

AGC: Aplicación para la gestión de contenidos de un portal

Autor : Juan Ojeda Duran
Estudios: ETIG
Consultor: Albert Grau Perisé
Fecha: 16-6-2006

AGC

(Aplicación para la gestión de contenidos de un portal)

Proyecto TFC:	AGC: Aplicación para la gestión de contenidos de un portal
Nombre fichero:	Juan_Ojeda_MEMORIA.doc
Creación:	05/04/2006
Modificación:	16/06/2006
Autor:	Juan Ojeda Duran
Versión:	5

Índice

1	Introducción	5
1.1.	Objetivos	6
1.2.	Enfoque y método	6
1.3.	Planificación del proyecto	7
1.4.	Producto obtenido	9
1.5.	Breve descripción de los siguientes capítulos	10
2	Especificación y análisis funcional por subsistemas	11
2.1.	Subsistema 1: Definición de tipos de contenidos.	11
2.1.1.	Pantalla: Gestión del maestro de tipos de bloque	11
2.1.2.	Pantalla: Edición / Creación de tipos de bloque.	12
2.2.	Subsistema 2: Administración de contenidos	13
2.2.1.	Pantalla: Gestión de contenidos.....	13
2.2.2.	Pantalla: Edición / Creación de contenido.....	14
2.3.	Subsistema 3: Acceso y seguridad.	15
2.3.1.	Pantalla: Validación de usuario.....	15
2.3.2.	Pantalla: Listado de opciones de administración.....	16
2.4.	Subsistema 4: Presentación	17
2.4.1.	Frontal Ejemplo1: Noticias de Actualidad	17
2.4.2.	Frontal Ejemplo 2: Preguntas Frecuentes	18
3	Diseño	20
3.1.	Arquitectura del sistema	20
3.2.	Patrones	21
3.3.	Diagramas de casos de uso	23
3.3.1.	Identificación de los actores.....	23

3.3.2.	Casos de uso del Administrador.....	24
3.3.3.	Casos de uso del Usuario:	25
3.4.	Paquetización	26
3.5.	Diagrama de secuencia. Paginación.....	27
3.6.	Diagrama de clases de entidad	28
3.7.	Modelo Físico de Datos	29
4	Conclusiones	30
5	Glosario	31
6	Bibliografía	33
7	Anexo 1.Pantallas de la aplicación. Interficies de Usuario .	34
7.1.1.	Pantalla Index para testeo de funcionalidades	34
7.1.2.	Pantalla de Acceso para administradores	35
7.1.3.	Pantalla de Error de Validación	36
7.1.4.	Pantalla Administración de Tipos de Bloque.....	37
7.1.5.	Pantalla de Alta/Modificar tipo de bloque: Documento Adjunto.....	38
7.1.6.	Pantalla de Alta/Modificar tipo de bloque: Canción	39
7.1.7.	Pantalla Administración de Contenidos Zona1 Noticias	40
7.1.8.	Pantalla Alta/Modificar Noticia	41
7.1.9.	Pantalla Administración de Contenidos Zona2 Preguntas	42
7.1.10.	Pantalla Alta/Modificar Pregunta	43
7.1.11.	Pantalla Frontal 1 Noticias	44
7.1.12.	Pantalla Detalle de Noticia	45
7.1.13.	Pantalla Frontal 2 Preguntas	46

1 Introducción

La idea de crear un CMS (Content Management System) basado en J2EE viene a partir de la propia voluntad de querer profundizar en el análisis, diseño y implementación de este tipo de aplicaciones, dando especial importancia a :

- La creación de **formularios dinámicos** en web.
- La creación de un **modelo de objetos** capaz de formar cualquier tipo contenido para un portal.
- La creación de una **base de servicios** genéricos.
- La creación final de la propia **aplicación AGC**.

Queda además totalmente justificado al contemplar que casi el 90% de este tipo de aplicaciones están programadas en PHP, ¿porque no una en java?.

El reto es conseguir los puntos anteriores y que la aplicación final mediante la implementación del "piloto" pueda ofrecer:

- La administración de la estructura de la información
- La administración de los contenidos
- La presentación de los contenidos (frontales de un portal)

Así que la aplicación tendrá una zona pública o frontales destinados a mostrar los contenidos y otra zona privada destinada a la administración de los mismos o backoffice.

Otras aportaciones del proyecto serán las derivadas de los patrones (DAO,MVC,Factoria,Singleton...) utilizados y del modelo de programación basado en 4 capas, esta ofrecerá una serie de mejoras en cuanto a mantenibilidad, extensibilidad y reutilización de componentes.

- **Capa de cliente:** Compuesta por el Interfaz de usuario.
- **Capa de presentación:** Compuesta por el conjunto de vistas o componentes que recuperarán la información a mostrar en el interfaz del cliente.
- **Capa de lógica de negocio:** Compuesta por los objetos de la lógica del negocio que serán reutilizables.
- **Capa de datos:** Compuesta por la capa de persistencia y los datos de la aplicación almacenados en una base de datos relacional.

1.1. *Objetivos*

Los objetivos principales del TFC son el análisis, el diseño, implementación y test de una aplicación para la gestión de contenidos (AGC).

Con especial énfasis se ha trabajado en la consecución de:

- Una base de servicios genéricos para la creación de aplicaciones web que usen el *framework* Struts y el motor de persistencia Hibernate, son:
 - Gestor pool de conexiones
 - Gestor de sesiones
 - Gestor genérico de persistencia
 - Carga de constantes
 - Carga de parámetros
 - Carga de literales
 - Sistema de paginación
 - Sistema de log
 - Sistema de inicialización de aplicaciones

- Un modelo de objetos capaz de representar cualquier contenido, comenzando desde los componentes más básicos (documento adjunto, imagen, texto y link) hasta llegar a los contenidos más complejos.
 - (componente → bloque → contenido)

- Una aplicación de backoffice parametrizable y reutilizable a partir del uso de formularios dinámicos para administrar cualquier tipo de contenido.
 - Administración de tipos de bloque
 - Administración de contenidos
 - Administrar Noticias
 - Administrar Preguntas

- Una muestra dos frontales diferentes que podrían aparecer en un supuesto portal.
 - Frontal 1: Noticias de Actualidad
 - Frontal 2: Preguntas Frecuentes

1.2. *Enfoque y método*

El enfoque utilizado en durante el proyecto ha sido la de mantener una disciplina de dedicación, es decir mantener durante todas las fases una constancia de trabajo, aún así y según el conocido "Síndrome del Estudiante" ha sido un reto el llegar a las últimas entregas.

1.3. Planificación del proyecto

La temporalización inicial para el proyecto fue:

Proyecto AGC		
Fase	Fecha Finalización	Entregable
Inicio Proyecto	4 de marzo de 2006	
Plan de Trabajo	13 de marzo de 2006	Si
Especificación y análisis (10d)	23 de marzo de 2006	
Especificación y funcionales (14d)	6 de abril de 2006	
Diseño (15d)	21 de abril de 2006	Si
Implementación (40d)	31 de mayo de 2006	Si
Pruebas (4d)	4 de junio de 2006	
Memoria - Entrega Final (12d)	16 de junio de 2006	Si

El detalle de actividades por semanas:

AGC - FASE DE IMPLEMENTACIÓN - MAYO-JUNIO 2006					
Actividad	Semana 1	Semana 2	Semana 2	Semana 3	Semana 4
Estudio software base					
Instalación software base					
Eclipse					
Tomcat					
Oracle					
Servicios compartidos					
Convención de nombres					
Carga de constantes					
Carga de parámetros					
Sistema de log					
Sistema de inicialización					
Motor de Persistencia Hibernate					
Configurar factoría					
Gestor pool de conexiones					
Gestor de sesiones					
Gestor genérico de persistencia					
Definición del modelo hbm's					
Creación de beans persistentes					
Framework Struts					
Instalación y configuración					
Ejercicio y pruebas de formularios dinámicos			X		
Subsistema 1: Definición de tipos de bloque			X		
Pantalla: Gestión del maestro de tipos de bloque			X		
Test y pruebas formulario dinámico tipos de bloque			X		
Creación capa controlador y vista (Action/Form/Jsp)			X		
Creación capa de persistencia (Helper/Persistence)			X		
Pantalla: Edición / Creación de tipos de bloque			X		
Creación capa controlador y vista (Action/Form/Jsp)			X		
Creación capa de persistencia (Helper/Persistence)			X		
Subsistema 2: Administración de contenidos.				X	

Pantalla: Gestión de contenidos.				X	
Creación capa controlador y vista (Action/Form/Jsp)				X	
Creación capa de persistencia (Helper/Persistence)				X	
Pantalla: Edición / Creación de contenido				X	
Creación capa controlador y vista (Action/Form/Jsp)				X	
Creación capa de persistencia (Helper/Persistence)				X	
Subsistema 3: Acceso y seguridad.					X
Pantalla: Validación de usuario					X
Creación capa controlador y vista (Action/Form/Jsp)					X
Creación capa de persistencia (Helper/Persistence)					X
Pantalla: Listado de opciones de administración					X
Creación capa controlador y vista (Action/Form/Jsp)					X
Creación capa de persistencia (Helper/Persistence)					X
Subsistema 4: Presentación.					X
Frontal Ejemplo1: Noticias de Actualidad					X
Creación capa controlador y vista (Action/Form/Jsp)					X
Creación capa de persistencia (Helper/Persistence)					X
Frontal Ejemplo2: Preguntas Frecuentes					X
Creación capa controlador y vista (Action/Form/Jsp)					X
Creación capa de persistencia (Helper/Persistence)					X
Test y pruebas piloto					X
Documentación					X
Generar Javadoc					X
Crear leeme.txt,howToInstall.txt,database.sql...					X
Crear Memoria					X
Crear Presentación					X

Con una X marco las actividades que han sufrido desajustes. En general las fases que finalmente han sufrido más desajustes han sido la de diseño e implementación y es debido a parte del volumen de trabajo a la curva de aprendizaje especialmente realizada y ampliada en:

Mapping de objetos en Hibernate (hbm's)
 Gestor pool de conexiones para Hibernate
 Gestor de sesiones Hibernate (session-per-request, filtros)
 Gestor genérico de persistencia Hibernate
 Creación de la capa controlador (struts actions)
 Creación de la capa de de lógica de negocio (helpers)
 Creación de la capa de de persistencia (persistence managers)
 Creación de la capa de presentación / vista a partir de formularios dinámicos JSTL (JSP Standard Tag Library (JSTL))

1.4. Producto obtenido

El producto obtenido finalmente es la aplicación AGC implementada junto con el código fuente y la documentación.

La entrega se realiza en dos ficheros comprimidos .zip

El primero JuanOjedaSOURCECODE.zip que contiene:

Las carpetas:

/javadoc:

Contiene la documentación de todo el código fuente de la aplicación

/para_tomcat:

Contiene el plugin para eclipse para usar tomcat (tomcatPluginV3beta.zip)

Contiene el parche para tomcat para usarlo con J2SE menores que la 5.0 (jakarta-tomcat-5.5.9-compatible.zip)

Contiene el fichero de contexto (agc.xml)

Los ficheros:

projecte_eclipse_agc.zip:

Contiene el proyecto completo de eclipse comprimido.

howToInstall.txt:

Describe los requerimientos software y la manera de instalar y ejecutar la aplicación a partir del projecte_eclipse_agc.zip

database.sql:

Contiene el script sql para la creación de tablas en la base de datos

Contiene el script sql para la inserción de los datos por defecto

El segundo JuanOjedaDOCUMENTACION.zip que contiene:

JuanOjedaMEMORIA.doc

JuanOjedaPRESENTACION.ppt

Véase al Anexo 1. Donde se muestran todas las pantallas de la aplicación implementadas con contenidos de ejemplo

1.5. Breve descripción de los siguientes capítulos

En el siguiente capítulo el punto 2 "Especificación y análisis funcional por subsistemas", veremos el análisis funcional de cada subssistema:

Subsistema 1: Definición de tipos de contenidos.

Subsistema 2: Administración de contenidos.

Subsistema 3: Acceso y seguridad.

Subsistema 4: Presentación. Frontales.

En el punto 3 Diseño veremos:

Arquitectura del sistema

Patrones

Diagramas de casos de uso

Paquetización

Diagrama de clases de entidad

Modelo Físico de Datos

y finalmente los últimos capítulos las conclusiones, el glosario de términos y la bibliografía.

Se incluye un Anexo final donde se listan las pantallas resultantes de la implementación.

2 Especificación y análisis funcional por subsistemas

2.1. Subsistema 1: Definición de tipos de contenidos.

2.1.1. Pantalla: Gestión del maestro de tipos de bloque

Esta gestión permite administrar los diferentes tipos de bloque que podrán usarse en la gestión de contenidos.

En esta pantalla se presentarán los diferentes tipos de bloque, permitiendo gestionarlos. El aspecto de la pantalla será como sigue:

Listado de tipos de bloque		
Tipo de bloque 1	E	B
Tipo de bloque 2	E	B
Tipo de bloque 3	E	B

<- Anterior

Siguiente ->

Alta

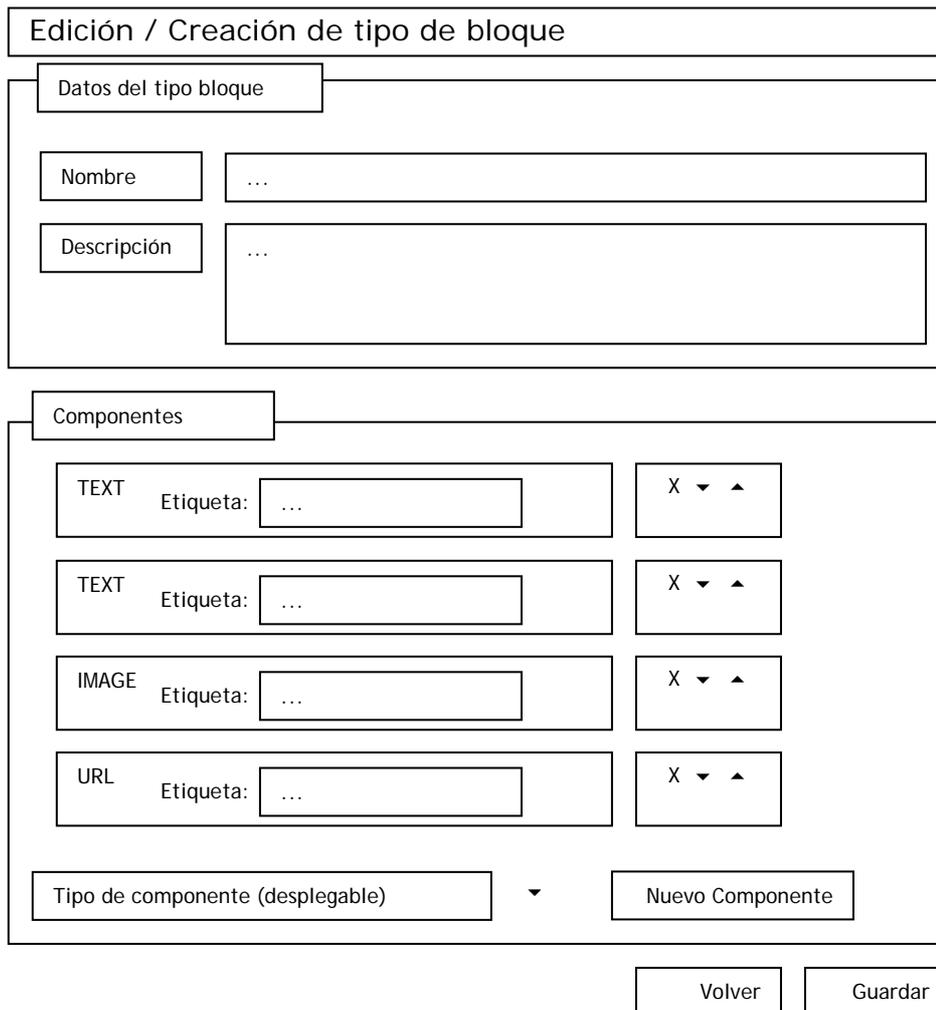
El comportamiento de la ventana será el siguiente:

- En la primera carga, se presentarán todos los tipos de bloque por orden alfabético, paginados de 10 en 10.
- Para cada tipo de bloque se podrá realizar las siguientes acciones:
 - o **E**: Editar. Llevará a la pantalla de edición de tipos de bloque.
 - o **B**: Borrar. Tras un alert de confirmación, borrará el tipo de bloque, avisando en caso de que existan contenidos que usen ese tipo de bloque..
- Pulsando al botón **Alta**, se presentará la pantalla de creación de nuevo tipo de bloque.

2.1.2. Pantalla: Edición / Creación de tipos de bloque.

Esta pantalla presentará la creación o la edición de un tipo de bloque. Un tipo de bloque estará formado su nombre, descripción y por un conjunto variable de componentes.

El aspecto de la ventana será como sigue:



Listado variable de componentes.

Donde el comportamiento de la ventana será el siguiente:

- Para el tipo de bloque será posible informar el Nombre, la Descripción y añadir de 1 a n componentes.
- La caja de componentes es dinámica, será posible añadir de 1 a n componentes seleccionando el tipo deseado en el desplegable y pulsando el botón **Nuevo Componente**.
- Los componentes pueden ser del tipo: (TEXT: Texto, IMAGE: Imagen, DOCUMENT: Documento Adjunto o URL: Link),
- Se permitirá modificar el orden (botones arriba / abajo) o borrar un componente (botón X).
- Para cada componente será posible indicar su etiqueta personalizada.

- El botón **Guardar** almacenará los cambios y **Volver** retornará al listado.

2.2. Subsistema 2: Administración de contenidos.

2.2.1. Pantalla: Gestión de contenidos.

En esta ventana se presentará el listado de todos los contenidos asociados a la zona del frontal administrado. El aspecto de la pantalla será como:

Búsqueda	<input type="text"/>	Buscar
----------	----------------------	--------

Listado de contenidos				
Fecha	Titulo	Autor	E	B
Fecha	Titulo	Autor	E	B
Fecha	Titulo	Autor	E	B
Fecha	Titulo	Autor	E	B

<- Anterior Siguiente ->

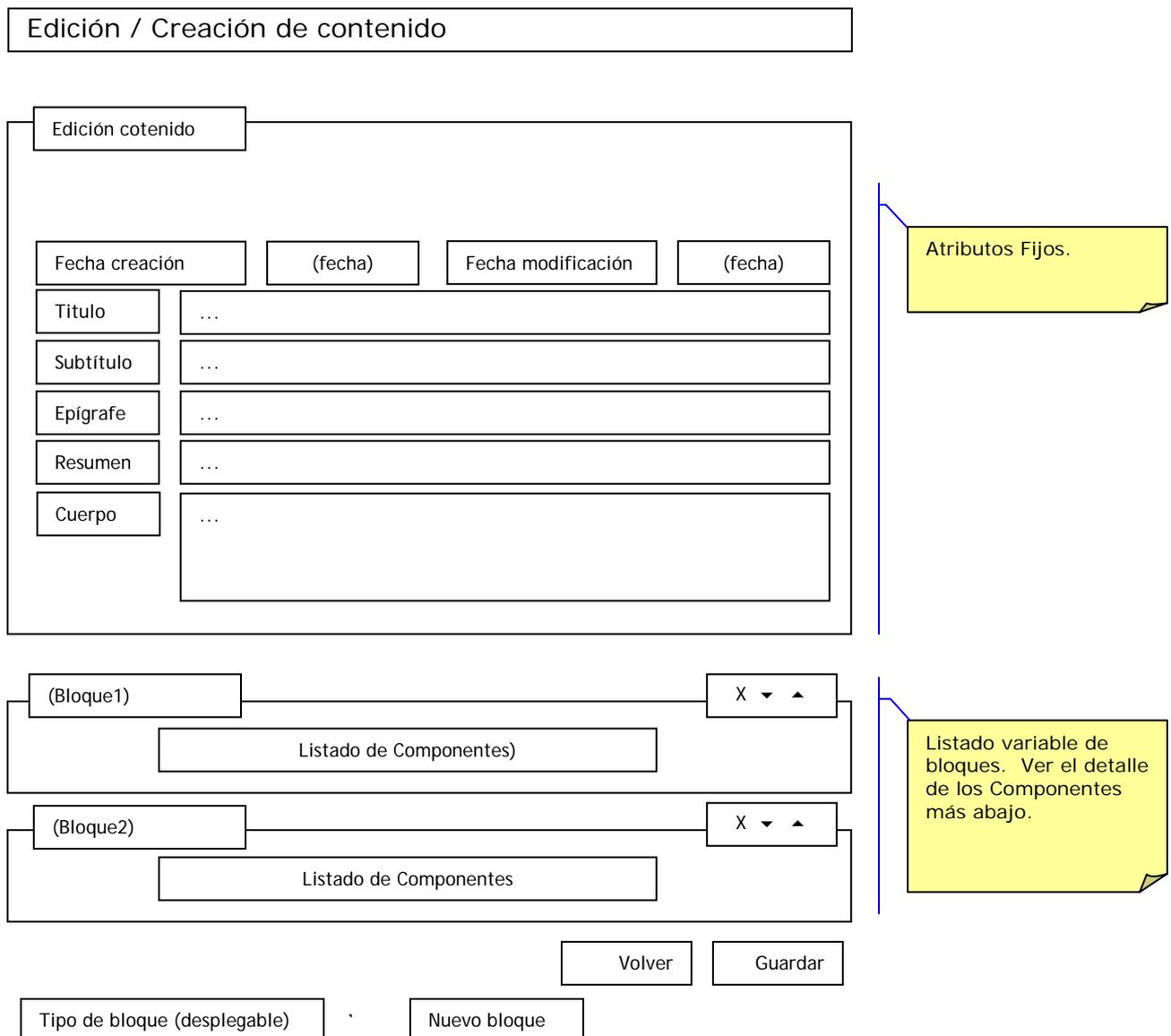
El comportamiento de la ventana será como sigue:

- En la primera carga, se presentarán todos los contenidos, ordenados por fecha descendiente (primero la más reciente). Dentro de la misma fecha, el orden será por autor + título
- Pulsando en la cabecera de cada uno de los campos, se reordenará la lista. Se paginará de 10 en 10.
- Introduciendo una cadena en el campo de búsqueda y pulsando **Buscar** se filtrarán las noticias por aquellas que en título, descripción contengan la cadena introducida.
- Para cada contenido se podrá realizar las siguientes acciones:
 - o **E**: Editar. Llevará a la pantalla de edición de contenidos.
 - o **B**: Borrar. Tras un alert de confirmación, borrará el contenido..
- Pulsando el botón **Alta**, se presentará la pantalla de creación de nuevo contenido.

2.2.2. Pantalla: Edición / Creación de contenido

Esta pantalla presenta la creación/edición del contenido. Un contenido estará formado por un conjunto fijo de atributos y por un conjunto variable de bloques.

El aspecto de la ventana será como sigue:



Edición / Creación de contenido

Edición contenido

Fecha creación (fecha) Fecha modificación (fecha)

Titulo ...

Subtítulo ...

Epígrafe ...

Resumen ...

Cuerpo ...

(Bloque1) X ▼ ▲

Listado de Componentes

(Bloque2) X ▼ ▲

Listado de Componentes

Volver Guardar

Tipo de bloque (desplegable) Nuevo bloque

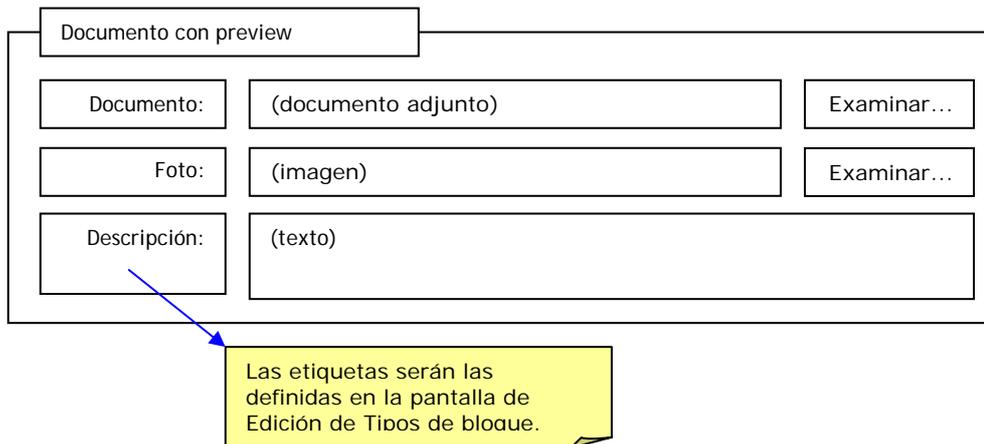
Atributos Fijos.

Listado variable de bloques. Ver el detalle de los Componentes más abajo.

Donde el comportamiento de la ventana será como sigue:

- Para el contenido será posible informar los atributos fijos y añadir de 0 a n bloques.
- La caja de bloques es dinámica, será posible añadir de 0 a n bloques seleccionando el tipo de bloque en el desplegable y pulsando el botón **Nuevo Bloque**.

- Se permitirá modificar el orden de los bloques (botones arriba / abajo) o borrar un bloque (botón X).
- El botón **Guardar** almacenará los cambios y **Volver** retornará al listado.
- Un ejemplo de "**Componentes**" puede ser "Documento con preview", en que la caja del bloque1 tendría un aspecto similar al que sigue:



Documento con preview

Documento:	(documento adjunto)	Examinar...
Foto:	(imagen)	Examinar...
Descripción:	(texto)	

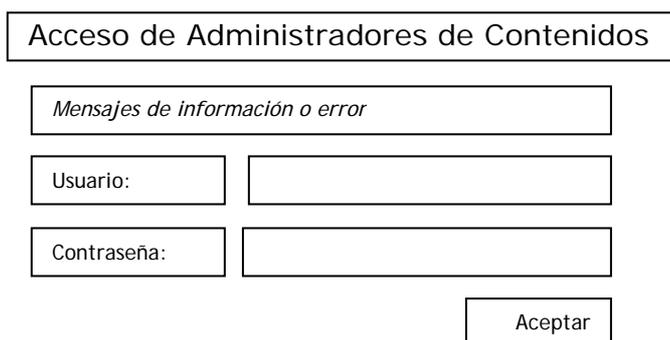
Las etiquetas serán las definidas en la pantalla de Edición de Tipos de bloque.

2.3. Subsistema 3: Acceso y seguridad.

2.3.1. Pantalla: Validación de usuario

Esta pantalla mostrará los campos necesarios para la validación de los administradores de contenidos.

El aspecto de la ventana será el siguiente:



Acceso de Administradores de Contenidos

Mensajes de información o error

Usuario:

Contraseña:

Aceptar

El comportamiento de la ventana será como sigue:

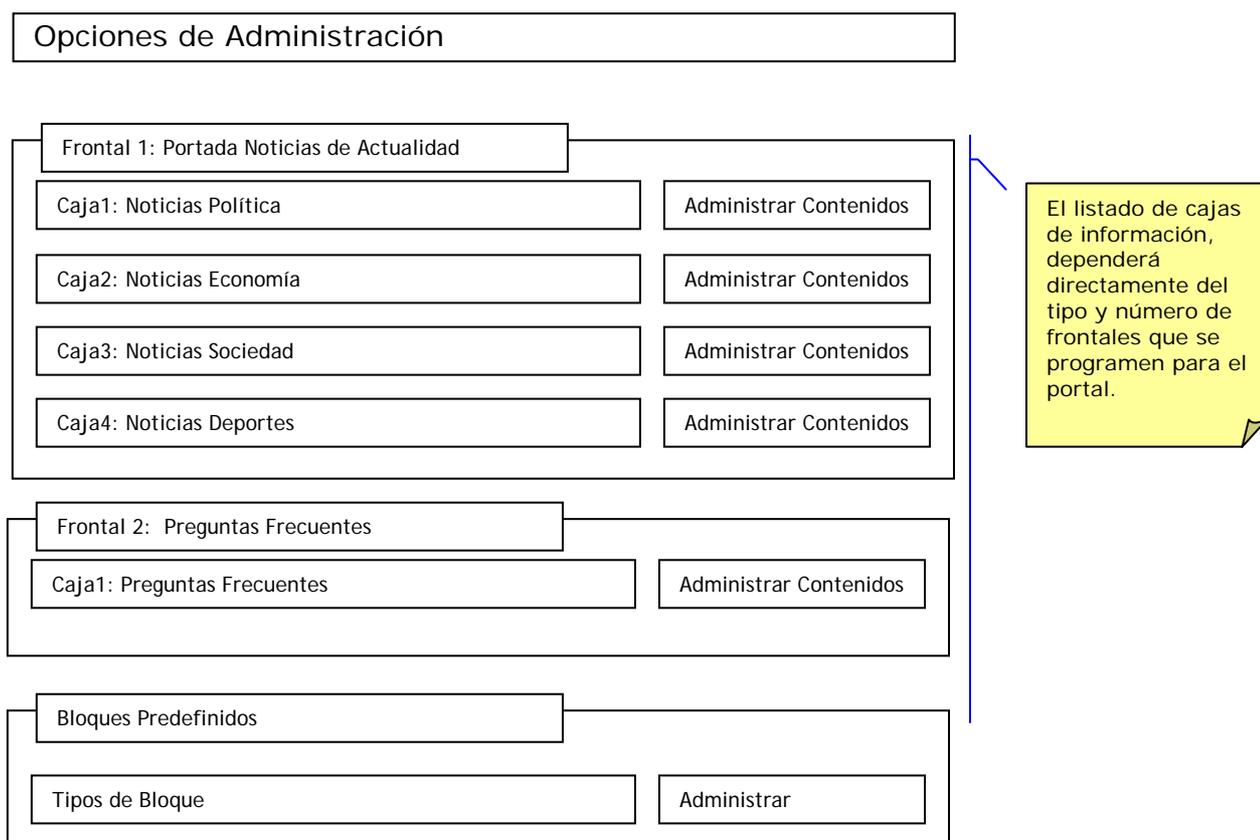
- El usuario introducirá su nombre de usuario y su contraseña, para finalizar pulsará en **Aceptar**.
- Si la validación no es correcta se informará al usuario mediante el cuadro de mensajes.

- Si la validación es correcta se le presentará al administrador las opciones de administración.

2.3.2. Pantalla: Listado de opciones de administración

Esta pantalla mostrará todas las opciones de administración a las que tenga acceso el usuario. Se mostrará el listado de frontales programados del portal y por cada uno de ellos se mostrará el listado de cajas de información definidas.

El aspecto de la ventana para la administración de un frontal será como sigue:



El comportamiento de la ventana será como sigue:

- El administrador gestionará los contenidos de un frontal pulsando en el botón de **Administrar Contenidos** de la caja deseada.
- El administrador podrá gestionar los tipos de bloque pulsando en el botón **Administrar**.

2.4. Subsistema 4: Presentación.

Para mostrar la versatilidad del sistema se detallan a continuación dos posibles frontales que presentarán el contenido al usuario final. Para nuevos frontales solo hará falta realizar el mismo ejercicio de definir los formatos deseados para cada caja.

2.4.1. Frontal Ejemplo1: Noticias de Actualidad

Portal AGC : Noticias de Actualidad																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left; padding: 5px;">Política</th> </tr> </thead> <tbody> <tr> <td style="width: 70%; padding: 5px;">Título Noticia 1</td> <td style="width: 30%; padding: 5px;">Fecha</td> </tr> <tr> <td style="width: 30%; padding: 5px;">Imagen Noticia 1 IMG</td> <td style="width: 70%; padding: 5px;">Resumen Noticia 1 Documento Adjunto</td> </tr> <tr> <td style="padding: 5px;">Título Noticia 2</td> <td style="padding: 5px;">Fecha</td> </tr> <tr> <td style="padding: 5px;">Imagen Noticia 2 IMG</td> <td style="padding: 5px;">Resumen Noticia 2 Documento Adjunto</td> </tr> </tbody> </table>	Política		Título Noticia 1	Fecha	Imagen Noticia 1 IMG	Resumen Noticia 1 Documento Adjunto	Título Noticia 2	Fecha	Imagen Noticia 2 IMG	Resumen Noticia 2 Documento Adjunto	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left; padding: 5px;">Economía</th> </tr> </thead> <tbody> <tr> <td colspan="2" style="padding: 5px;">Título Noticia 1</td> </tr> <tr> <td style="width: 40%; padding: 5px;">Fecha</td> <td style="width: 60%; padding: 5px;">Autor</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Resumen Noticia 1</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Título Noticia 2</td> </tr> <tr> <td style="padding: 5px;">Fecha</td> <td style="padding: 5px;">Autor</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Resumen Noticia 2</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Título Noticia 3</td> </tr> <tr> <td style="padding: 5px;">Fecha</td> <td style="padding: 5px;">Autor</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Resumen Noticia 3</td> </tr> </tbody> </table>	Economía		Título Noticia 1		Fecha	Autor	Resumen Noticia 1		Título Noticia 2		Fecha	Autor	Resumen Noticia 2		Título Noticia 3		Fecha	Autor	Resumen Noticia 3					
Política																																			
Título Noticia 1	Fecha																																		
Imagen Noticia 1 IMG	Resumen Noticia 1 Documento Adjunto																																		
Título Noticia 2	Fecha																																		
Imagen Noticia 2 IMG	Resumen Noticia 2 Documento Adjunto																																		
Economía																																			
Título Noticia 1																																			
Fecha	Autor																																		
Resumen Noticia 1																																			
Título Noticia 2																																			
Fecha	Autor																																		
Resumen Noticia 2																																			
Título Noticia 3																																			
Fecha	Autor																																		
Resumen Noticia 3																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left; padding: 5px;">Sociedad</th> </tr> </thead> <tbody> <tr> <td style="width: 70%; padding: 5px;">Título Noticia 1</td> <td style="width: 30%; padding: 5px;">Fecha</td> </tr> <tr> <td style="padding: 5px;">Imagen Noticia 1 IMG</td> <td></td> </tr> <tr> <td colspan="2" style="padding: 5px;">Epígrafe Noticia 1</td> </tr> <tr> <td colspan="2" style="padding: 5px;">Cuerpo Noticia 1</td> </tr> <tr> <td colspan="2" style="padding: 5px;"> Urls asociadas: LINK1 LINK2 LINK3 </td> </tr> </tbody> </table>	Sociedad		Título Noticia 1	Fecha	Imagen Noticia 1 IMG		Epígrafe Noticia 1		Cuerpo Noticia 1		Urls asociadas: LINK1 LINK2 LINK3		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: left; padding: 5px;">Deportes</th> </tr> </thead> <tbody> <tr><td colspan="2" style="padding: 5px;">Título Noticia 1</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 2</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 3</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 4</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 5</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 6</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 7</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 8</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 9</td></tr> <tr><td colspan="2" style="padding: 5px;">Título Noticia 10</td></tr> </tbody> </table>	Deportes		Título Noticia 1		Título Noticia 2		Título Noticia 3		Título Noticia 4		Título Noticia 5		Título Noticia 6		Título Noticia 7		Título Noticia 8		Título Noticia 9		Título Noticia 10	
Sociedad																																			
Título Noticia 1	Fecha																																		
Imagen Noticia 1 IMG																																			
Epígrafe Noticia 1																																			
Cuerpo Noticia 1																																			
Urls asociadas: LINK1 LINK2 LINK3																																			
Deportes																																			
Título Noticia 1																																			
Título Noticia 2																																			
Título Noticia 3																																			
Título Noticia 4																																			
Título Noticia 5																																			
Título Noticia 6																																			
Título Noticia 7																																			
Título Noticia 8																																			
Título Noticia 9																																			
Título Noticia 10																																			

Esta pantalla es un frontal propuesto como ejemplo para el portal. Es un caso práctico donde podemos ver las noticias separadas en 4 zonas o cajas informativas. Cada una de ellas presentará la información de forma diferente, se proponen cuatro formatos:

Formato de la Caja1: Política

Presentará las dos últimas noticias publicadas.

Se mostrarán los campos **Título**, **Fecha** y **Resumen**. Se cargará el primer componente **Imagen** y un posible componente **Documento Adjunto**.

Formato de la Caja2: Economía

Presentará las tres últimas noticias publicadas.

Se mostrarán los campos **Título**, **Fecha**, **Autor** y **Resumen**.

No se admitirán componentes.

Formato de la Caja3: Sociedad

Presentará la última noticia publicada.

Se mostrarán los campos **Título**, **Fecha**, **Epígrafe** y **Cuerpo**.

Se cargará el primer componente **Imagen** y el listado de componentes de tipo **URL**.

Formato de la Caja4: Deportes

Presentará las diez últimas noticias publicadas.

Se mostrará el campo **Título**.

No se admitirán componentes.

2.4.2. *Frontal Ejemplo 2: Preguntas Frecuentes*

Portal AGC : Preguntas Frecuentes

<p>Pregunta:</p> <p>Pregunta 1: Pregunta</p> <p>Respuesta 1: Respuesta</p> <p>Url asociada: LINK1 Documento adjunto: Documento Adjunto</p>
<p>Pregunta:</p> <p>Pregunta 2:</p> <p>Respuesta 2:</p> <p>Url asociada: LINK1 Documento adjunto: Documento Adjunto</p>

Esta pantalla es un segundo ejemplo de frontal. Es un caso práctico donde podemos ver el listado de preguntas con sus respuestas. La información se mostrará dependiendo del formato propuesto:

Formato de la Caja: Pregunta

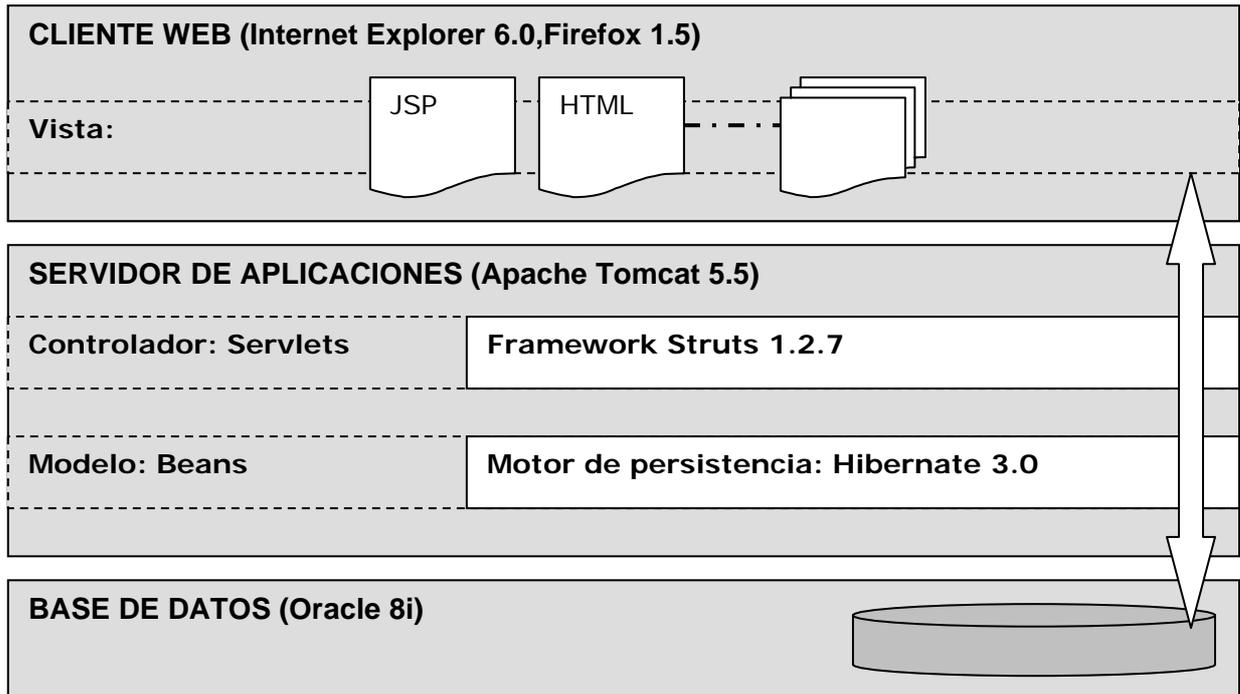
Se mostrará el campo **Pregunta** y **Respuesta**.

Se admitirá un componente **Documento Adjunto** y un componente **URL** por pregunta.

3 Diseño

3.1. Arquitectura del sistema

El siguiente esquema presenta la arquitectura del sistema indicando las capas y las versiones de los productos que se han utilizado en cada una de ellas.



3.2. Patrones

GRASP es el acrónimo de General Responsibility Assignment Software Patterns. Una de las cosas más complicadas en Orientación a Objeto consiste en elegir las clases adecuadas y decidir como estas clases deben interactuar. En el desarrollo de la aplicación se ha tenido en cuenta una serie de recomendaciones para la elección de los patrones son:

Bajo Acoplamiento

Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia.

Como ejemplo (de mal diseño), en muchos diseños Web se puede ver como se crea un servlet base con capacidades de paginación y se hereda de él para construir los demás. La paginación es un servicio que se podría usar en aplicaciones no Web, por lo que es más adecuado mantener estas capacidades en clases externas.

Otro ejemplo clásico se produce cuando se pasan los objetos relacionados con la capa de presentación a la capa de negocio (HttpRequest, HttpResponse).

Alta Cohesión

Cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable.

Ejemplos de una baja cohesión son clases que hacen demasiadas cosas.

En todas las metodologías se considera la refactorización. Uno de los elementos a refactorizar son las clases saturadas de métodos.

Ejemplos de buen diseño se producen cuando se crean los denominados "paquetes de servicio" o clases agrupadas por funcionalidades que son fácilmente reutilizables (bién por uso directo o por herencia).

Experto

La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos) . Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada

Creador

El hecho de crear objetos tiene casuísticas particulares:

- Pool de Objetos
- Caches
- Instancias únicas

Estos casos son candidatos para la utilización de otros patrones más concretos (de diseño). En caso de java:

- o No confundir referencias y el valor referenciado a la hora de retornar objetos desde métodos.
- o Conocer la peculiaridad de las String (cadenas constantes)
- o Identificar problemáticas del recolector de basura y el uso de recursos del sistema
- o Conocer el ámbito y naturaleza de distintos tipos de componentes:
 - En servlets y JSPs pueden descargarse y estar activos en más de una máquina virtual.
 - Los servlets y JSPs son multi-thread a priori.
 - Los tags son reciclados en JSP (ojo con valores anteriores de atributos de clase).

Controlador

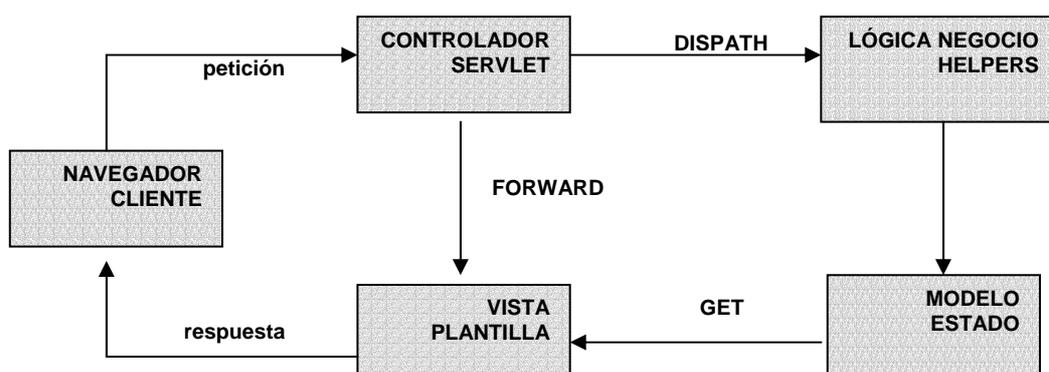
Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.) . El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema

Patrón MVC

En especial la aplicación sigue el patrón (MVC) gracias a Struts que permite separar perfectamente la vista (presentación), el modelo (reglas de negocio) y el controlador (dispatcher).

El siguiente diagrama muestra el flujo de llamadas entre componentes para el patrón MVC:



Patrón DAO

Para capa de persistencia se ha utilizado el patrón DAO (Data Access Objects), que permite el alta, la baja y la modificación del modelo de beans (POJO's) del sistema

Patrón Factoria

Para la obtención de los diferentes managers de persistencia de cada bean se ha creado una factoria DAO es decir una única clase del sistema que es capaz de ofrecer los managers de persistencia. Así queda centralizado y aislamos la capa de lógica de negocio (helpers) de la capa de persistencia (persistences).

3.3. Diagramas de casos de uso

3.3.1. Identificación de los actores

Administrador: Es el responsable de administrar los contenidos del portal.

Puede:

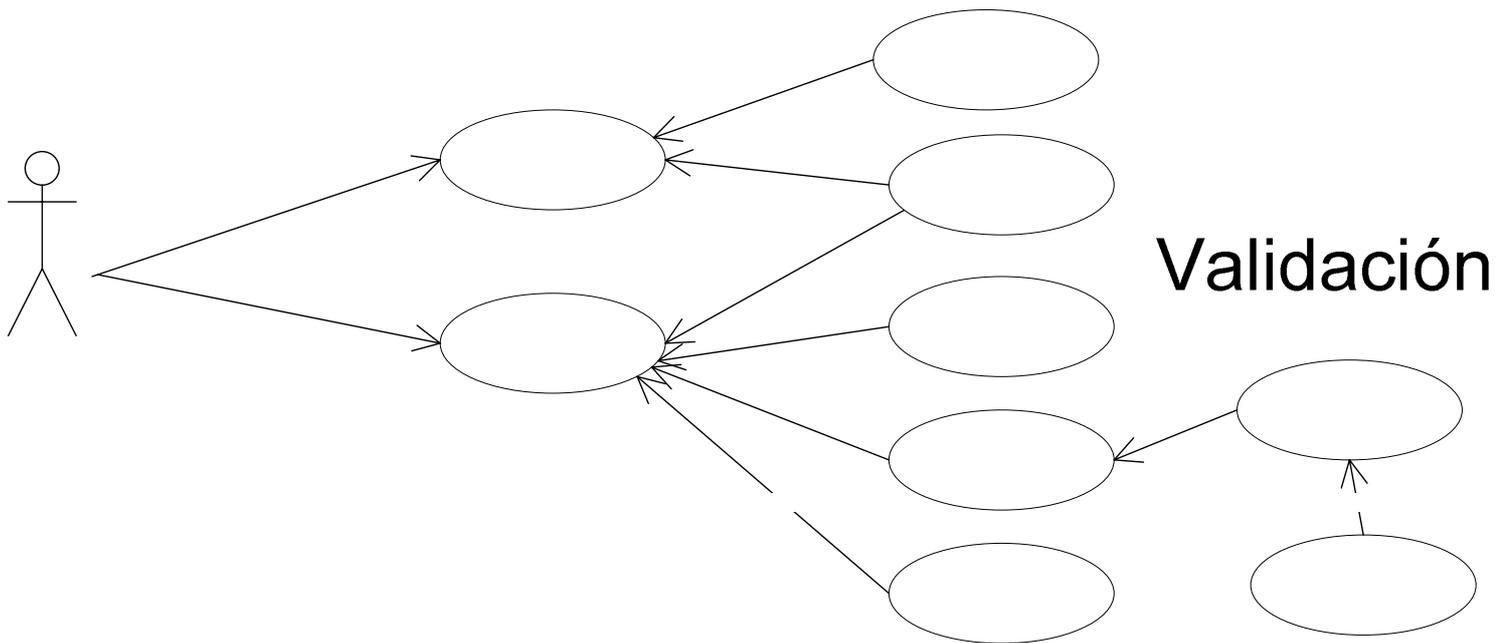
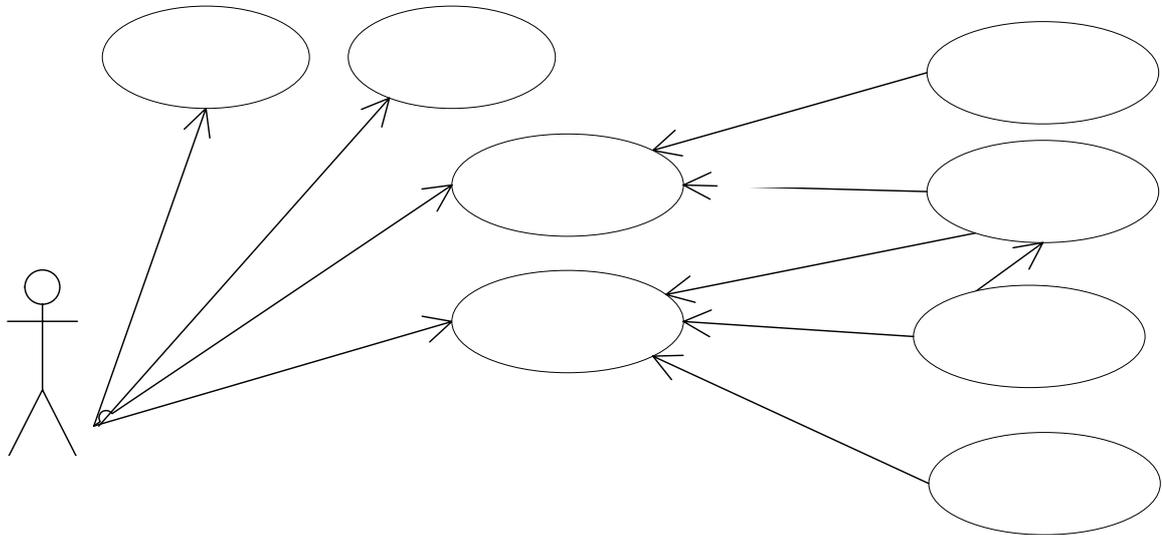
- Validarse y acceder al índice de opciones de administración
- Crear tipos de bloque
- Crear contenidos
- Buscar contenidos
- Ver el detalle de un contenido

Usuario: Es el usuario final que navegando de forma anónima puede visualizar los contenidos en los diferentes frontales programados del portal

Puede:

- Listar los contenidos de los frontales
- Ver el detalle de un contenido

3.3.2. Casos de uso del Administrador



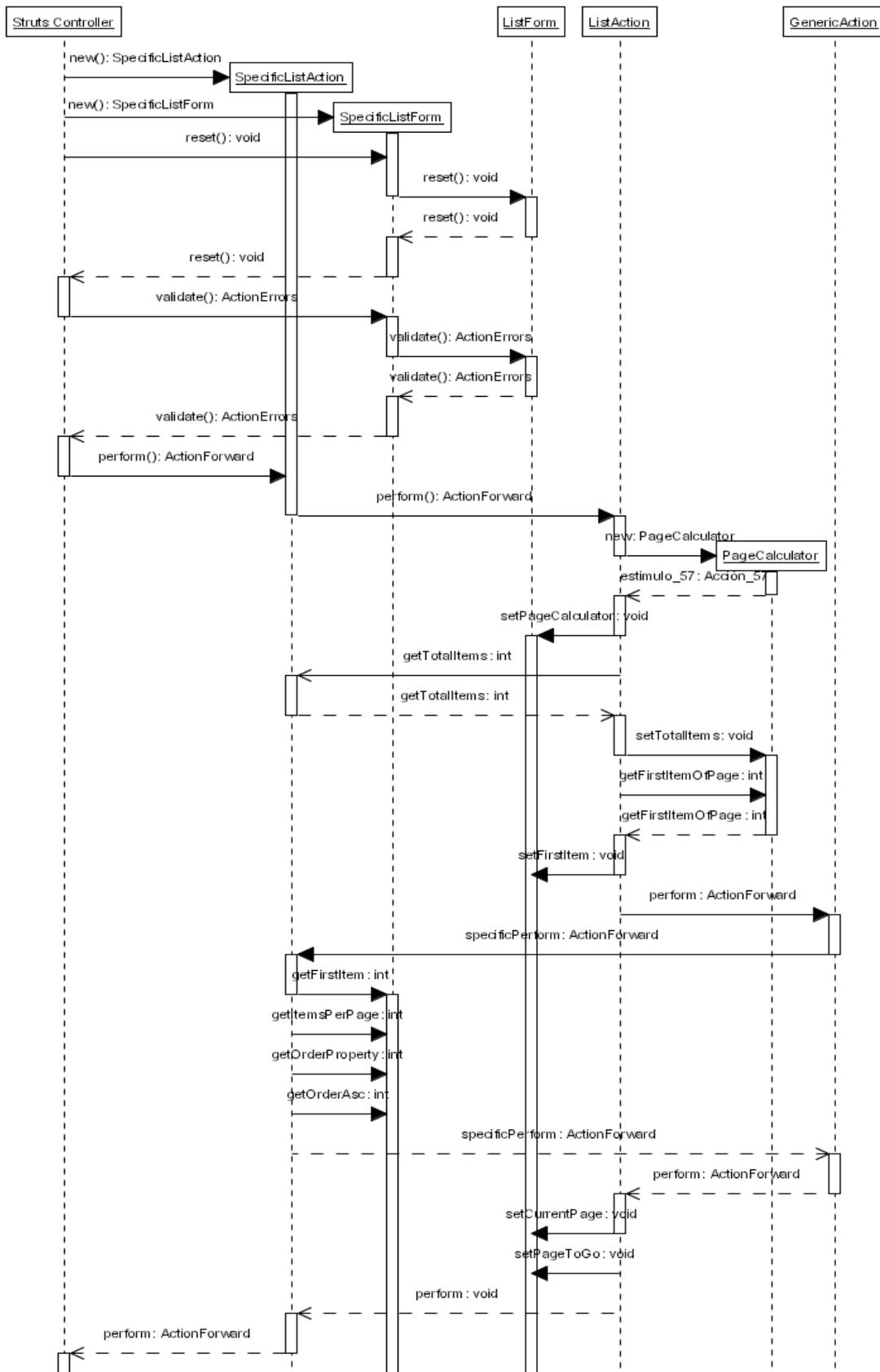
3.4. Paquetización

A continuación se listan la serie de paquetes más importantes que forman la aplicación:

agc.init	Clases para la inicialización de la aplicación
agc.cfg	Clases para la configuración y parametrización
agc.excepcion	Clases para el tratamiento de errores o excepciones
agc.log	Clases para el tratamiento del log
agc.pool.hibernate	Clases para la gestión de I pool, conexiones y gestor genérico de persistencia para hibernate
agc.service.action	Clases para las acciones de Struts. (Controlador)
agc.service.bean	Clases para los beans y formularios de Struts (Vista)
agc.service.helper	Clases para la lógica de negocio
agc.service.bean	Clases para el modelo de objetos hibernate.
agc.service.persistence	Clases para la capa de persistencia
agc.util	Clases de utilidad para el tratamiento de ficheros literales, cadenas de texto, fechas y paginación

Los paquetes en azul forman el conjunto de servicios compartidos.

3.5. Diagrama de secuencia. Paginación



4 Conclusiones

Personalmente mi conclusión sobre el proyecto es que la experiencia durante todo el proceso ha sido muy positiva y enriquecedora, principalmente por todo lo aprendido sobre:

Las diferentes capas que se necesitan para la creación de aplicaciones basadas en tecnología J2EE.

Los diferentes servicios compartidos que se pueden reutilizar en cualquier aplicación.

El diseño de aplicaciones CMS y con especial énfasis en el modelado de los contenidos.

Los patrones a aplicar para desarrollar software de calidad.

Formularios dinámicos con JSTL y la capacidad de aumentar y disminuir el número campos administrables en tiempo de ejecución y como persistirlo.

La persistencia de objetos mediante mapeos y transformaciones directas de objetos java a la base de datos.

A parte ha sido un éxito la consecución final de los objetivos planteados inicialmente, y además haber podido ejercer el rol de "Jefe de Proyecto", llevando así la dirección, el control y la planificación de cada fase.

5 Glosario

CMS: Un Sistema de gestión de contenido (Content Management System, en inglés, abreviado CMS) es una aplicación que permite la creación y administración de contenidos principalmente en páginas web.

STRUTS: Es un framework para el desarrollo de aplicaciones web que implementa el patrón de diseño MVC (Model View Control) el cual se ha convertido en un estándar muy usado en el mundo del desarrollo en Java

HIBERNATE: Hibernate es un motor de persistencia open-source que hace sencillo el trabajo con bases de datos relacionales. Simula al desarrollador como si la base de datos trabajara directamente con objetos (POJO's), lo que lleva a que el desarrollador se preocupe por el negocio de la aplicación y no del cómo se persistirá/recuperará/eliminará la información de la base de datos.

JSTL: La librería JSTL es un componente dentro de la especificación del Java 2 Enterprise Edition (J2EE) y es controlada por Sun Microsystems. JSTL no es más que un conjunto de librerías de etiquetas simples y estándares que encapsulan la funcionalidad principal que es usada comúnmente para escribir páginas JSP.

MVC: (Model-View-Controller) es un patrón que define la organización independiente del Model (Objetos de Negocio), la View (interfaz con el usuario u otro sistema) y el Controller (controlador del workflow de la aplicación: "si estoy aquí y me piden esto entonces hacer tal cosa, si sale bien mostrar esto y sino lo aquello otro").

Para la aplicación AGC:

Administrador: persona encargada de administrar los contenidos del portal. Deberá validarse y acceder a las opciones de administración.

Usuario: persona que navega por un portal consultando contenidos públicos

Componente: Unidad más básica que formar un tipo de bloque, son Documentos adjuntos, imágenes, textos o urls.

Tipo de bloque: Es un conjunto de componentes que al agruparlos ya representan algo con cierto significado. Por ejemplo una ficha para representar los datos de una canción, o de una película o una noticia, etc.

Contenido: Es un conjunto de tipos de bloques

Noticia: Es un tipo de contenidos configurado para mostrarlo como ejemplo en el frontal 1.

Pregunta Frecuente: Es un tipo de contenido configurado para mostrarlo como ejemplo en frontal2.

Frontal: Interficie que está en contacto directo con el usuario

Formulario Dinámico: Pantalla con campos administrables capaz de aumentar y reducir el número de campos en tiempo de ejecución.

6 Bibliografía

TÈCNIQUES DE DESENVOLUPAMENT DEL PROGRAMARI Autor: Xhafa, Fatos
Editorial: Editorial UOC, S.L.

ENGINYERIA DEL PROGRAMARI I Autor: Campderrich Falgueras, Benet
Editorial UOC, S.L.

ENGINYERIA DEL PROGRAMARI II Autor: Campderrich Falgueras, Benet
Editorial UOC, S.L.

Hibernate in Action: Practical Object/Relational Mapping Autores: Gavin King,
Christian Bauer

Struts in Action: Building Web Applications with the Leading Java Framework
Autores: Ted Husted, Cedric Dumoulin, George Franciscus, David Winterfeldt,
Craig R. McClanahan

Core JSTL: Mastering the JSP Standard Tag Library. Autores: David Geary,
David M. Geary

Thinking in Patterns
<http://www.mindview.net/Books/TIPatterns/>

Design Patterns and Core J2EE patterns
<http://www.patterndepot.com/put/8/JavaPatterns.htm>

JSP(tm) Standard Tag Library 1.1 implementation
<http://jakarta.apache.org/taglibs/doc/standard-doc/intro.html>

The Apache Jakarta Tomcat 5 Servlet/JSP Container
<http://tomcat.apache.org/tomcat-5.0-doc/index.html>

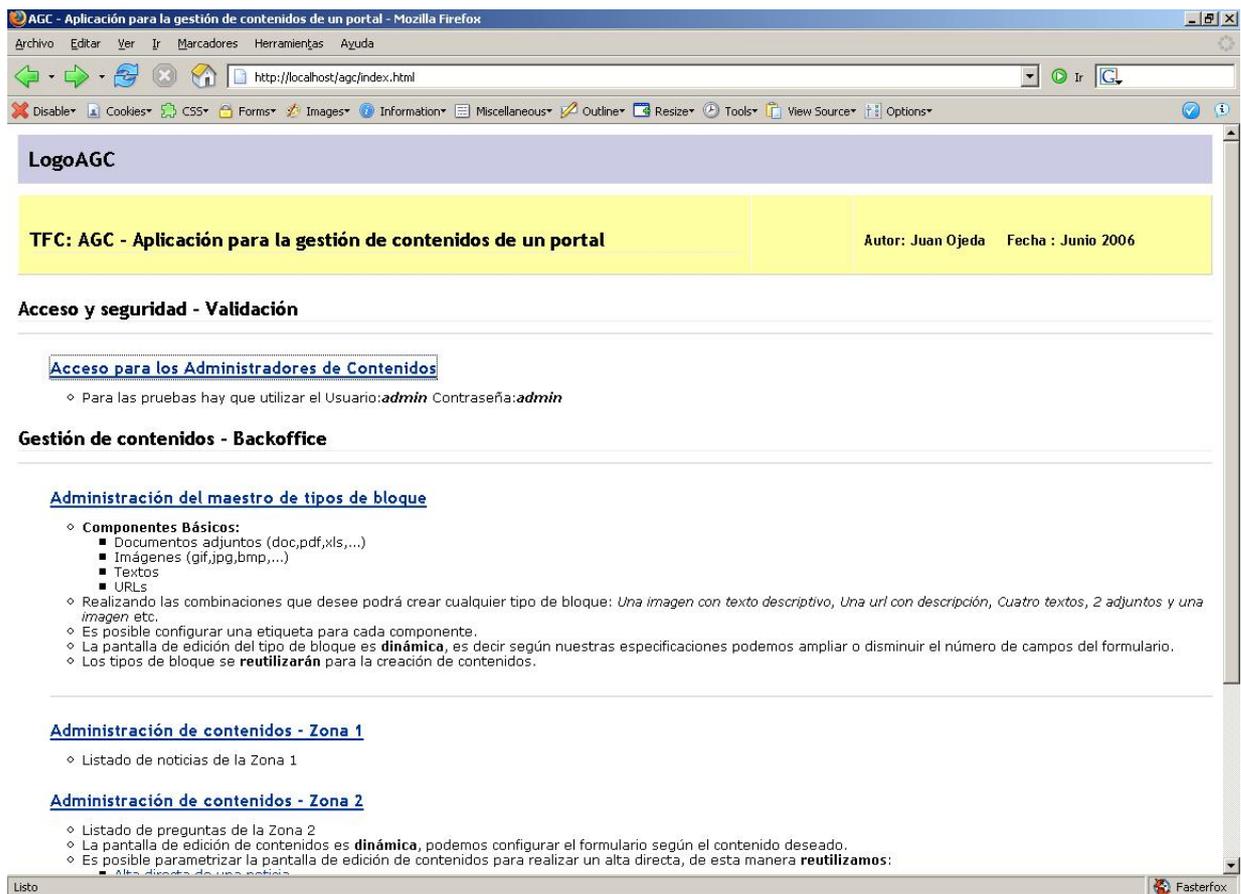
HIBERNATE - Relational Persistence for Idiomatic Java
http://www.hibernate.org/hib_docs/reference/en/html/

Apache Jakarta Struts Documentation
<http://struts.apache.org/struts-action/userGuide/index.html>

7 Anexo 1. Pantallas de la aplicación. Interficies de Usuario

Se listan a continuación las pantallas resultantes de la implementación. Se ha realizado el test funcional de la aplicación creando desde el backoffice contenidos de ejemplo Noticias y Preguntas Frecuentes. Finalmente se muestran los frontales con dichos contenidos.

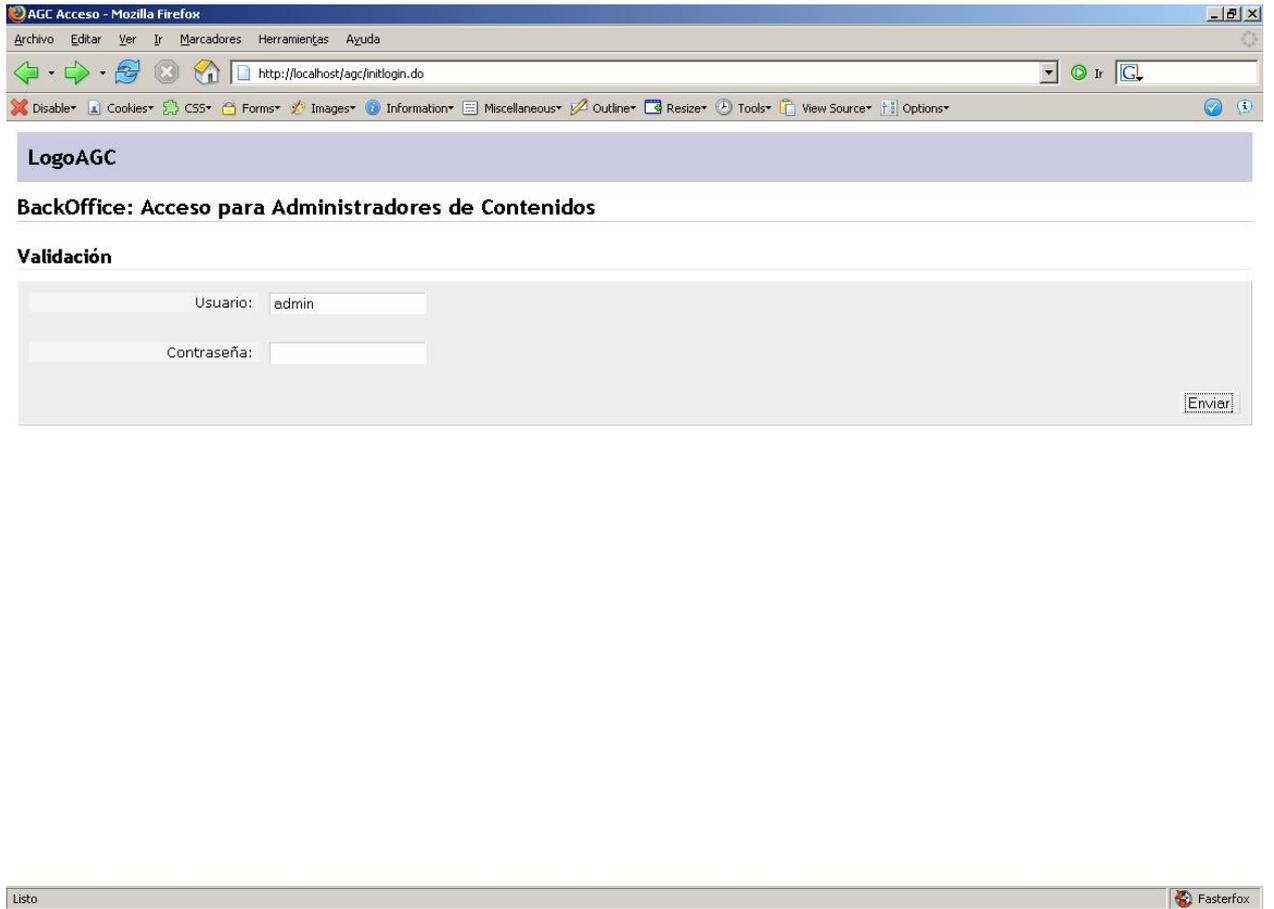
7.1.1. Pantalla Index para testeo de funcionalidades



The screenshot shows the index page of the AGC application. The browser window title is "AGC - Aplicación para la gestión de contenidos de un portal - Mozilla Firefox". The address bar shows "http://localhost/agc/index.html". The page content includes:

- LogoAGC** (purple header)
- TFC: AGC - Aplicación para la gestión de contenidos de un portal** (yellow banner)
- Autor: Juan Ojeda Fecha : Junio 2006** (yellow banner)
- Acceso y seguridad - Validación**
 - [Acceso para los Administradores de Contenidos](#)
 - Para las pruebas hay que utilizar el Usuario: *admin* Contraseña: *admin*
- Gestión de contenidos - Backoffice**
 - [Administración del maestro de tipos de bloque](#)
 - Componentes Básicos:**
 - Documentos adjuntos (doc, pdf, xls, ...)
 - Imágenes (gif, jpg, bmp, ...)
 - Textos
 - URLs
 - Realizando las combinaciones que desee podrá crear cualquier tipo de bloque: *Una imagen con texto descriptivo, Una url con descripción, Cuatro textos, 2 adjuntos y una imagen etc.*
 - Es posible configurar una etiqueta para cada componente.
 - La pantalla de edición del tipo de bloque es **dinámica**, es decir según nuestras especificaciones podemos ampliar o disminuir el número de campos del formulario.
 - Los tipos de bloque se **reutilizarán** para la creación de contenidos.
 - [Administración de contenidos - Zona 1](#)
 - Listado de noticias de la Zona 1
 - [Administración de contenidos - Zona 2](#)
 - Listado de preguntas de la Zona 2
 - La pantalla de edición de contenidos es **dinámica**, podemos configurar el formulario según el contenido deseado.
 - Es posible parametrizar la pantalla de edición de contenidos para realizar un alta directa, de esta manera **reutilizamos:**
 - Alta directa de una noticia

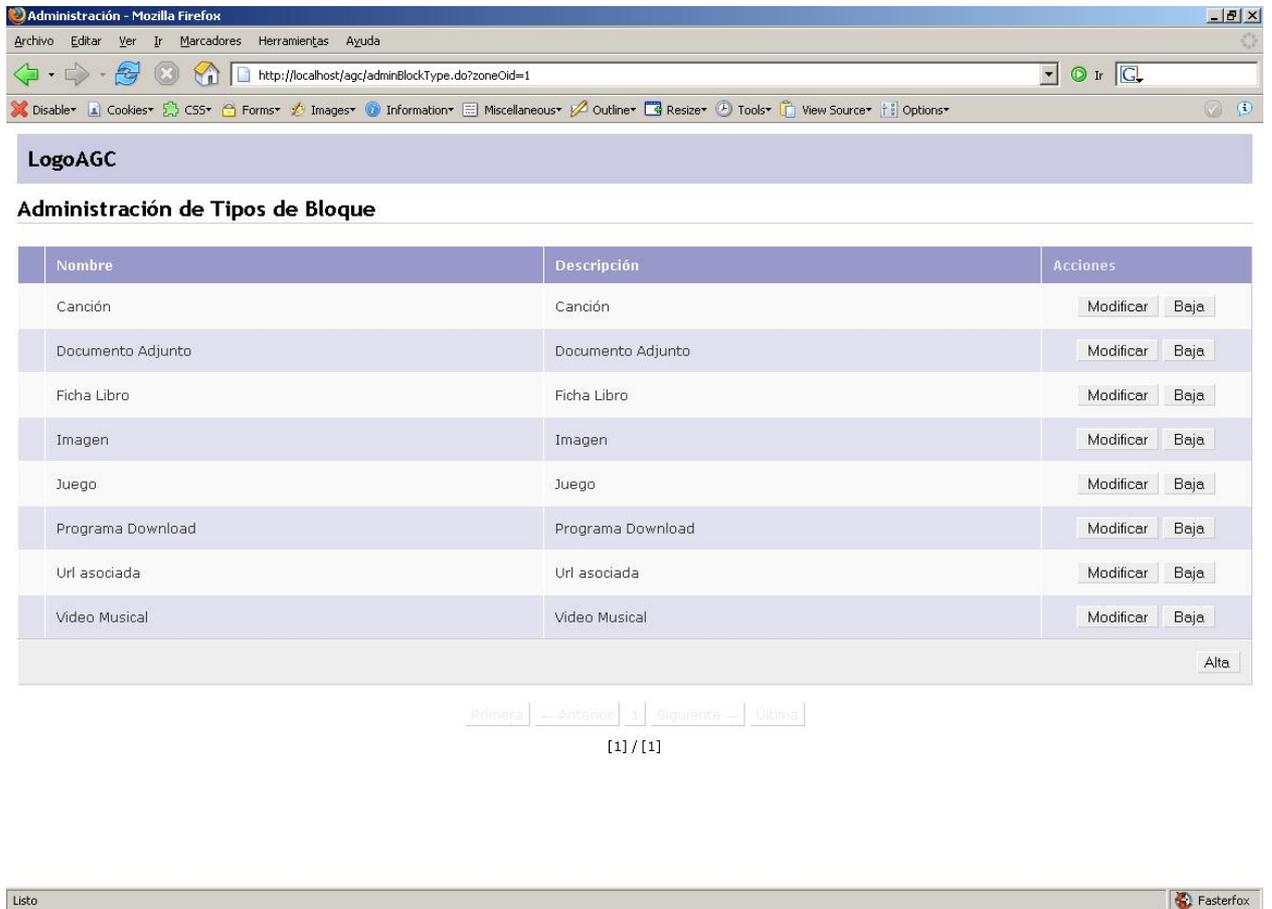
7.1.2. Pantalla de Acceso para administradores



7.1.3. Pantalla de Error de Validación



7.1.4. Pantalla Administración de Tipos de Bloque



LogoAGC

Administración de Tipos de Bloque

Nombre	Descripción	Acciones
Canción	Canción	Modificar Baja
Documento Adjunto	Documento Adjunto	Modificar Baja
Ficha Libro	Ficha Libro	Modificar Baja
Imagen	Imagen	Modificar Baja
Juego	Juego	Modificar Baja
Programa Download	Programa Download	Modificar Baja
Url asociada	Url asociada	Modificar Baja
Video Musical	Video Musical	Modificar Baja

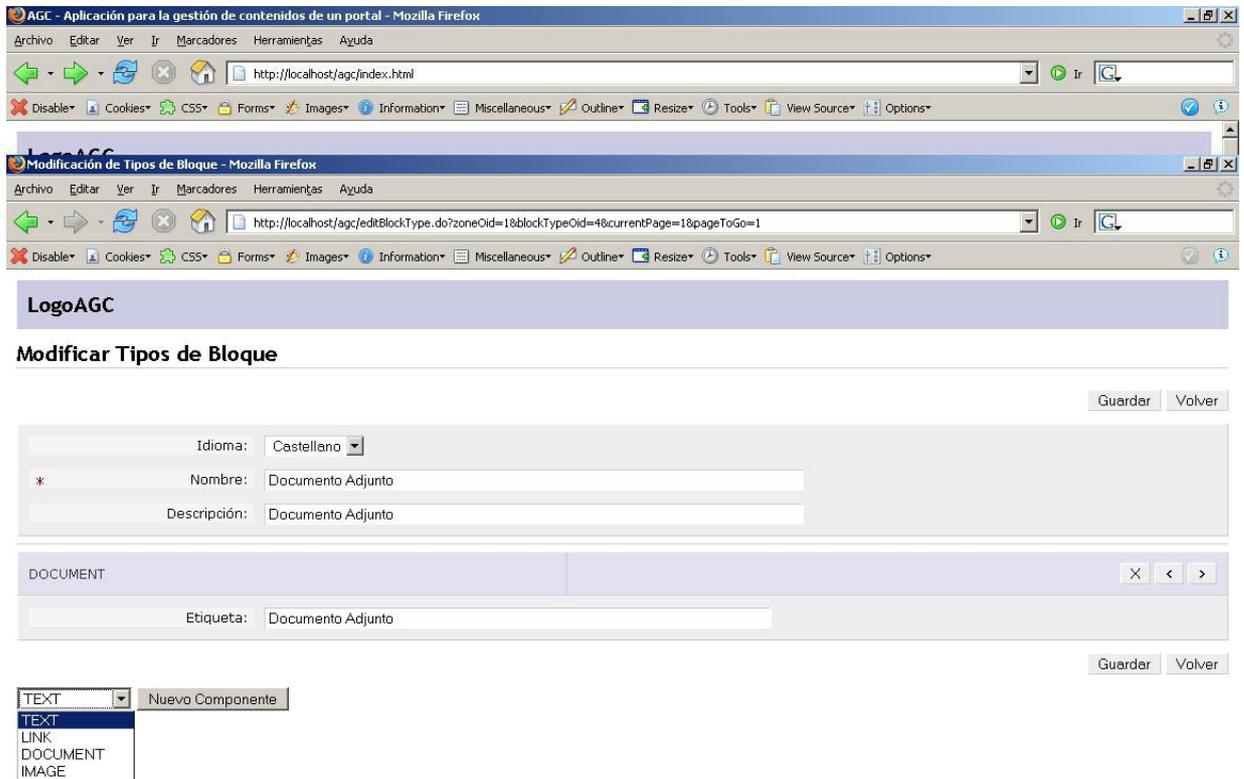
Alta

Primera Anterior Siguiente Última

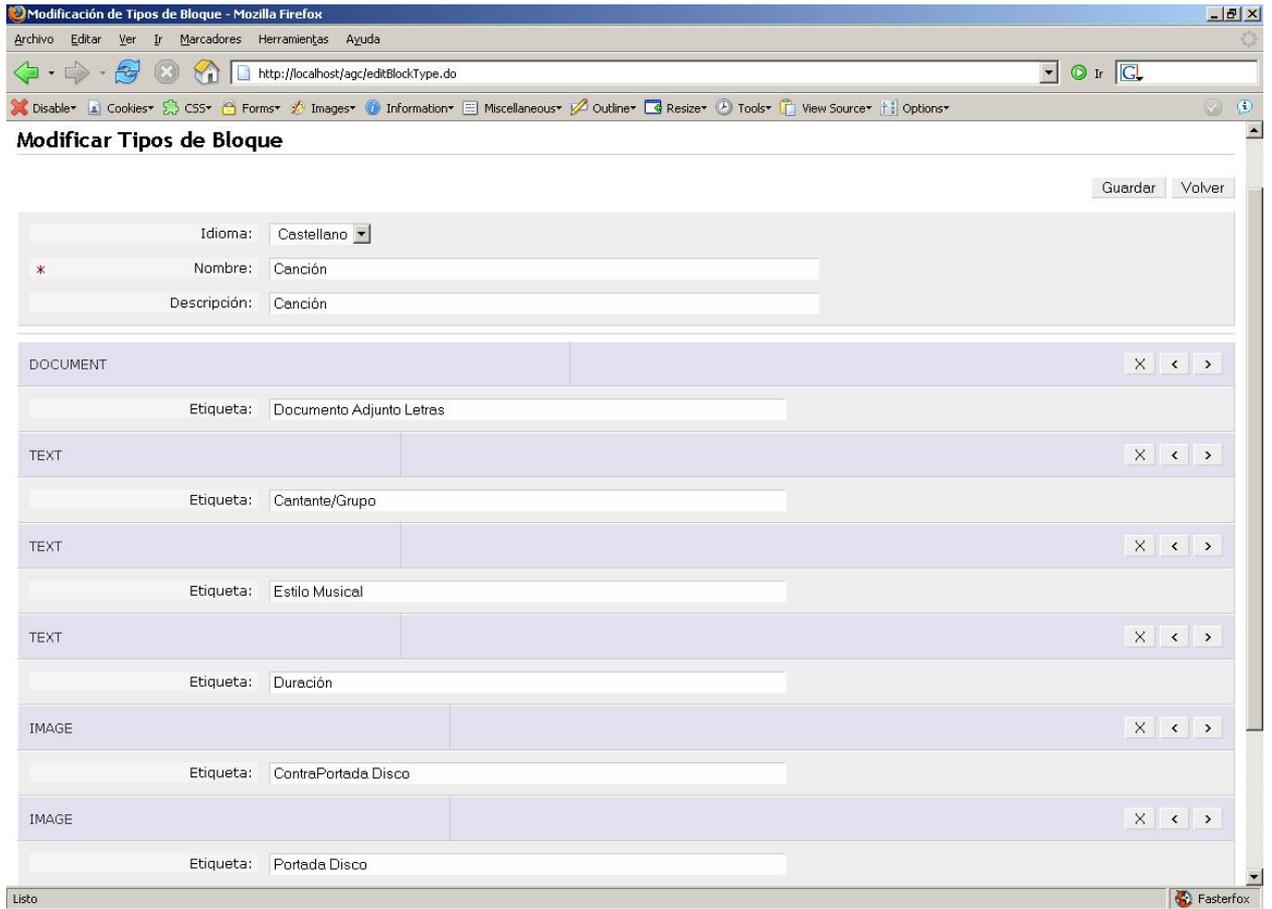
[1] / [1]

Listo Fasterfox

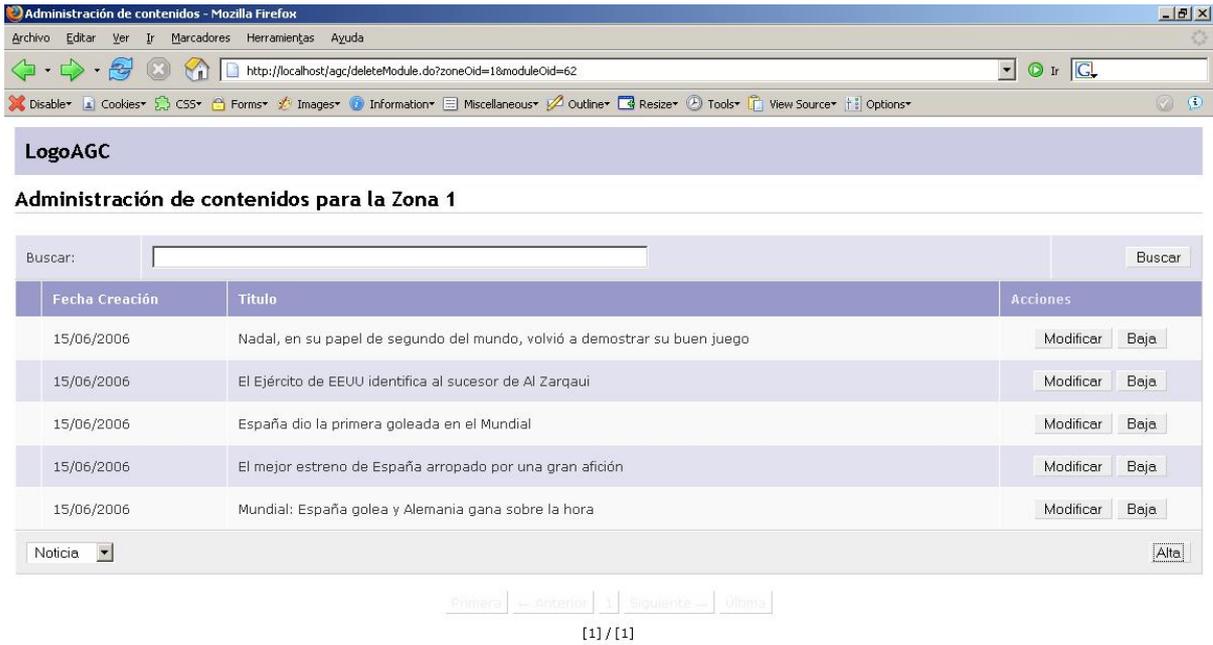
7.1.5. Pantalla de Alta/Modificar tipo de bloque: Documento Adjunto



7.1.6. Pantalla de Alta/Modificar tipo de bloque: Canción



7.1.7. Pantalla Administración de Contenidos Zona1 Noticias



LogoAGC

Administración de contenidos para la Zona 1

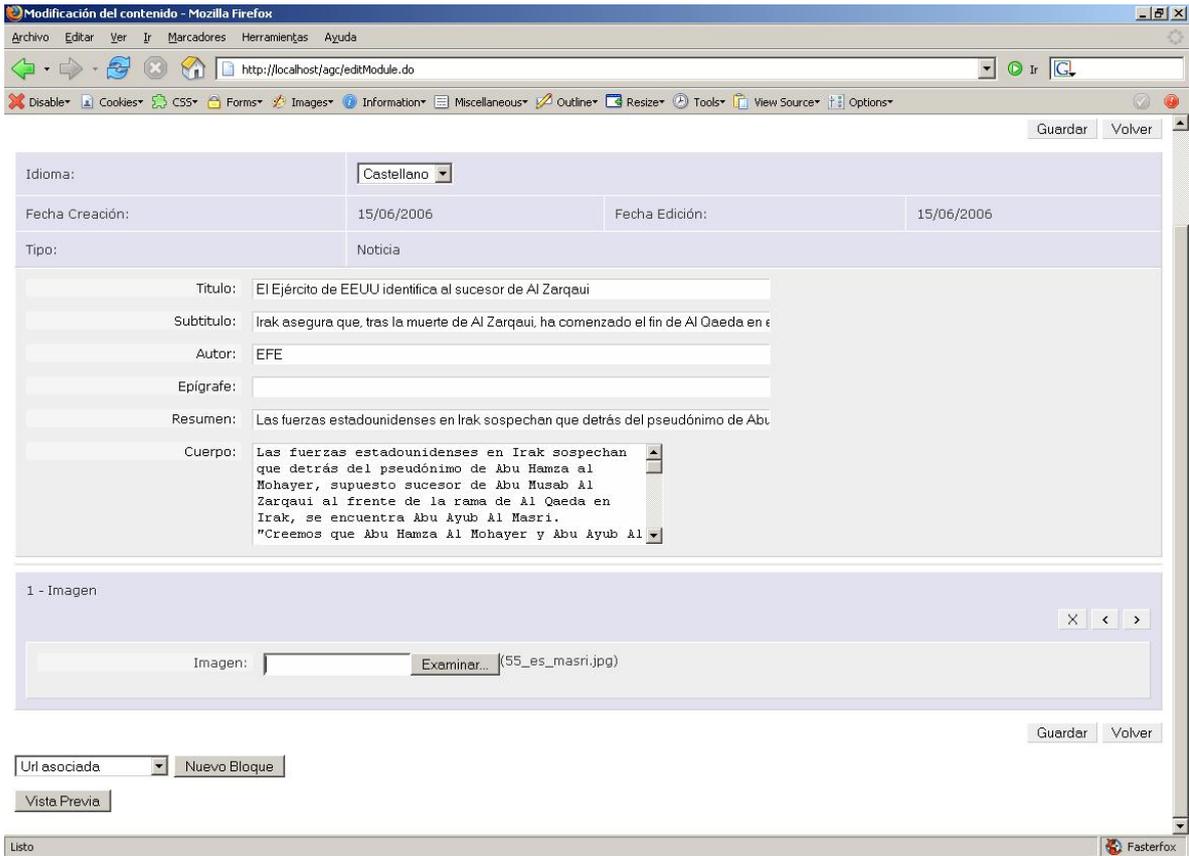
Buscar:

Fecha Creación	Título	Acciones
15/06/2006	Nadal, en su papel de segundo del mundo, volvió a demostrar su buen juego	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	El Ejército de EEUU identifica al sucesor de Al Zarqawi	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	España dio la primera goleada en el Mundial	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	El mejor estreno de España arropado por una gran afición	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	Mundial: España golea y Alemania gana sobre la hora	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>

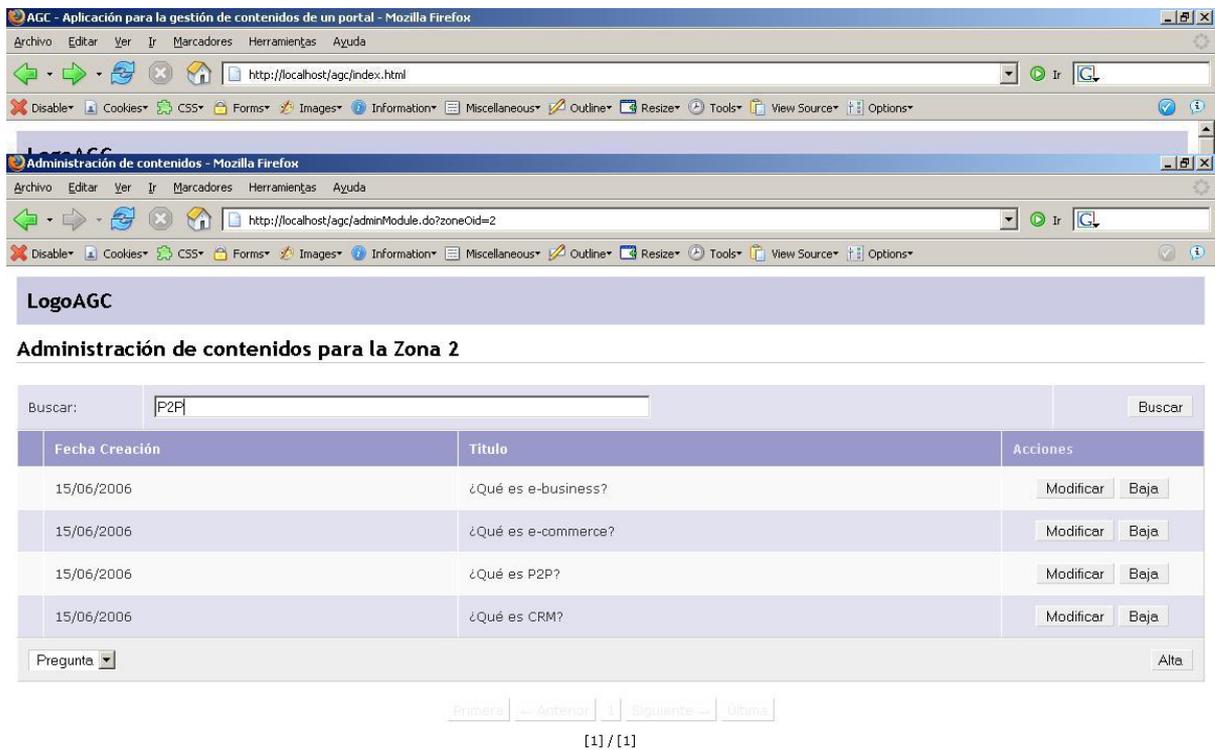
Noticia:

[1] / [1]

7.1.8. Pantalla Alta/Modificar Noticia



7.1.9. Pantalla Administración de Contenidos Zona2 Preguntas



LogoAGC

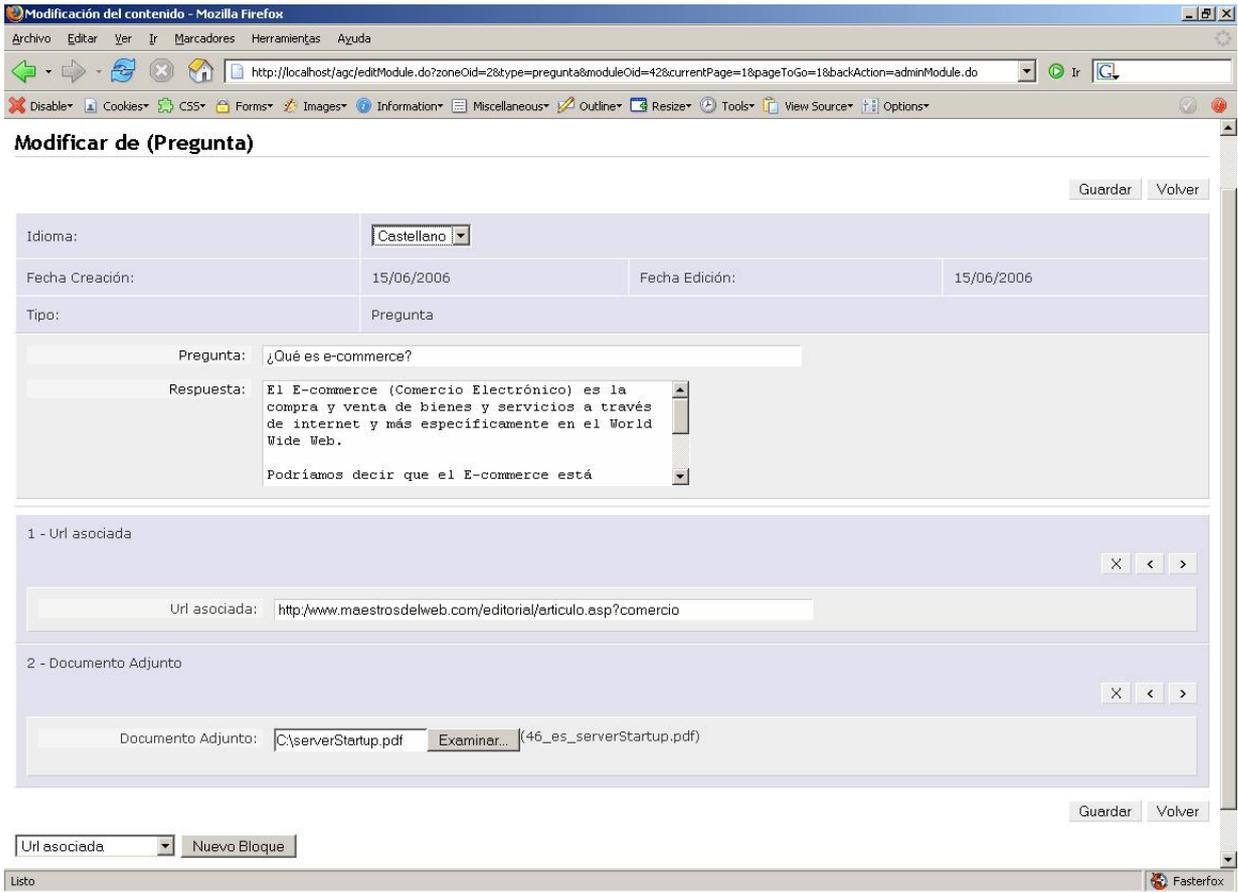
Administración de contenidos para la Zona 2

Buscar:

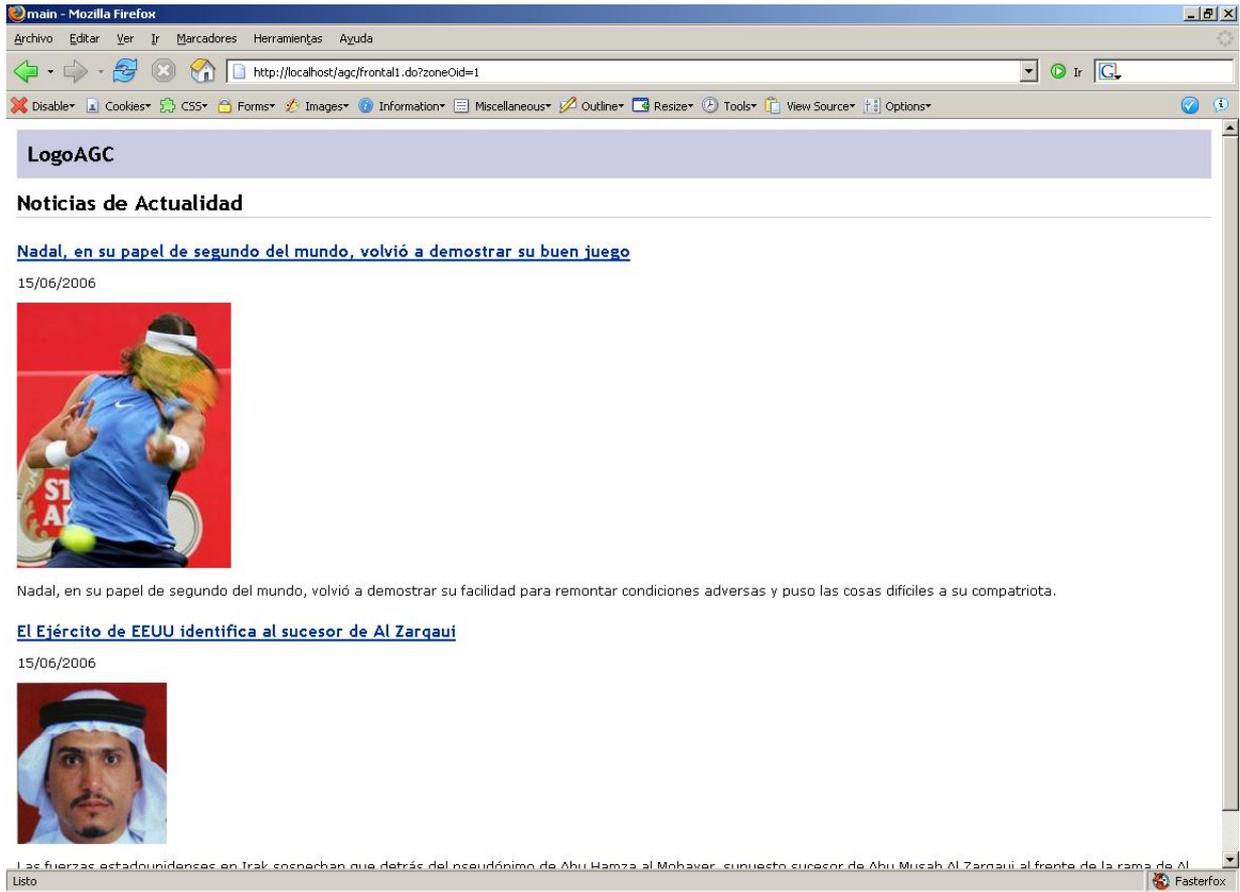
Fecha Creación	Título	Acciones
15/06/2006	¿Qué es e-business?	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	¿Qué es e-commerce?	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	¿Qué es P2P?	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>
15/06/2006	¿Qué es CRM?	<input type="button" value="Modificar"/> <input type="button" value="Baja"/>

[1] / [1]

7.1.10. Pantalla Alta/Modificar Pregunta



7.1.11. Pantalla Frontal 1 Noticias



7.1.12. Pantalla Detalle de Noticia



LogoAGC

Fecha Creación: 15/06/2006

Fecha Edición: 15/06/2006

Título: El Ejército de EEUU identifica al sucesor de Al Zaraqui

Subtítulo: Irak asegura que, tras la muerte de Al Zaraqui, ha comenzado el fin de Al Qaeda en el país

Autor: EFE

Epígrafe:

Resumen: Las fuerzas estadounidenses en Irak sospechan que detrás del pseudónimo de Abu Hamza al Mohayer, supuesto sucesor de Abu Musab Al Zaraqui al frente de la rama de Al Qaeda en Irak,

Cuerpo: Las fuerzas estadounidenses en Irak sospechan que detrás del pseudónimo de Abu Hamza al Mohayer, supuesto sucesor de Abu Musab Al Zaraqui al frente de la rama de Al Qaeda en Irak, se encuentra Abu Ayub Al Masri. "Creemos que Abu Hamza Al Mohayer y Abu Ayub Al Masri son en realidad la misma persona", dijo hoy el portavoz de las fuerzas norteamericanas en Irak, general William Caldwell, durante una rueda de prensa. Tras la muerte de Abu Muzab Al Zaraqui, el pasado miércoles en un bombardeo norteamericano en la aldea de Hibhib, en las proximidades de Baquba, la rama de Al Qaeda en Irak anunció en una página de internet que su sucesor sería Al Mohayer, un nombre hasta ese momento casi desconocido. Caldwell añadió que Al Masri, supuestamente uno de los dirigentes de Al Qaeda, conoció a Al Zaraqui en un campamento militar en Afganistán entre los años 2000 y 2001. Además, comentó que el "número uno de la célula organizativa de Al Qaeda en Irak (Al Masri)", mantiene una estrecha relación con el "segundo" de esta organización terrorista, Ayman Al Zawahiri, lugarteniente del líder de la red Al Qaeda, Osama Bin Laden.



Aceptar

Listo 

7.1.13. Pantalla Frontal 2 Preguntas

