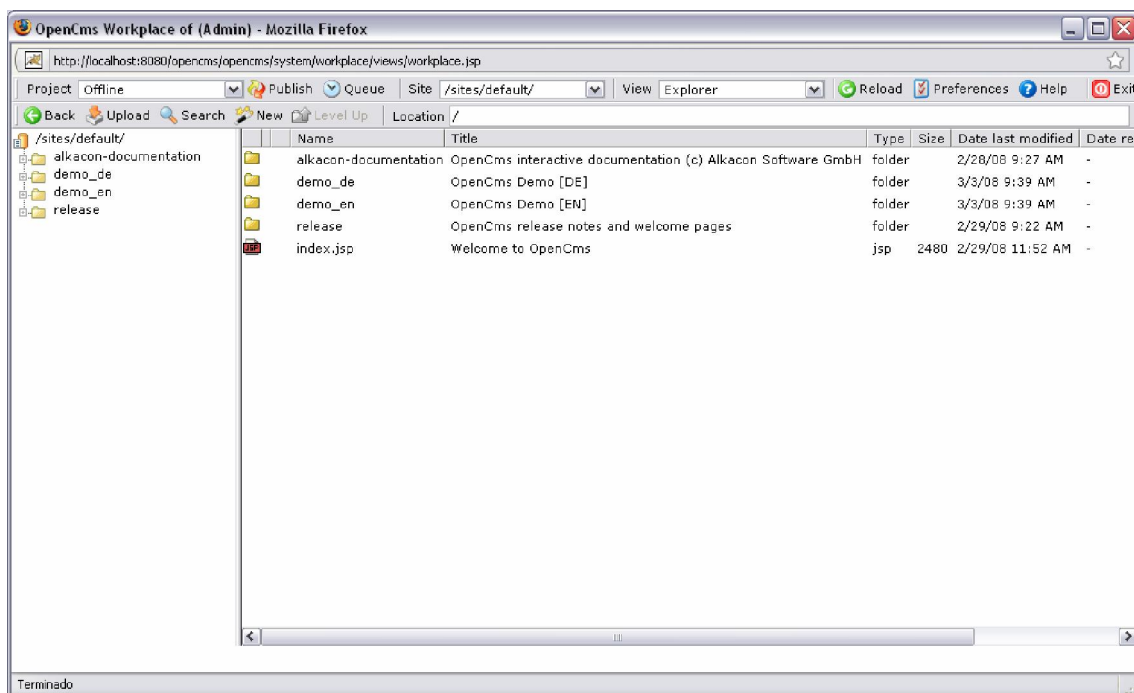
En esta página podemos acceder a la administración del portal OpenCms.



## 3. INTERFAZ DE OPENCMS

### 3.1 Elementos comunes

A continuación se van a describir aquellos elementos que están disponibles en la interfaz de usuario de OpenCms, independientemente de la vista que se use.



#### 3.1.1 Proyecto

Permite seleccionar el proyecto con el que se quiere trabajar en cada momento.

Un proyecto es un conjunto de carpetas que están disponibles para trabajar con ellas, es decir, un proyecto es un espacio de trabajo, en el que sólo están disponibles aquellos recursos que se hayan incluido en el proyecto.

Con un proyecto se asegura que al trabajar en él, no se van a realizar cambios sobre recursos que no estén incluidos en el proyecto.

### 3.1.2 Publicación proyecto

Permite publicar el proyecto actual.

### 3.1.3 Vista actual

Permite elegir la vista que se desea utilizar en cada momento.

Una vista ofrece una perspectiva particular sobre el entorno de OpenCms, que determina además a qué funcionalidades de éste se tiene acceso.

Hay dos vistas: **administración**, **explorador**. Las vistas se describen en detalle más adelante.

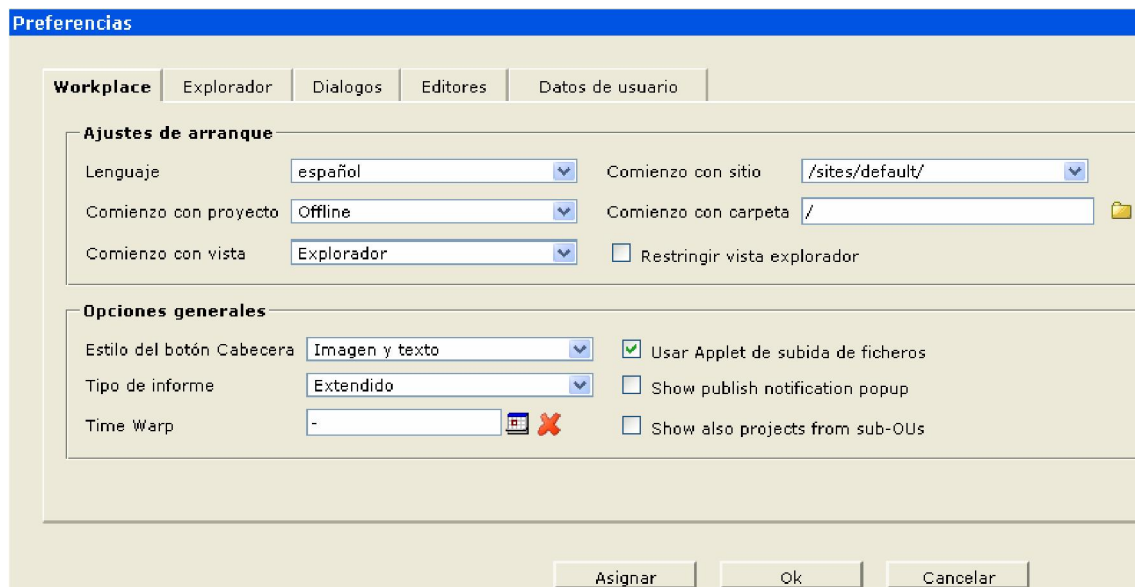
### 3.1.4 Recargar

Recarga la página actual.

### 3.1.5 Preferencias

Establece las preferencias del entorno para el usuario actual. Éstas se describen a continuación.

#### 3.1.5.1 Workplace

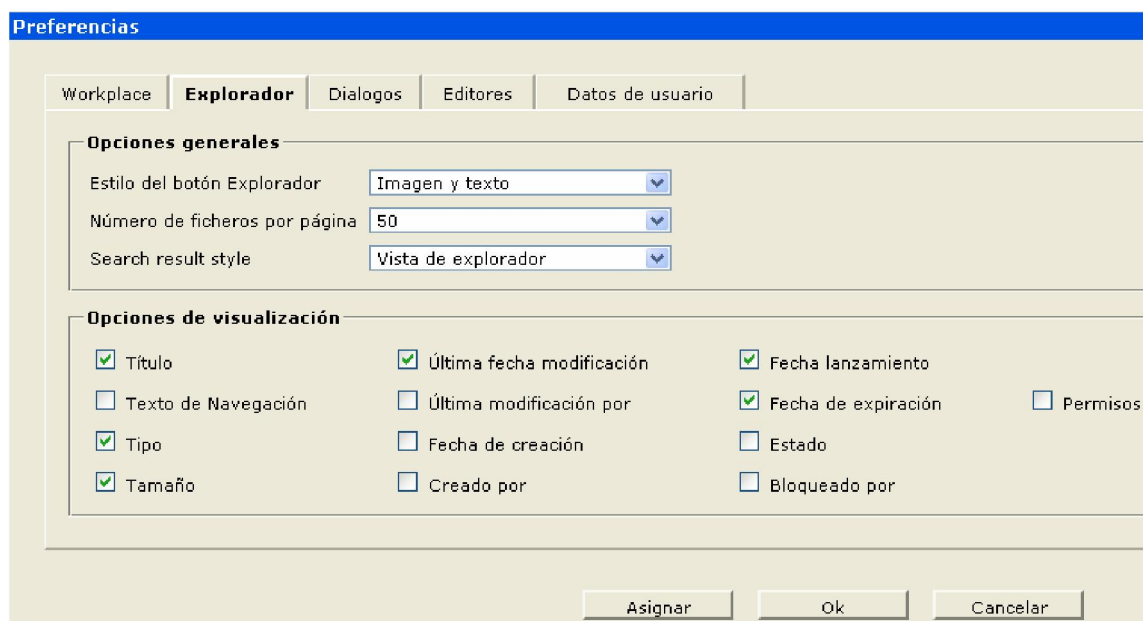


Opciones por defecto para todo el entorno tras logarse en OpenCms, son las siguientes:

- Idioma
- Proyecto con el que se empieza
- Vista con la que se empieza
- Comienzo con sitio.
- Comienzo con carpeta
- Opciones generales: Permite elegir algunas opciones que se muestran en la vista del explorador

### 3.1.5.2 Explorador

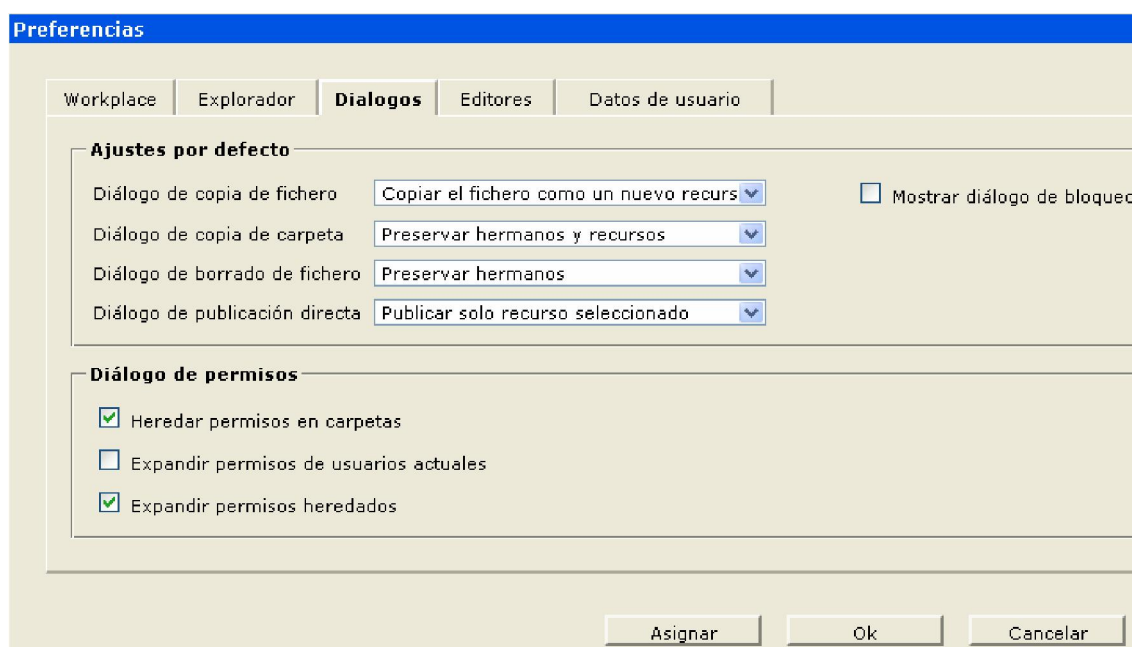
Permite elegir las columnas que se muestran en la vista de explorador para cada elemento de una carpeta.



### 3.1.5.3 Diálogos

Permite ajustar permisos para los nuevos ficheros que se creen, a través del borrado, copiados, y publicación directa.

Con cada operación de bloqueo se muestra un diálogo de confirmación (en realidad una nueva página) de la operación, que indica la situación en la que quedaría el elemento si se lleva a cabo la operación.



### 3.1.5.4 Editores

Permite cambiar las opciones generales de los botones y del contenido de la interfaz.



The screenshot shows the 'Preferencias' (Preferences) dialog box with the 'Editores' (Editors) tab selected. The dialog has a blue title bar and a light beige background. At the top, there are five tabs: 'Workplace', 'Explorador', 'Dialogos', 'Editores', and 'Datos de usuario'. The 'Editores' tab is active. Below the tabs, there are two sections: 'Opciones generales' (General options) and 'Editores preferidos' (Preferred editors). In the 'Opciones generales' section, there are two dropdown menus: 'Estilo del botón Editor' (Editor button style) and 'Estilo del botón "Edición directa"' (Direct editing button style), both set to 'Solo imagen' (Only image). In the 'Editores preferidos' section, there are three dropdown menus: 'Página con texto libre' (Free text page), 'Fichero de texto' (Text file), and 'XML Content', all set to 'Mejor disponible' (Best available). At the bottom right, there are three buttons: 'Asignar' (Assign), 'Ok', and 'Cancelar' (Cancel).

### 3.1.5.5 Datos de usuario

Muestra los datos del usuario y permite cambiar la contraseña, para cambiar el resto de los datos hay que utilizar la gestión de grupos de la vista de administración.

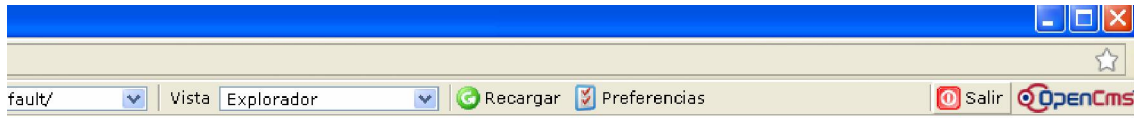
The screenshot shows the 'Preferencias' (Preferences) dialog box with the 'Datos de usuario' (User data) tab selected. The dialog has a blue title bar and a light beige background. At the top, there are five tabs: 'Workplace', 'Explorador', 'Dialogos', 'Editores', and 'Datos de usuario'. The 'Datos de usuario' tab is active. Below the tabs, there is a section titled 'Datos personales' (Personal data). This section contains a table with the following data:

Usuario	Admin	Email
Nombre		Ruta
Nombre		
Descripción	root administrator	

Below the table, there is a button labeled 'Cambiar clave' (Change password). At the bottom right, there are three buttons: 'Asignar' (Assign), 'Ok', and 'Cancelar' (Cancel).

### 3.1.6 Salir

Cierra la sesión actual y pasa a la página de login.



## 3.2 Vista de administración

Muestra las funciones presentes en OpenCms. Según la vista que se emplee, aparecerán habilitados o no.

Por defecto, OpenCms incluye una serie de funciones, y es posible añadir más mediante la creación o importación de módulos en OpenCms.

Las funciones incluidas por defecto en OpenCms son las siguientes:

### 3.2.1 Administración proyectos

Permite la creación y gestión de proyectos. También ofrece acceso al historial de cambios de los proyectos.

Todo lo anterior está accesible en las siguientes opciones:

#### 3.2.1.1 Nuevo proyecto

Esta función crea un proyecto. Permite poner sus datos básicos y propiedades.

Como datos básicos estarían el nombre y la descripción. El nombre debe seguir las normas de nombrado de paquetes de Java, indicando una jerarquía en la que los niveles se separan mediante puntos. La descripción es de tipo textual.

Entre las propiedades están: las carpetas a las que se accede con este proyecto, los canales, y qué grupo tiene acceso al proyecto.

#### 3.2.1.2 Proyectos actuales

Desde esta función se puede desbloquear, publicar, borrar y acceder a diversa información del proyecto.

Esta información permite obtener una lista de los cambios producidos en los ficheros del proyecto. Esta lista se puede filtrar para mostrar los ficheros modificados, los nuevos, los borrados, o los bloqueados.

#### 3.2.1.3 Historial del proyecto

Muestra los cambios realizados en los proyectos.

### 3.2.1.4 Gestionar propiedades del historial

Permite borrar el historial, así como habilitarlo o deshabilitarlo.

## 3.3 Usuarios y grupos

Desde esta función se pueden editar los usuarios, los grupos y enviar mensajes a todos los usuarios.

### 3.3.1 Gestión de Usuarios

Para los usuarios se introducen diversos datos, los siguientes son obligatorios: usuario, nombre, e-mail y clave.

También hay que indicar los grupos a los que pertenece.

Una vez creado un usuario, salvo el 'nombre del usuario' todos los demás datos pueden editarse.

También es posible deshabilitar un usuario, esto no lo elimina pero impide su acceso a OpenCms.

Desde esta función también es posible borrar un usuario.

Hay un usuario 'admin' por defecto para la administración de OpenCms. Hay otro usuario por defecto llamado 'Guest' para los usuarios que acceden a los contenidos publicados, es decir, para los visitantes de las páginas que ofrece OpenCms.

### 3.3.2 Gestión de Grupos

Los grupos son contenedores de usuarios y grupos. Permiten un realizar el control de acceso sobre los recursos, para un conjunto de usuarios o grupos en vez de tener que hacerlo de forma individual para cada uno.

-----  
Un grupo tiene un nombre único y una descripción. También es posible indicar un grupo padre del que heredará sus permisos.

Así mismo, es posible indicar el propósito del grupo: gestión de proyectos, usuarios de proyecto, o rol para tareas. Pudiendo tener asignado más de uno.

En OpenCms existen los siguientes grupos por defecto:

- Administrators: Grupo para los administradores, tienen acceso a todo.
- Guests: Es el grupo para los usuarios que acceden a los recursos publicados de OpenCms.
- ProjectManager: Grupo para los manager del proyecto.

- Users: Grupo para los usuarios que tengan que acceder a edición y publicación de contenidos.

Al crear un nuevo grupo, habrá que hacerlo heredar de alguno de los anteriores si se desea que tengan las capacidades descritas para cada uno de estos grupos.

### 3.4 Gestión de base de datos

Desde esta función se tiene acceso a la importación y exportación de OpenCms. Hay tres posibilidades:

Exportar base de datos: Permite indicar un conjunto de carpetas y ficheros a exportar, seleccionándolos de la jerarquía de carpetas.

Importar archivos con http: Permite indicar un conjunto de carpetas y ficheros a exportar, seleccionándolos de la jerarquía de carpetas.

Importar archivo desde el servidor: Permite importar desde un fichero, que puede estar en la máquina local o en el servidor.

*Al exportar hay que tener en cuenta que se exporta lo que contiene el proyecto 'online'.*

### 3.5 Gestión de módulos

Permite la creación y gestión de módulos. Los módulos permiten tratar como una unidad un conjunto de elementos diferentes, que pueden incluir: plantillas, clases java, librerías y recursos. Pueden utilizarse para incluir sólo uno de estos elementos o todos ellos. Son además, la forma de extender la funcionalidad de OpenCms.

Los módulos pueden ser creados, administrados, importados, exportados y borrados.

La administración de un módulo permite establecer su versión, las dependencias con respecto a otros módulos, así como asociarle otros recursos.

La exportación e importación emplean el formato zip, y permiten llevar un módulo de una instalación de OpenCms a otra. Tras la importación puede ser necesario reiniciar OpenCms.

Las opciones sobre módulos:

- Módulo nuevo
- Importar módulo vía servidor
- Importar módulo vía http

### 3.6 Historial de Ficheros

Tienes dos opciones: Limpiar el historial y Opciones de Historial

Opciones de Historial: Se guarda el número de versiones de archivo y puedes elegir:

- No guardar la versión
- Guardar sin versiones
- Guardar con versiones

### 3.7 Validación de Enlaces

Validación de enlaces internos: Permite gestionar los enlaces internos utilizados y chequea los enlaces del proyecto actual para determinar si alguno está roto.

Validación de enlaces externos: Permite gestionar los enlaces externos utilizados, chequea también los enlaces para determinar si alguno de ellos está roto.

### 3.8 Galerías

Permite añadir imágenes a OpenCms. Se pueden organizar en carpetas, y se ubican en '/system/galleries/pics'.

Hay diferentes opciones:

- Galería de descargas
- Galería HTML
- Galería de imágenes
- Galería de enlaces externos
- Galería de tablas

### **3.8.1 Galería de descargas**

Permite añadir cualquier tipo de fichero a OpenCms, con el propósito de referenciarlo para descarga. Se pueden organizar en carpetas, y se ubican en ‘/system/galleries/download’.

### **3.8.2 Galería HTML**

Permite disponer de una serie de páginas HTML.

### **3.8.3 Galería de imágenes.**

Permite añadir imágenes a OpenCms. Se pueden organizar en carpetas, y se ubican en ‘/system/galleries/pics’

### **3.8.4 Galería enlaces externos**

Permite gestionar los enlaces externos utilizados.

## **3.9 Gestión de Búsquedas**

Gestiona las acciones de los índices.

Permite crear un nuevo índice, ver los orígenes del índice, y ver las configuraciones de los campos.

## **3.10 Tareas programadas**

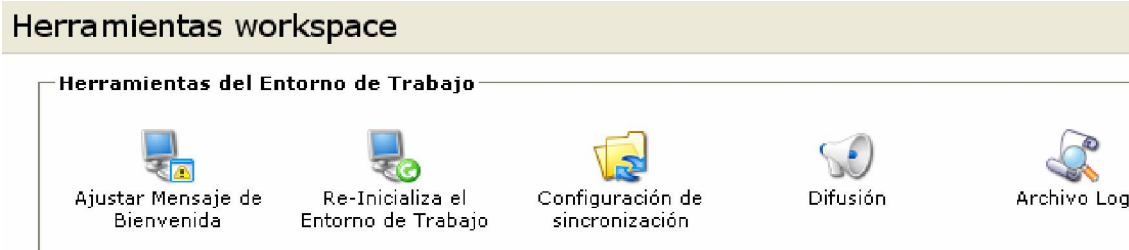
Esta función permite que OpenCms realice de forma automática la ejecución de una tarea. Para ello la tarea se programa a una fecha y hora determinada, de forma similar a como se hace en el cron de Unix, con un usuario, grupo. También se le indica la clase que ofrece la funcionalidad deseada, y los parámetros necesarios.

### 3.11 Administración de Flex cache

Permite gestionar la cache, eliminando de ésta aquellos contenidos que se deseen quitar en cada momento.

### 3.12 Herramientas Workspace

Son herramientas del entorno de trabajo entre ellas:



Ajustar Mensaje de Bienvenida: Muestra un mensaje al logearse en OpenCms.

Re-Inicializa el Entorno de Trabajo: Vuelve a establecer los valores por defecto reiniciando el entorno de trabajo.

Configuración de sincronización: Restablecer los valores desde un sitio o carpeta determinada.

Difusión: Muestra las sesiones actuales abiertas.

Además te ofrece un servicio de mensajería:

- Enviar un email a todos
- Enviar email a grupos
- Enviar mensaje a todos
- Enviar mensaje a grupos

Archivo Log: Ofrece acceso a las entradas del log de OpenCms. Muestra las últimas entradas del fichero de log, que está ubicado en 'opencms/WEB-INF/log'.

## 4. VISTA DE EXPLORADOR

La vista de explorador muestra el sistema de ficheros virtual de OpenCms (VFS) mediante dos zonas en el navegador. A la izquierda muestra un árbol con la estructura jerárquica del sistema de ficheros, a la derecha muestra los contenidos de la carpeta que está abierta.

La navegación se puede realizar empleando cualquiera de las dos vistas. Para acceder a los recursos y sus propiedades se emplea la ventana de la derecha.

En la ventana de la derecha, se muestran para cada recurso una serie de atributos. Los atributos que se muestran en cada momento, se establecen desde las preferencias del usuario.

En la zona de contenidos, el acceso a las operaciones que se pueden realizar sobre cada recurso se realiza haciendo clic con el botón izquierdo sobre el icono del recurso en cuestión.

El proyecto que se tenga seleccionado determinará las carpetas y recursos con los que se podrá trabajar.

### 4.1 Detalles de la vista del explorador

Para cada recurso se pueden mostrar los siguientes atributos (configurables desde las preferencias):

- Icono del recurso: Representa el tipo de recurso que se empleó para crear el recurso.
- Icono de bloqueo: Indica si el recurso está bloqueado. Si el icono es un candado bloqueado, indica que el recurso lo tiene bloqueado otro usuario. Si el icono es un candado abierto, indica que el recurso está bloqueado por el usuario actual. Si no hay ningún icono, el recurso no está bloqueado.
- Icono de modificación: Indica si el recurso ha sido modificado. El icono muestra una bandera.
- Nombre: Es el nombre de fichero que se le da.
- Título: Es una descripción del recurso.



- Tipo: Nombre del tipo de recurso utilizado para crear el recurso.
  - Fecha cambiada: Hora y fecha de la última modificación.
  - Tamaño: Tamaño del recurso.
  - Estado: Indica si está modificado o no.
  - Propietario: Muestra el propietario del recurso.
  - Grupo: Grupo que tiene asignado el fichero.
- 
- Permisos: Permisos que tiene asignados el recurso, siguen el esquema del sistema de ficheros de Unix, permisos de propietario, grupo y resto de usuarios. El permiso de ejecución ha sido sustituido por el de visualización (v).
  - Bloqueado por: Muestra el usuario que tiene bloqueado el fichero.

Además cuando un recurso no está modificado, su información se muestra con la fuente en color negro. Si está modificado depende de si había sido publicado previamente o no, cuando ha sido publicado se muestra en rojo, si no está publicado aún aparece en azul.

## 4.2 Operaciones sobre la carpeta actual

Para la carpeta a la que se está accediendo se dispone de las siguientes operaciones:

### 4.2.1 Volver

Carga la anterior url mostrada por el navegador.

### 4.2.2 Arriba

Pasa a la carpeta que contiene a la actual.

### 4.2.3 Buscar

Permite buscar empleando una serie de filtros a partir de la carpeta actual. Permite dos tipos de búsqueda, por nombre, o combinando los siguientes criterios de búsqueda: tipo de recurso, propiedad, valor.

### 4.2.4 Nuevo

Permite crear un nuevo recurso en la carpeta actual solo si el proyecto está offline.

OpenCms muestra una serie de tipos, hay que seleccionar uno y pulsar en continuar. En función del tipo de recurso se mostrará un diálogo con diferentes propiedades. Éstas se describen a continuación:

- Nombre: Nombre del recurso, incluye la extensión que le corresponde.
- Título: Representa una breve descripción del recurso.
- Plantilla: Plantilla XML que se empleará para generar la página que se mostrará. También puede optarse por no usar ninguna plantilla si se selecciona 'Empty template'. La lista de plantillas se genera a partir de todas las plantillas XML disponibles en las carpetas 'template' de todos los módulos de OpenCms.
- Añadir a navegación: Añade información de navegación al recurso. Ésta se puede emplear para generar un mapa web, un menú de navegación o un rastro de miga, y en general para generar cualquier información que requiera mostrar la ruta en la que se encuentra ubicado el recurso.
- Texto de navegación: Es el texto que se muestra para el recurso en la navegación.
- Insertar después de: Permite indicar la posición en la que se inserta este recurso en la estructura de navegación.

### 4.3 Operaciones sobre los recursos del VFS

Estas operaciones están disponibles haciendo click sobre el icono del recurso con el botón izquierdo del ratón.

En función de que un recurso se encuentre bloqueado, y de los permisos que tenga el usuario actual, se podrán realizar diferentes operaciones. Las operaciones que puedan suponer cambios en el recurso o en sus propiedades sólo están disponibles cuando el recurso está bloqueado.

Las operaciones disponibles son:

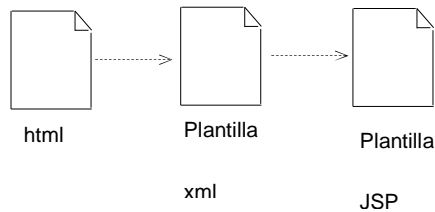
- Bloquear/Desbloquear/Quitar bloqueo: Cada una de ellas estará disponible en función del estado del recurso. Bloquear, bloquea el recurso para el usuario actual. Desbloquear, le quita al recurso el bloqueo que ejerce sobre él el usuario actual. Quitar bloqueo, en este caso, el recurso está bloqueado por otro usuario, lo que hace esta operación, es quitarle el bloqueo al otro usuario, y bloquearlo para el usuario actual.
- Publicar directamente (disponible sólo si está desbloqueado): Publica el recurso, es decir, lo hace disponible a todos los usuarios.

- Editar pagina: Edita el recurso empleando un editor WYSIWYG, que muestra el recurso tal como sería visible a los clientes.
- Editar código fuente: Edita el contenido del recurso en modo texto, sin interpretar su contenido.
- Copiar: Permite copiar el recurso
- Renombrar: Renombra el recurso.
- Mover: Mueve el recurso a otra ubicación de la estructura.
- Borrar: Borra el recurso. Si el recurso está publicado, aparece tachado con una línea. Si a continuación se publica, lo elimina también del proyecto 'online', por lo que desaparece completamente de OpenCms.
- Tocar: Pone el recurso como modificado con la fecha actual.
- Deshacer cambios: Deshace los últimos cambios realizados.
- Mostrar Enlaces del sistema de ficheros: Pasa mostrar las direcciones en base al sistema de ficheros virtual.
- Cambiar propietario: Cambia el propietario del recurso.
- Cambiar grupo: Cambia el grupo al que está asignado el recurso.
- Cambiar permisos: Permite cambiar los permisos del recurso.
- Cambiar navegación: Cambia las propiedades de navegación del recurso.
- Editar código de control: Permite editar el documento XML que establece la plantilla y el recurso empleados para generar la página correspondiente al recurso.
- Historial: Da acceso al histórico de versiones del recurso.
- Propiedades: Permite modificar las propiedades del recurso.

## 5. WEB DE EJEMPLO

## 5.1 Estructura de la web

Una página en OpenCMS se generará a partir de al menos una plantilla JSP, siguiendo la siguiente estructura:



## 5.2 Estructura del módulo

En primer lugar se crea un módulo, que contendrá todos los recursos del proyecto, pero antes que eso hay que hacerlo con el proyecto offline. Para ello se realizan los siguientes pasos:

- Elegir Vista de administración
- Pulsar sobre el icono de gestión de módulos
- Crear uno nuevo pulsando sobre el icono de nuevo modulo
- Ponerle un nombre con el formato de paquetes java: ej: es.cursoopencms.example

## 5.3 Plantilla XML

Crear una plantilla XML ‘ejemplo.xml’ en el directorio ‘/system/modules/es.cursoopencms.example/templates/’ y escribir:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmltemplate>
<stylesheet>/system/modules/com.alkacon.documentation.howto-
template/resources/mystyle.css</stylesheet>
<template>
  <element name="jsptemplate"/>
</template>
```

```
<elementdef name="jsptemplate">
  <class>com.opencms.flex.CmsJspTemplate</class>
  <template>../jsptemplates/ejemplo.jsp</template>
</elementdef>
</xmltemplate>
```

Es conveniente poner como título de la página un nombre descriptivo, pues este nombre es el que aparecerá cuando queramos hacer una página basada en este tipo de plantilla, por ejemplo 'CURSO – Ejemplo'. Para editar el título pulsar con el botón izquierdo sobre el icono del recurso y pulsar sobre propiedades y editar la propiedad título.

## 5.4 Plantilla JSP

Crear una plantilla JSP 'ejemplo.jsp' en el directorio '/system/modules/es.cursoopencms.example/jsptemplates/' y escribir

```
<% @ page session="false" %>
<% @ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>
<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title><cms:property name="title" escapeHtml="true" /></title>
<meta HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
  CHARSET=<cms:property name="content-encoding" default="ISO-8859-1" />">
<link type="text/css" href="<cms:link>../resources/mystyle.css</cms:link>">
</head>
<body><
<h2>Mi primer plantilla</h2>
<!--Cuerpo principal empieza aqui -->
<cms:include element="body"/>
<!--El cuerpo termina aquí -->
<h2>El pie de la plantilla</h2><
</body>
</html>
```

## 5.5 Página web

En cualquier directorio crear una nueva página html

Ponerle el nombre, título y elegir como plantilla XML la que acabamos de crear.

Después editar como página html y escribir un contenido.

Una vez terminado salir y pulsar sobre la página para observar el resultado.

## 6. USO de JSP en OPENCMS

Hay dos formas de acceder a la funcionalidad de OpenCms desde JSP:

- Mediante la librería de etiquetas de OpenCms
- Usando la clase 'CmsJspActionElement' del paquete 'com.opencms.flex.jsp'.

Con ambas formas se obtienen los mismos resultados. Ambas se describen más adelante.

### 6.1 Procesamiento de JSP en OpenCms

Todos los contenidos de OpenCms residen en base de datos, en lo que se llama el **sistema de ficheros virtual** de OpenCms. Por lo tanto no será posible encontrar los ficheros correspondientes a estos contenidos en el sistema de ficheros de la máquina.

Cuando se recibe una petición de una página JSP, en primer lugar se procesa para resolver las directivas que hagan referencia a otros ficheros JSP, a continuación se escribe en el disco, y por último se llama al servlet pasándole la ubicación del fichero JSP para que lo procese.

### 6.2 Estilos de sintaxis en JSP

En JSP hay dos formas de escribir una página JSP, el estilo JSP de versiones anteriores y un nuevo estilo llamado estilo XML.

En OpenCMS sólo se puede usar por el momento el estilo JSP, ya que el parser que se utiliza actualmente para procesar los ficheros JSP, no reconoce la sintaxis XML.

## 7. LIBRERÍA DE ETIQUETAS DE OPENCMS

### 7.1 Introducción

En OpenCms hay definidas una serie de etiquetas, que permiten acceder a las propiedades y funciones de OpenCms.

### 7.2 Directiva de la librería de etiquetas

Para emplear esta librería de etiquetas es necesario incluir la siguiente directiva en la página.

```
<% @ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>
```

Las etiquetas existentes se describen a continuación

### 7.3 Etiqueta cms::user

Permite obtener las propiedades del usuario actual.

### 7.4 Etiqueta cms::property

Permite obtener las propiedades del fichero actual.

### 7.5 Etiqueta cms::link

Convierte una url relativa a la estructura de OpenCms, en una url accesible externamente.

### 7.6 Etiqueta cms::label

*Cada etiqueta label está asociada únicamente a un control de formulario.*

## 7.7 Etiqueta cms::include

Permite incluir otros ficheros JSP de forma dinámica.

## 7.8 Etiqueta cms::template

Divide una página en varias partes que podrán ser incluidas posteriormente mediante la etiqueta cms::include.

## 7.9 Etiqueta cms::info

Permite obtener diversa información del entorno de ejecución de OpenCms.

# 8. API DE OPENCMS

## 8.1 Clase CmsJspActionElement

Esta clase proporciona la misma funcionalidad que ofrece la librería de etiquetas de OpenCms.

Para hacer uso de un objeto de esta clase hay que inicializarlo, lo que se realiza insertando el siguiente código.

```
<% com.opencms.flex.jsp.CmsJspActionElement cms = new  
CmsJspActionElement(pageContext, request, response); %>
```

De forma alternativa, se puede iniciar el objeto cms como un bean.

```
<jsp:useBean id="cms" class="com.opencms.flex.jsp.CmsJspActionElement">  
<% cms.init(pageContext, request, response); %>  
</jsp:useBean>
```

Las principales operaciones disponibles son:

### 8.1.1 user

Permite obtener las propiedades del usuario actual.



### 8.1.2 property

Permite obtener las propiedades del fichero actual.

### 8.1.3 link

Convierte una url relativa a la estructura de OpenCms, en una url accesible externamente.

### 8.1.4 label

Representa el texto asociado a controles que no tienen implícitamente un valor de texto.

### *8.1.5 include*

Permite incluir otros ficheros JSP de forma dinámica.

### *8.1.6 template*

Divide una página en varias partes que podrán ser incluidas posteriormente mediante la etiqueta `cms::include`.

### *8.1.7 info*

Permite obtener diversa información del entorno de ejecución de OpenCms.

### *8.1.8 getNavigation*

Devuelve una instancia de la clase `CmsJspNavBuilder` inicializada para el recurso actual.

### *8.1.9 GetCmsObject*

Devuelve un objeto de la clase `CmsObject`

## ***8.2 Clase CmsJspNavElement***

Esta clase proporciona información de navegación de un recurso de OpenCms. Entre las operaciones mas relevantes están:

### ***8.2.1 getDescription***

Obtiene el valor de la propiedad 'Description'.

### ***8.2.2 getFileName***

Devuelve el nombre del fichero correspondiente al recurso.

### ***8.2.3 getInfo***

Devuelve la información de navegación asociada al elemento.

### ***8.2.4 GetNavText***

Devuelve el texto de navegación asociada al elemento.

### ***8.2.5 getNavTreeLevel***

Devuelve el nivel asociado a la navegación del elemento.

### ***8.2.6 getProperty***

Devuelve el valor asociado a la propiedad proporcionada.

### ***8.2.7 getTitle***

Devuelve el valor de la propiedad 'Title'.

### *8.2.8 isFolderLink*

Devuelve 'true' si el elemento es una carpeta.

### *8.2.9 isInNavigation*

Devuelve 'true' si el elemento está incluido en la navegación

## 9. IMPORTACIÓN DEL MÓDULO DE ESPAÑOL

Antes que nada nos bajamos el módulo en la página de [www.opencmshispano.es](http://www.opencmshispano.es), en la vista de administración nos vamos a gestión de módulos vamos a Importar módulo vía http y subimos nuestro módulo, ahora que está subido vamos a preferencias y le damos a la pestaña workplace y en ajustes de arranque lenguaje elegimos español, entonces la plataforma cambiará el idioma por completo.



Workplace	Explorador	Dialogos	Editores	Datos
<b>Ajustes de arranque</b>				
Lenguaje	español			
Comienzo con proyecto	Offline			
Comienzo con vista	Explorador			

Una vez hecho esto para su correcto funcionamiento, tenemos que ir a tu proyecto opencms, ir a la carpeta web-inf y por ultimo a config y editar el fichero opencms\_system, copiar y poner esto

```
<localesconfigured>
  <locale>es</locale>
  <locale>en</locale>
  <locale>de</locale>
</localesconfigured>
<localesdefault>
  <locale>es</locale>
  <locale>en</locale>
```

```
<locale>de</locale>
```

Reiniciamos tomcat y ya tendremos cargado el modulo y todos los xml. Si queremos que nuestro módulo funcione por defecto en el proyecto. Primero vamos a vista explorador en /sytes/default, creamos una carpeta que va a darle el nombre del proyecto y en las propiedades de la carpeta pulsamos avanzado y cambiamos la variable locale poniendo lo que está en la imagen, primero cargará el español por defecto y de segundo el inglés.

locale	es,en
--------	-------

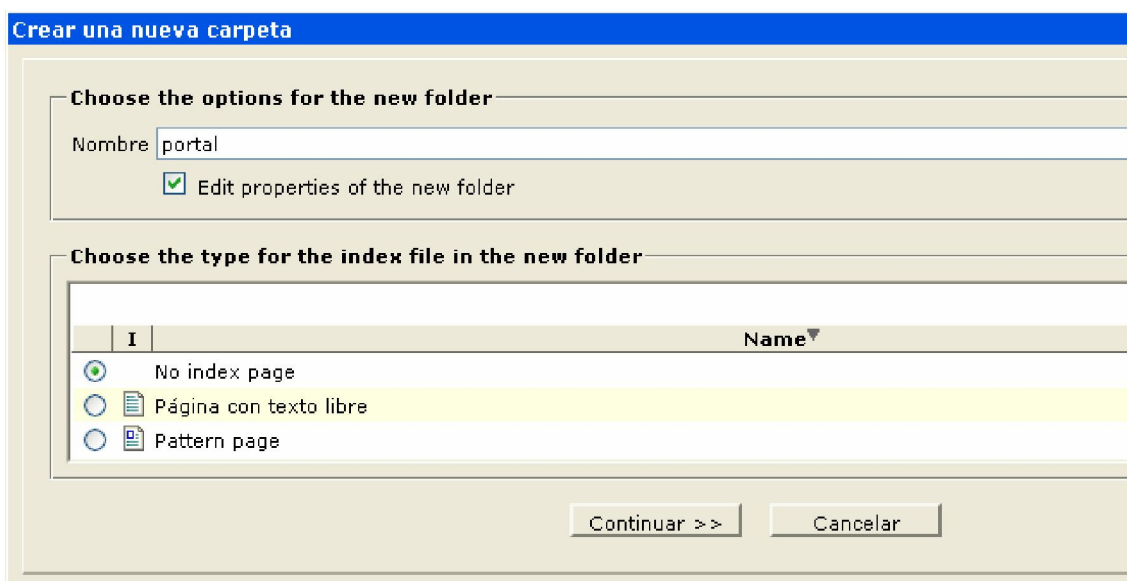
Ahora dentro de la carpeta cualquier documento que creemos por defecto lo hará en español.

## 10. CREACIÓN DE UNA CARPETA PROYECTO

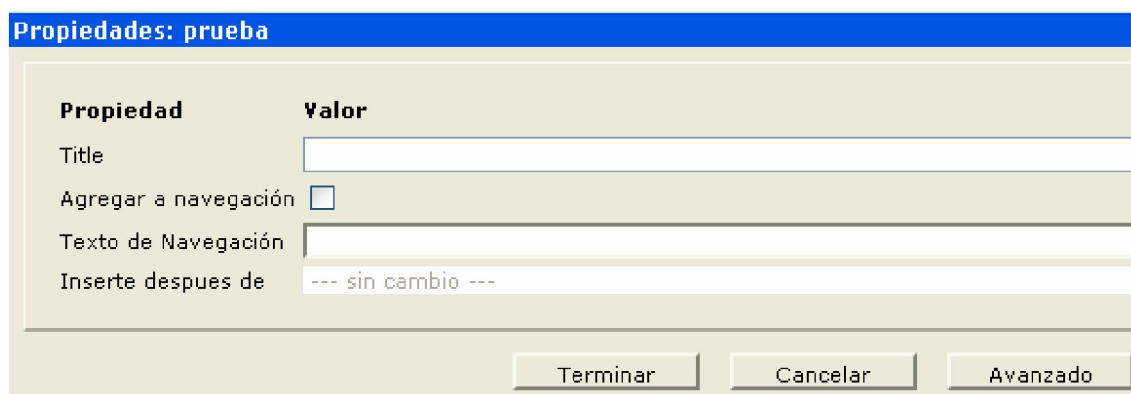
Primero creamos una carpeta en /system/default, le damos a nuevo.



Pinchamos en carpeta y le ponemos un nombre tal y como aparece en la siguiente imagen



Que va a tener el nombre de nuestro proyecto a continuación ponemos un titulo y le damos a terminar.



Nos quedará la carpeta con una bandera y bloqueado, ya que esta carpeta no está publicada









Luego dentro de esa carpeta podría contener otras carpetas que podrían usarse para las secciones de los menús y de la página en cuestión.

En este portal de ejemplo usamos las plantillas de template two, que vienen dentro del propio cms, así que para crear nuestro propio portal basándonos en esa plantilla hemos cogido dos carpetas.

Situándonos en /demo\_en/ nos copiamos las carpetas \_config\_ e \_info\_ y las llevamos a nuestra carpeta del portal.

## 10.1. CREACION DE UNA CAJA (TTLISBOX)

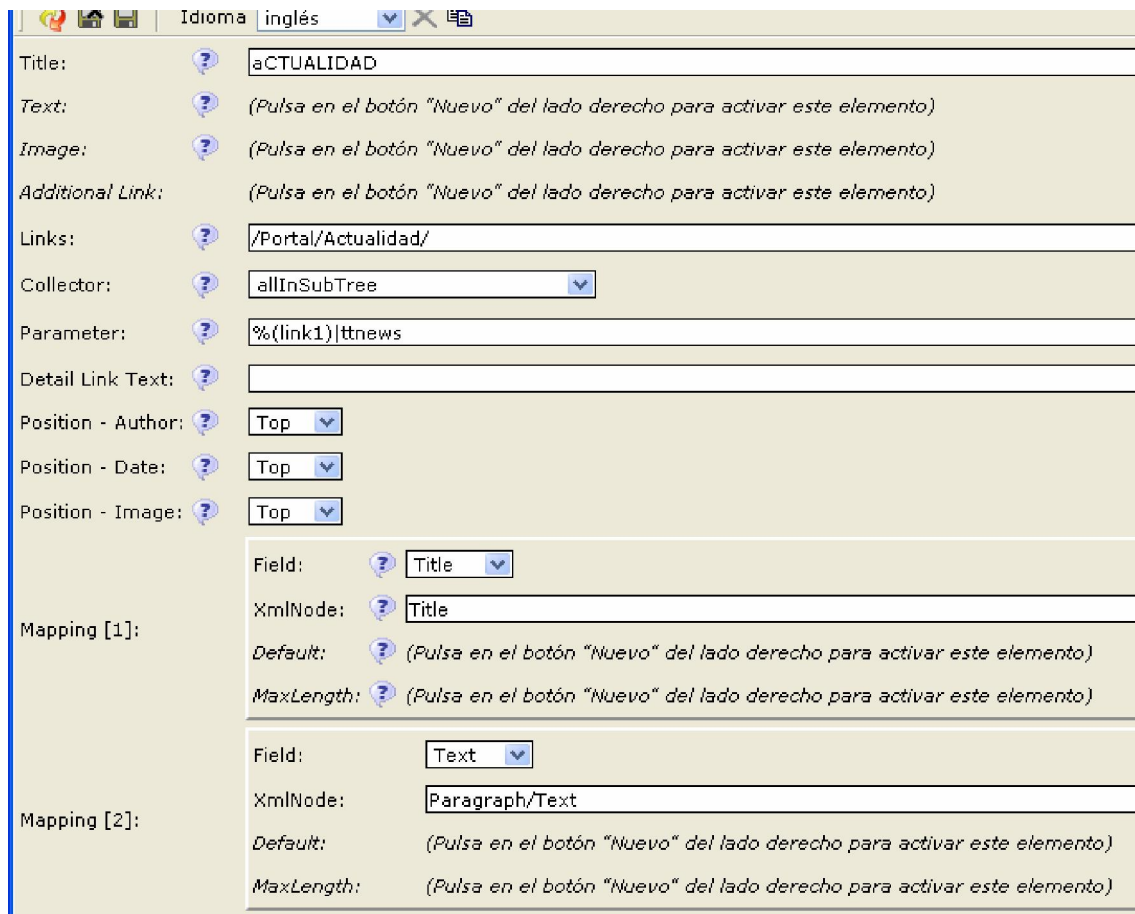
En la carpeta \_config\_ estarán todos nuestros ttlistbox, que son los diseños de las cajas que aparecerán en nuestro portal

	Nombre	Título	Navigation Text	Tipo
	layout	Page layout		folder
	Actualidades.html	aCTUALIDAD		ttlistbox
	cajaeventos.html	Eventos		ttlistbox
	eventos	Eventos		ttlistbox
	Noticias.html	Noticias de la web		ttlistbox
	noticias_portal.html	Título de las noticias		ttlistbox

para la creación de un ttlistbox, hay que ir a “nuevo” elegimos “templatetwo”, y a continuación elegimos “ttlisbox”



Le ponemos el nombre y no marcamos nada más. Una vez que lo tenemos hecho pulsamos sobre editar y nos aparecerá la siguiente imagen



Idioma: inglés									
Title:	aCTUALIDAD								
Text:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
Image:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
Additional Link:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
Links:	/Portal/Actualidad/								
Collector:	allInSubTree								
Parameter:	%(link1) ttnews								
Detail Link Text:									
Position - Author:	Top								
Position - Date:	Top								
Position - Image:	Top								
Mapping [1]:	<table border="1"><tr><td>Field:</td><td>Title</td></tr><tr><td>XmlNode:</td><td>Title</td></tr><tr><td>Default:</td><td>(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)</td></tr><tr><td>MaxLength:</td><td>(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)</td></tr></table>	Field:	Title	XmlNode:	Title	Default:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)	MaxLength:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)
Field:	Title								
XmlNode:	Title								
Default:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
MaxLength:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
Mapping [2]:	<table border="1"><tr><td>Field:</td><td>Text</td></tr><tr><td>XmlNode:</td><td>Paragraph/Text</td></tr><tr><td>Default:</td><td>(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)</td></tr><tr><td>MaxLength:</td><td>(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)</td></tr></table>	Field:	Text	XmlNode:	Paragraph/Text	Default:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)	MaxLength:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)
Field:	Text								
XmlNode:	Paragraph/Text								
Default:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								
MaxLength:	(Pulsa en el botón "Nuevo" del lado derecho para activar este elemento)								

Le indicamos un nombre de titulo a la caja, en el link le indicamos donde queremos que busque el contenido que contendrá la caja, es decir si queremos guardar noticias pues creamos una carpeta noticias y lo enlazamos a esa carpeta.

En collector le decimos donde queremos que haga la búsqueda.

En parameter introducimos estos parámetros %(link1)|ttnews que significa que cuando ponemos link1 coja el primer link y si ponemos ttnews lo que habrá dentro por ejemplo de la carpeta noticias serán de tipo ttnews, o pueden ser de cualquier otro tipo, si ponemos %(link1)|ttnews|1: cogerá solo uno

%(link1)|ttnews|2: cogerá solo dos

%(link1)|ttnews: cogerá todo lo que haya de ttnews

También puedes ponerlo usando otros links haciendo el mismo procedimiento.

En Mapping podemos agregar todos los que queramos, nosotros creamos dos.

En Mapping[1] en el campo Field le decimos que queremos que aparezca en la caja, el titulo de la noticia y ponemos “Title”

En XmlNode: ponemos “Title” también.

En Mapping[2]: Queremos que salga parte del texto, elegimos “Text” y en XmlNode ponemos Paragraph/Text , que es donde aparecerá el cuerpo del texto.

En MaxLength podemos poner un máximo de tamaño del cuerpo.

## 10.2. CONFIGURAR ESTILO DE LA PLANTILLA

Situándonos en /Portal/\_config\_/ esta la carpeta layout que es donde se configura el diseño de nuestro portal, los ficheros que nos harán falta son: default,options,style,threecolumns,twocolumns.

Default: Es el fichero que usará el portal por defecto para la configuración, es decir lo que cambiemos aparecerá en la página por defecto.

Options: Sirve para cambiar la cabecera y el pie del portal

Style: Puedes cambiar los colores de todo lo que haya en la página, puedes cambiar el color de la cabecera, las cajas, el pie, los menús los enlaces, etc...

Threecolumns: Puedes elegir tu portal que contenga tres columnas y de la manera que desees.

Twocolumns: Puedes elegir tu portal que contenga dos columnas y de la manera que desees.

En la carpeta \_info\_ estarán todos nuestros textbox que son otro tipo de cajas, en nuestro caso, yo tengo una caja que es la que por defecto aparece en todas las secciones del portal. Gracias al fichero /Portal/\_config\_/default le indiqué que por defecto cogiera esta caja en toda la página.



## 10.3 CREACIÓN DE TU PROPIO MENÚ

Podemos crear nuestro propio menú creando las carpetas que queramos, así que por defecto tendremos que crear un index, y luego el resto de contenido. Tenemos que crear siempre un index porque si no cuando pulsemos en el menú correspondiente aparecerá en blanco, así que tenemos que crear un index a partir del template two.

Creamos nuestra propia carpeta dentro del /Portal/. Le damos a nuevo carpeta

Ponemos el nombre que queramos que aparezca en el menú, le damos a siguiente y marcamos agregar texto a la navegación

Propiedad	Valor
Title	<input type="text"/>
Agregar a navegación	<input type="checkbox"/>
Texto de Navegación	<input type="text"/>
Inserte despues de	--- sin cambio ---

Terminar Cancelar Avanzado

así que ponemos el titulo que queremos que aparezca en el menú, le damos a terminar y a continuación creamos nuestro index. Una vez creado lo editamos añadiendo el titulo y si queremos poner algo en el cuerpo , pero hay algo fundamentalmente importante

Links: (Pulsa en el botón "Nuevo" del lado derecho para activar este)

Preset: ? (Pulsa en el botón "Nuevo" del lado derecho para activar este)

	Box:	Orientation:	Color Schema:
Element [1]:	? /Portal/_config_/Noticias.html	? Center	? Box Schema 3
Element [2]:	/Portal/_info_/lenguajes.html	Right	Box Schema 3

En Element[1] donde pone Box elegiremos la caja que hemos creado de noticias que está en la ruta `/_config_/Noticias.html`

En orientation elegiremos en que posición queremos que aparezca la caja, en este caso centrado, y en Color Shema elegimos el esquema de colores de la caja.








En Element[2] elegimos la caja que por defecto aparece en el lateral derecho .Si queremos podemos añadir las cajas que queramos que aparezcan en nuestro índice.

## 10.4. MÓDULO TEMPLATE TWO

Mi portal cms está basado en el modulo templatetwo, y todas las plantillas estilos las utilizo de ese módulo.

El módulo de templatetwo está situado en `/system/modules/org.opencms.frontend.templatetwo/`

Dentro de ese modulo hay varias carpetas.

	Nombre	Titulo	Navigator
	classes		
	elements		
	java_src		
	pages		
	resources		
	schemas		
	templates		

Nosotros usaremos la carpeta templates, shemas y elements.

En la carpeta templates: `main.jsp`

Hemos editado ese fichero para que coja la plantilla que queremos usar.

```
<!--[if lte IE 6]>
```

```
<link
href="<cms:link>%(link.weak:/system/modules/org.opencms.frontend.templatetwo/resources/cs
s/patch_ie.css:8ffaafcc-e131-11dc-bbcd-3bdd2ea0b1ac)</cms:link>"          rel="stylesheet"
type="text/css">

<![endif]-->
```

En la carpeta shemas hay un fichero que se llama style.xsd que es un constructor de xml, que está ligado con el fichero de /\_config\_/layout/style.css , y le hemos añadido la siguiente línea, para modificar nuestro menú.

```
<layout          element="nav.main"          widget="SelectorWidget"
configuration="value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav
_slidingdoors.jsp'          option='${ key.menu.main.slidingdoors }'          |
value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav_skeleton.jsp'
option='${ key.menu.main.skeleton }'          |
value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav_dolphin.jsp'
option='${ key.menu.main.dolphin }'          |
value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav_style2.jsp'
option='${ key.menu.main.style2 }'          |
value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav_menu.jsp'
option='Menu'          |
value='/system/modules/org.opencms.frontend.templatetwo/elements/menu/nav_vacio.jsp'
option='Vacio' "/>
```

En la carpeta elements/menú hemos creado dos ficheros que se llaman nav\_menu.jsp y nav\_vacio que son dos variables que usamos en style.xsd.

Estructura interna de **nav\_menu**:

```
<% @page          session="false"          import="org.opencms.frontend.templatetwo.*,
org.opencms.file.*"%>

<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<% @ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

```
<% @ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms"%><%
```

```
CmsTemplateMenu cms = new CmsTemplateMenu(pageContext, request, response);
```

```
cms.setElements(cms.getNavigation().getNavigationTreeForFolder(cms.getRequestContext().getUri(), 1, 3));
```

```
    pageContext.setAttribute("cms", cms);
```

```
%>
```

```
<div id="nav_main" class="gradient">
```

```
<c:if test="${ !empty cms.elements} ">
```

```
    <ul>    <c:set var="oldLevel" value="" />
```

```
<c:forEach items="${ cms.elements} " var="elem">
```

```
<c:set var="currentLevel" value="${ elem.navTreeLevel} " />
```

```
<c:choose>
```

```
<c:when test="${ empty oldLevel} "></c:when>
```

```
<c:when test="${ currentLevel > oldLevel} "><ul></c:when>
```

```
<c:when test="${ currentLevel == oldLevel} "></li></c:when>
```

```
<c:when test="${ oldLevel > currentLevel} ">
```

```
<c:forEach begin="${ currentLevel+1} " end="${ oldLevel} "></li></ul></c:forEach>
```

```
</c:when>
```

```
</c:choose>
```

```
</li>
```

```
<a href="<cms:link>${elem.resourceName}</cms:link>" <c:choose><c:when  
test="${fn:startsWith(cms.requestContext.uri, elem.resourceName)}">class="gradient  
current"</c:when><c:otherwise>class="gradient"</c:otherwise></c:choose>>${elem.navText}  
</a>
```

```
<c:set var="oldLevel" value="${currentLevel}" />
```

```
</c:forEach>
```

```
<c:forEach begin="${cms.topLevel+1}" end="${oldLevel}"></li></ul></c:forEach>
```

```
</ul>
```

```
</c:if>
```

```
</div>
```

Estructura interna de **nav\_vacio**

```
<% @page session="false" import="org.opencms.frontend.templatetwo.*,  
org.opencms.file.*"%>
```

```
<% @ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
```

```
<% @ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

```
<% @ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms"%><%
```

```
CmsTemplateMenu cms = new CmsTemplateMenu(pageContext, request, response);
```

```
pageContext.setAttribute("cms", cms);
```

```
%>
```

Si nos vamos a `/_config_/layout/style.css` veremos como están esas dos variables en el desplegable, menú y vacío.

Box Style 3 - Background Color: #FFFFFF

Box Style 3 - Header Background Color: #000000

Box Style 3 - Header Font Color: #FFFFFF

Main Menu: Vacio

Main Menu - Background Color: #FFFFFF

Main Menu - Font Color: #FFFFFF

Main Menu - Background Hover Color: #B31B34

Main Menu - Font Hover Color: #FFFFFF

Main Menu - Background Active Color: #B31B34

Sliding Doors  
Skeleton  
Dolphin  
Style #2  
Menu

"Nuevo" del lado derecho para activar este elemento.

Ahora cuando elijamos una de las dos, por ejemplo vacío se quitará el menú de la cabecera del portal y si elegimos menú aparecerá nuestro propio menú.

## 10.5. AÑADIENDO GOOGLE MAPS AL PORTAL

Primero obtenemos la clave que nos permitirá utilizar google maps con todas sus funciones.

Para conseguirla ve a <http://www.google.com/apis/maps/>. Esta clave que nos proporciona google deberemos insertarla en todas las páginas donde vayamos a utilizar google maps, entre las etiquetas <head></head> de la siguiente forma:

```
<script src="http://maps.google.com/maps?file=api&v=2&oe=ISO-8859-1;&key=ABQIAAAAXfyLOV-DBHsmkpuY-LUUZBRvMuQNe3bCQ9tCDXjXwHZUjgdNBhQG32AJg5mKqo03Qmq9WX7GTGdmvw" type="text/javascript"></script>
```

en key="insertamos la clave que nos ha proporcionado google para nuestra web"

También añadiremos este código: &oe=ISO-8859-1, esto servirá para que muestre los caracteres en castellano y solucionar el problema con las tildes y demás caracteres.

Necesitarás una cuenta de gmail y poner el nombre del dominio donde usaras el mapa, ya que así Google nos da permiso para hacerlo. La clave puede ser sólo usada en páginas de ese dominio que pongas. Creamos en nuestro portal un fichero que se llamará google\_maps.html

Lo editamos como avanzado –editar código de control, no podemos usar el editar normal porque se carga la página y no funcionaría. El contenido de google\_maps.html es este

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<TTArticles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="opencms://system/modules/org.opencms.frontend.templatet
wo.demo/schemas/article.xsd">
```

```
<TTArticle language="en">
```

```
<Title/>
```

```
<Paragraphs>
```

```
<Text name="Text0">
```

```
<links/>
```

```
<content><![CDATA[<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:v="urn:schemas-microsoft-com:vml">
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
```

```
<title>Google Maps JavaScript API Example: Simple Map</title>
```

```
<script
```

```
src="http://maps.google.com/maps?file=api&v=2&sensor=false&key=ABQIAA
AAyBhMi_2T6UvidcODNT8LAhRjr-i8J2oF48VNTChkaK-
XudsyuxTqFzixX_wZJaHjfSvscOlRzWn3Ow"
```

```
type="text/javascript"></script>
```

```
<script type="text/javascript">
```

```
function initialize() {  
  
    if (GBrowserIsCompatible()) {  
  
        var map = new GMap2(document.getElementById("map_canvas"));  
  
        map.setCenter(new GLatLng(37.277604,-5.915623), 13);  
  
        map.setUIToDefault();  
  
    }  
  
}  
  
</script>  
  
</head>  
  
<body onload="initialize()" onunload="GUnload()">  
  
    <div id="map_canvas" style="width: 450px; height: 300px"></div>  
  
</body>  
  
</html>]]></content>  
  
</Text>  
  
</Paragraphs>  
  
</TTArticle>  
  
</TTArticles>
```



En la página principal del portal hemos añadido un enlace a esta página para ver el mapa que apunta a dos hermanas

## 11. CONCLUSIÓN

Este proyecto me ha servido para darme cuenta, lo importante que es este opencms, ya que las ventajas para trabajar en él son muy amplias, ya que al ser código abierto podemos cambiar todo a nuestro antojo, las ventajas que nos ofrece de poder trabajar varias personas en un mismo proyecto, y que cada usuario tenga clara sus limitaciones. Podemos trabajar con diferentes lenguajes a la vez, y albergar diferentes proyectos de otras páginas.

Opencms nos brinda la posibilidad de ser nosotros mismos los administradores y gestores de nuestra aplicación y de forma gratuita, cualquier empresario que quiera traspasar su web a este gestor de contenidos saldrá ganando por todos sitios, ya que una vez que se aprende bien el funcionamiento de opencms, ya no parece tan difícil para el administrador u desarrollador. He aprendido mucho sobre este proyecto, y pocas personas se dedican a esto como profesión, esto me servirá mucho ya que en la calle habrá muy pocas personas que sepan manejar esto, así que me brindará muchas posibilidades de encontrar trabajo mas cómodamente.

