



Universidad de Valladolid

E.U. de Informática (Segovia)

Grado en Ingeniería Informática de Servicios y Aplicaciones

---

APPS WEB

DESARROLLO DE UNA APLICACIÓN EN LA NUBE

---

Alumno: Diego Yagüe Juárez

Tutor: Fernando Díaz Gómez



**BLOQUE I**

---

**MEMORIA DEL PROYECTO**



<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>7</b>
1.1	IDENTIFICACIÓN DEL PROYECTO .....	7
1.2	ORGANIZACIÓN DE LA DOCUMENTACIÓN .....	7
	INTRODUCCIÓN .....	9
	<b>DESCRIPCIÓN GENERAL DEL PROYECTO .....</b>	<b>10</b>
1.3	OBJETIVOS .....	12
1.4	METODOLOGÍA .....	13
1.5	TECNOLOGÍAS DE DESARROLLO .....	14
	1.5.1 HTML ( <i>Hiper Text Markup Language</i> ) .....	14
	1.5.2 GOOGLE APP ENGINE .....	14
	1.5.3 DISEÑO: CSS ( <i>Cascading Style Sheets</i> ) .....	14
	1.5.4 AJAX ( <i>Asynchronous JavaScript And Xml</i> ) .....	15
<b>2</b>	<b>DESCRIPCIÓN GENERAL DEL PRODUCTO .....</b>	<b>16</b>
2.1	FUNCIONALIDADES DEL PRODUCTO .....	17
2.2	ARQUITECTURA DEL PRODUCTO .....	18
2.3	DESLIEGUE DEL PRODUCTO .....	19
<b>3</b>	<b>PLANIFICACIÓN Y PRESUPUESTO .....</b>	<b>21</b>
3.1	ESTIMACIÓN DE COSTES MEDIANTE PUNTOS DE FUNCIÓN (PFA) .....	21
	3.1.1 EXPLICACIÓN DEL PROCESO .....	21
	3.1.2 ESTIMACIÓN POR PUNTOS DE FUNCIÓN .....	23
3.2	ESTIMACIÓN COSTES MEDIANTE COCOMO .....	26
3.3	PLANIFICACIÓN .....	27
3.4	PRESUPUESTO .....	29
<b>4</b>	<b>CUESTIONES RESEÑABLES .....</b>	<b>30</b>
<b>5</b>	<b>CONCLUSIONES Y POSIBLES AMPLIACIONES .....</b>	<b>33</b>
5.1	CONCLUSIONES .....	33
5.2	CONSECUCCIÓN DE OBJETIVOS .....	33
5.3	POSIBLE AMPLIACIONES .....	33
<b>6</b>	<b>BIBLIOGRAFÍA .....</b>	<b>34</b>
<b>7</b>	<b>ANEXOS .....</b>	<b>35</b>
7.1	GLOSARIO DE TÉRMINOS .....	35



## 2.2 INTRODUCCIÓN

El presente documento contiene información que describe el problema y la solución que se adopta para el desarrollo del proyecto, en ella se guiará a través de diferentes etapas, a medida que avancemos en el libro se irán concretando mas los aspectos que han tenido que realizarse para el correcto desarrollo de este proyecto.

## 2.3 IDENTIFICACIÓN DEL PROYECTO

Título: Apps Web. Desarrollo de una aplicación en la nube.

Autor: Diego Yagüe Juárez.

Tutor: Fernando Díaz Gómez.

Departamento: Informática.

Área: Ciencias de la Computación e Inteligencia Artificial.

## 2.4 ORGANIZACIÓN DE LA DOCUMENTACIÓN

La documentación que a continuación se presenta está dividida en tres bloques, atendiendo a las recomendaciones de la E.U. de Informática de Segovia para la elaboración de la documentación de proyectos Fin de Carrera.

Estas tres secciones son independientes entre sí y están compuestas por diferentes unidades bien diferenciadas. En ellas se sigue un proceso lógico para el desarrollo de un proyecto software. Describimos a continuación cada una de las secciones correspondientes al presente documento.

La Sección I, denominada “Memoria del proyecto”, comprende un balance global de las consideraciones tenidas en cuenta para el proceso de desarrollo software, además de unas previsiones iniciales. Más concretamente, podemos encontrar:

- Descripción del proyecto.
- Descripción del producto software desarrollado.
- Memoria personal
- Planificación y presupuesto. PFA y COCOMO
- Cuestiones de diseño y de implementación que condicionaron el desarrollo y la realización de la documentación
- Conclusiones y ampliaciones
- Bibliografía

La Sección II es la correspondiente a la parte técnica del proyecto. En esta parte se realiza un análisis detallado de los recursos necesarios para desarrollar un sistema que se ajuste a los requisitos especificados por el cliente y reflejados en forma de: información relevante que debe almacenar el sistema, restricciones y funcionalidades que debe poseer el sistema. En la segunda parte de la descripción técnica se ha realizado un análisis pormenorizado de cómo se quiere desarrollar el sistema y por tanto está enfocado al diseño del sistema. Se supone que estas dos fases serán realizadas con la suficiente rigurosidad por parte de los analistas con la finalidad de que resulte más fácil a los programadores entender el alcance y funcionalidad global del sistema y así poder implementarlo lo mas eficientemente. En resumen podemos definir una serie de apartados en los que se divide esta parte:

- Análisis del Sistema

En este apartado se definen los requisitos de información que va a gestionar el sistema, se definen los objetivos que van a guiar el desarrollo del software, y se documentan los requisitos funcionales mediante casos de uso. Se definen también que tipos de usuarios van a interactuar con el sistema.

- Diseño del Sistema

Profundizamos más en el diseño del sistema y definimos nuestra estructura lógica de datos mediante los diagramas entidad-relación y el modelo relacional. Describimos el comportamiento estático del sistema definiendo los objetos del dominio, mediante los diagramas de clases y también describimos el comportamiento dinámico del sistema mediante la definición de diagramas de secuencia.

- Implementación

En este apartado se incluirán aquellos aspectos relevantes de cara a la implementación final de los componentes y unidades que van a formar la aplicación. Se detallarán también las tecnologías de las que nos hemos ayudado para realizar la implementación.

- Pruebas

Para corroborar que el funcionamiento del sistema es el correcto, se ha utilizado una herramienta para comprobar que los tipos de datos, funciones, enlaces, etc. funcionan como cabría esperar.

La sección III se corresponde con la documentación de usuario, que esta dividida en dos partes: el manual de instalación de la aplicación y el manual de uso. En ellos encontraremos documentación e ilustraciones correspondientes a la instalación de programas auxiliares, así como una serie de vistas que ilustran el funcionamiento de la aplicación.

## 2.5 INTRODUCCIÓN

Cuando un usuario inexperto en desarrollo Web desea realizar una aplicación, el primer problema con el que se encuentra, es que la información sobre cómo comenzar, esta dispersa por numerosos sitios en Internet y así gasta un preciado tiempo en realizar búsquedas hasta encontrar el resultado deseado.

Queremos unificar toda esa información y a los usuarios que hacen uso de ella a través de una herramienta Web que sea capaz de gestionar todo el contenido y conectar con redes sociales.

Con el paso del tiempo se han diversificado las herramientas de expresión a través de Internet y el ámbito de las comunidades online se ha extendido también al resto de aplicaciones de la web 2.0.

Un problema es el despliegue de la aplicación. ¿ Cumplirá el hosting con nuestras expectativas? ¿ Cuanto almacenamiento contrato? ¿ Qué ancho de banda necesito?

Estas y otras preguntas quedan resueltas conociendo algún proveedor que ofrezca sus servicios en la nube y satisfaga nuestras necesidades.

Existen aplicaciones similares, tomar como ejemplo las grandes comunidades de usuarios con editores, comunicaciones en tiempo real, integración con redes sociales, gestión de foros, etc. Lo que va a diferenciar a la aplicación va a ser el contenido que se va a ofrecer y la manera de comunicarse con los usuarios.

Con el uso de una plataforma en la nube, se ahorra tiempo en implementar cuestiones relativas a la seguridad y persistencia de los datos. También ayuda a no tener que prever las capacidades del hosting, proporcionando todas las necesidades requeridas para soportar el ciclo completo de construir y entregar aplicaciones Web y servicios desde Internet.

La nube permite que la aplicación sea accesible desde diferentes dispositivos, no solo PCs, también por tablets, smartphones o portátiles. Un tema importante que queda resuelto es el de la elasticidad: “Habilidad para decidir si hacer un uso mayor o menor de los recursos informáticos en función de las necesidades.”

La aplicación dispondrá de las operaciones básicas para gestionar las unidades de información. Estas operaciones son Alta, baja, modificación, consulta y listados.

La aplicación tendrá la capacidad de comunicarse con otras aplicaciones web en tiempo real. Los siguientes definen las características a grandes rasgos.

- Diseño de plantillas.
- Diseño de interfaces funcionales.
- Configuraciones
- Gestión de contenido: temas y artículos
- Gestión de usuarios

## DESCRIPCIÓN GENERAL DEL PROYECTO

El proyecto implementa un sistema que es capaz de gestionar una plataforma que permita la edición de contenido online por diferentes usuarios, que dispondrán de diferentes roles y tareas dentro del mantenimiento del sistema. La idea del proyecto es que estos usuarios sin tener conocimientos sobre páginas web puedan escribir, modificar o eliminar artículos que a fin de cuentas son páginas web. Estos artículos se publicarán automáticamente para facilidad de los administradores.

Esta plataforma necesita de una infraestructura donde poder almacenar la información correspondiente a artículos, usuarios, configuraciones, etc. Para ello se ha escogido la nube de Google y App Engine para gestionar la información que será almacenada en sus servidores y desplegar la aplicación. App Engine dispone de una zona de trabajo para poder editar las configuraciones y parámetros del servidor donde vamos alojar la aplicación.

El ámbito de uso de la aplicación será para usuarios que pretendan conocer más sobre el mundo del desarrollo web, los cuales podrán realizar una visita por el sitio visualizando la información disponible, agregando comentarios, llegando incluso a poder publicar sus propios artículos. Para el administrador es la forma que tiene de publicar contenido y gestionar la información de todo el sistema.

Así pues, las necesidades que han motivado al estudio del problema son dos:

- Por una parte, se tiene la necesidad de automatizar el proceso de almacenamiento y gestión de los datos.
- Por otra parte, se necesita, crear interfaces (vistas) de usuario adecuadas para una navegación agradable. También realizar un sistema de gestión de plantillas para mejorar la eficiencia a la hora de escribirlas.

El sistema tiene varias áreas diferenciadas.

- La gestión de contenidos abarca principalmente la administración de información relativa a artículos, como su contenido y vista previa, información asociada a estos como pueden ser los comentarios, estadísticas e información adicional como el autor y fecha de publicación.

También elementos comunes que comparten las páginas web como el encabezado y el pie de página.

- También necesitamos gestionar los ficheros como imágenes, hojas de estilo, scripts, etc. Estos recursos son imprescindibles para el buen funcionamiento del sistema, se necesita llevar un control de quien utiliza el recurso y donde está almacenado así por ejemplo podremos subir nuestras imágenes al servidor y utilizarlas desde alguna página de contenido.
- Gestionar información correspondiente a configuraciones del sitio como idioma, etiquetas, usuarios y estadísticas del sitio web,
- Gestionar plantillas para ahorrar tiempo a la hora de publicar una página, se necesita un sistema capaz de cargar el contenido en esas plantillas de forma rápida y eficaz.

Ahora que hemos definido nuestras unidades de información básicas, debemos pensar en los responsable encargados de administrar esas unidades de información. En este momento nos surge el problema de cómo queremos que se gestionen esos datos y quién los va a gestionar.

Se define una persona que se encargará de las tareas de administración del sistema y varios usuarios con el rol de editores. Como hay varias personas que se encargan de esta tarea, el sistema tendrá tantos editores como se deseen.

Para la gestión de los datos se ha utilizado un base de datos no relacional ya que una de las restricciones por utilizar App Engine es que debemos hacer uso de su almacén de datos y solo hay este.

- ¿Cómo diferenciaremos entre usuario y administrador?

Para ello se ha diseñado un sistema de registro e identificación que permite diferenciar los usuarios que deseen acceder al sistema haciéndolo como administradores.

- ¿Cómo realizamos las vistas de usuario y administrador?

Debido a la gran diferencia que hay entre las funcionalidades desempeñadas por un usuario y un administrador, también tendrán diferencias en sus vistas. Por una parte, un administrador puede manipular los datos que contiene el sistema, así que deberá tener las funciones adecuadas en su vista para poder interactuar fácilmente con las fuentes de información que pueda manejar el sistema. Por otro lado el usuario sólo necesita ver y consultar la información disponible en el sistema.

- Vista administrador

En la vista de administración ha sido prioritario la funcionalidad frente al diseño estético. Para realizar satisfactoriamente esta tarea se debía tener en cuenta qué funcionalidades iban a estar disponibles para el administrador. Básicamente son todas aquellas necesarias para manipular información, así se definieron unas funcionalidades básicas para cada entidad de información: alta, baja, modificación, consulta y listados.

Con esto en mente, las vistas de administrador están compuestas por árboles de directorios con operaciones disponibles, ya que dependerán del contexto en el que nos encontremos.

- Vista usuario

En las vistas de usuario hemos priorizado el diseño estético. Esto es así, porque entre otras cosas, lo que necesita el usuario del sistema es visualizar la información contenida en él, y a ser posible de una manera sencilla, cuidada y elegante. Se ha tenido en cuenta que un usuario, esté donde esté navegando en nuestro sistema, pueda acceder de forma rápida a otro recurso. Y por último que sea sencillo de utilizar.

## 2.6 OBJETIVOS

El objetivo perseguido por el proyecto es desarrollar una plataforma software donde puedan interactuar con el sistema los usuarios y administradores de modo que las fuentes de información se mantengan actualizadas por parte de los administradores y puedan beneficiarse de ello los usuarios.

Que los administradores puedan cambiar esta información, teniendo para ello un sistema en el que apoyarse sin necesidad de tener conocimiento acerca de los diferentes lenguajes empleados en el desarrollo Web, es el objetivo prioritario de este proyecto.

En consecuencia con lo anterior, definimos unos objetivos globales :

- Gestionar la información de contenido.
- Gestionar la información de plantillas.
- Gestionar la información de usuarios.
- Gestionar la información de configuración.
- Gestionar la información de recursos.

## 2.7 METODOLOGÍA

Para estructurar, planificar y controlar el desarrollo del software se va a utilizar una metodología orientada a objetos ya que nos brinda unas características adecuadas para el desarrollo del proyecto, entre ellas destacamos el polimorfismo, la herencia, la abstracción y el encapsulamiento.

Para llevarla a cabo tendremos que hacer uso del lenguaje de modelado UML que nos proporciona el mecanismo de desarrollo software bajo una arquitectura en capas, y formalizado con un lenguaje estándar y unificado. Es decir, se requiere que cada una de las partes que comprende el desarrollo del software de diseño orientado a objetos, se visualice, especifique y documente con un lenguaje común.

A la metodología escogida necesitamos darle un enfoque que guíe el desarrollo, y para ello hemos escogido el proceso incremental. Se ha escogido este enfoque porque el software a desarrollar puede ser dividido en diferentes fases (minicascadas), permitiendo así, abordar cada parte del problema por separado.

Como se muestra en la Ilustración 3 las fases consideradas han sido la iniciación, elaboración, construcción y transición. Cada una se corresponde con un estado en el que se encuentra el desarrollo del software.

Por otro lado tenemos las diferentes partes que componen el dominio del problema visto en la primera fase de modelado de negocio. Seguidamente comenzaremos con la fase de definición de requisitos, donde a medida que avancemos en nuestro proceso de desarrollo y avancemos iteraciones, iremos encontrando nuevos requisitos que serán necesarios para una correcta comprensión del sistema realizado.

La fase de implementación ha sido la más duradera. Las pruebas han sido realizadas con las herramientas de depuración de Firefox y Chrome.

El despliegue se ha realizado en una máquina local.

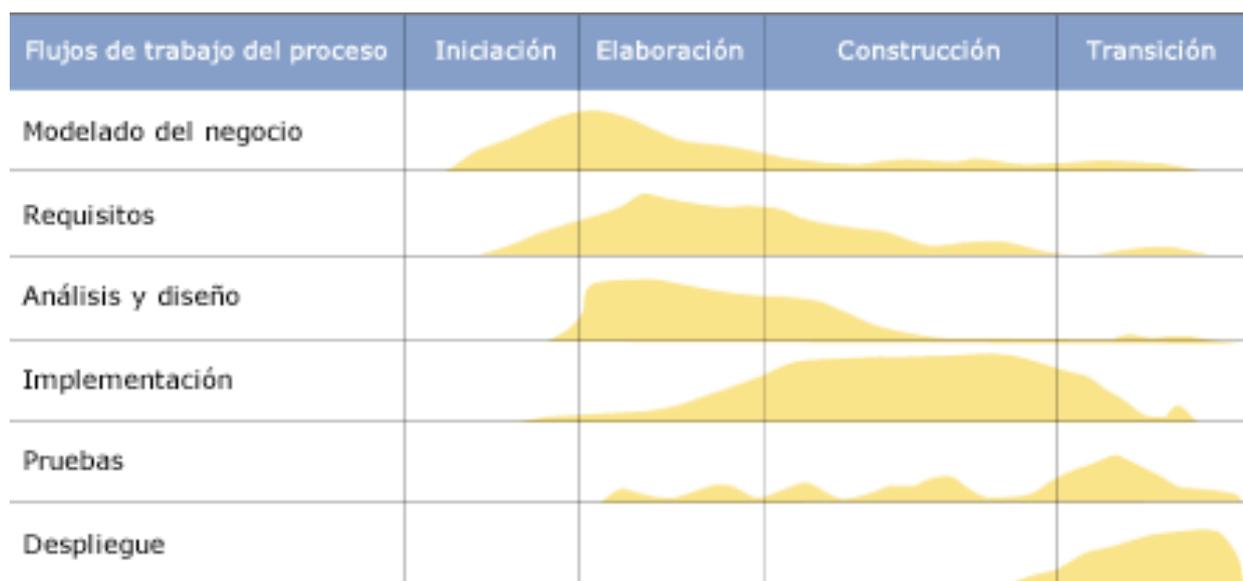


Ilustración 1: Metodología de desarrollo

## 2.8 TECNOLOGÍAS DE DESARROLLO

### 1.1.1 HTML (*Hiper Text Markup Language*)

El lenguaje de marcado de hipertexto, es el lenguaje con el que se definen las páginas Web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página Web. Es el lenguaje que entiende el navegador.

HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo *JavaScript*), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

### 1.1.2 GOOGLE APP ENGINE

App Engine es un servicio de alojamiento web que presta Google de forma gratuita hasta determinadas cuotas, este servicio permite ejecutar aplicaciones sobre la infraestructura de Google. Si no se cuenta con un dominio propio, Google proporciona uno con la siguiente estructura, midominio.appspot.com. También permite implementar un dominio propio a través de Google Apps. Por el momento las cuentas gratuitas tienen un límite de un gigabyte de almacenamiento permanente y la suficiente cantidad de ancho de banda y CPU para cinco millones de visitas mensuales, y si la aplicación supera estas cuotas, se pueden comprar cuotas adicionales.

Actualmente las aplicaciones Google App Engine se implementan mediante los lenguajes de programación Python, Java y Go.



**Ilustración 2:** Google App Engine

### 1.1.3 DISEÑO: CSS (*Cascading Style Sheets*)

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.

Es posible usar HTML, o incluso abusar del mismo, para añadir formato a los sitios Web. Sin embargo, CSS ofrece más opciones y es más preciso y sofisticado. CSS está soportado por todos los navegadores de hoy día.

Lo realmente interesante de las hojas de estilo es que se pueden vincular a ella varios documentos HTML y con un único fichero podemos controlar toda la presentación. También es una forma de ahorrar tiempo, ya que los cambios realizados en la hoja de estilo se verán reflejados en todos los documentos con los que esté asociada.

### 1.1.4 AJAX (Asynchronous JavaScript And Xml)

Ajax, acrónimo de *Asynchronous JavaScript And XML*, es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. *JavaScript* es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como *JavaScript* y *Document Object Model (DOM)*.

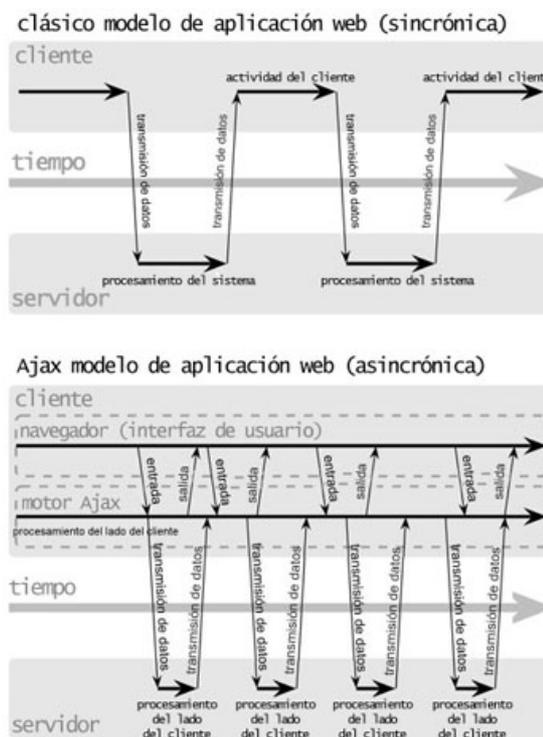


Ilustración 3: Funcionamiento de AJAX

## 2 DESCRIPCIÓN GENERAL DEL PRODUCTO

Desde un principio la idea era que si un usuario inexperto utiliza la aplicación sea capaz de hacer uso de la mayor parte de las funciones intuitivas, se ha huido de las interfaces cargadas de opciones donde supone mucho esfuerzo encontrar la acción más adecuada.

La interfaz que ven los administradores, es decir, donde los usuarios van a visualizar el contenido que el administrador ha agregado, tiene la estructura de un blog donde los artículos están clasificados por fecha y etiquetas. También hay un menú principal en la parte superior.

El pie de página está formado por información adicional sobre el contenido, comentarios que se hacen, publicaciones y algún enlace externo.

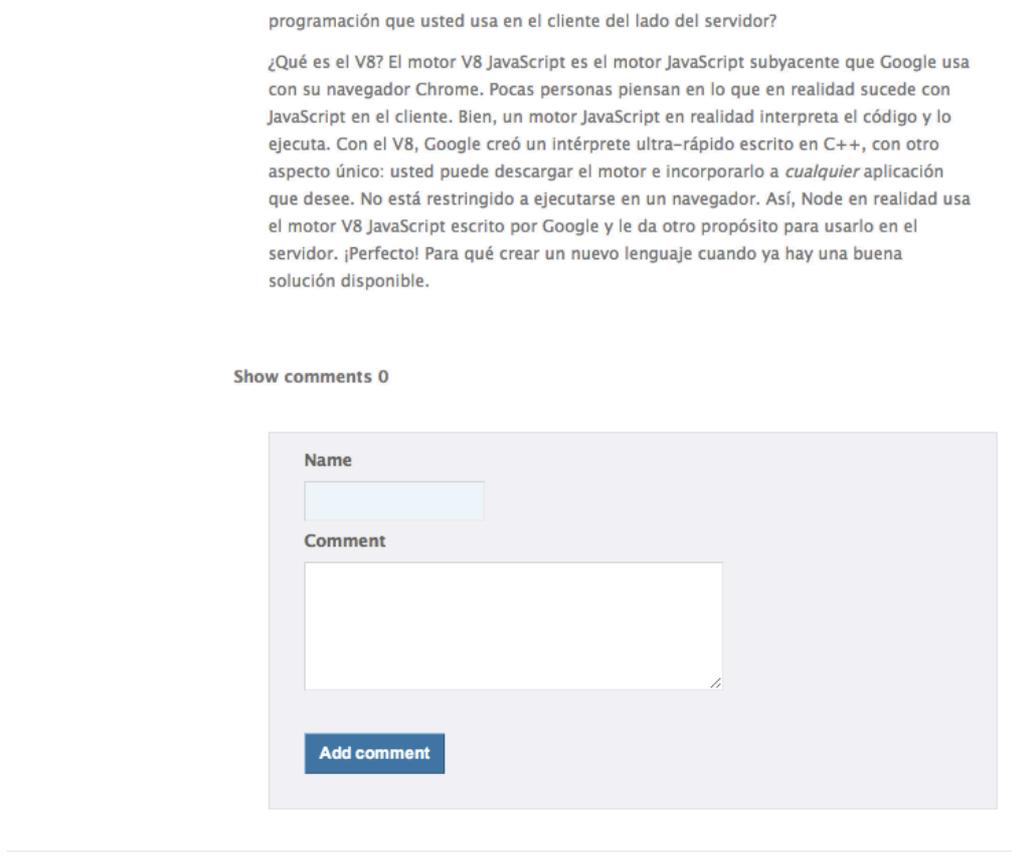
Para los administradores se han proporcionado vistas donde prime la rápida identificación y edición de recursos y contenido a través de árboles de directorio. Se persigue así un fácil y rápido acceso a las funciones del sistema.

La siguiente imagen muestra la página principal donde se van a visualizar los artículos publicados.



**Ilustración 4:** Funcionamiento de AJAX

En la imagen siguiente se muestra parte del contenido de un artículo y se muestra la zona donde los usuarios pueden comentar ese artículo.



**Ilustración 5:** Añadir comentario

## 2.9 FUNCIONALIDADES DEL PRODUCTO

Para enumerar las funcionalidades del producto vamos agrupar las entidades en base a las funcionalidades que soportan.

Gestionar plantillas: alta, baja, modificación, consulta y listado.

Gestionar permisos: registro e identificación de usuarios.

Gestionar contenido: alta, baja, modificación, consulta y listado.

Gestionar comentarios: alta, baja, modificación, consulta y listado.

Gestionar recursos: alta, baja, modificación, consulta y listado.

## 2.10 ARQUITECTURA DEL PRODUCTO

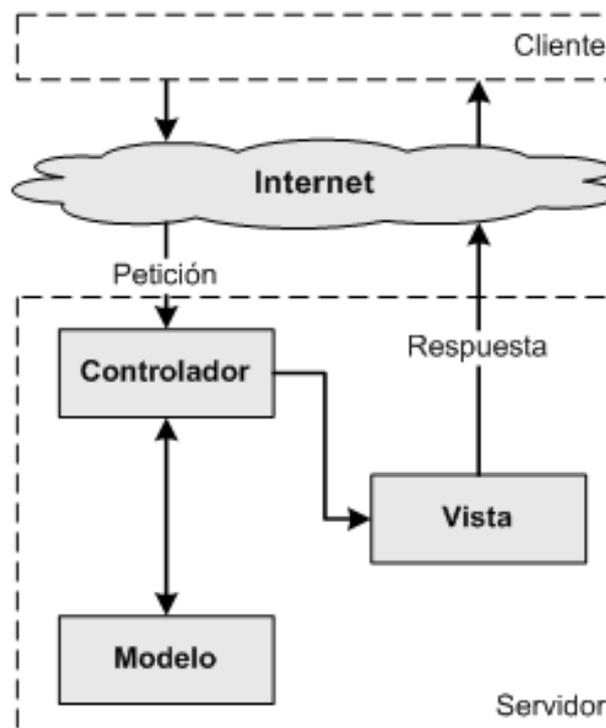
Está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.

El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

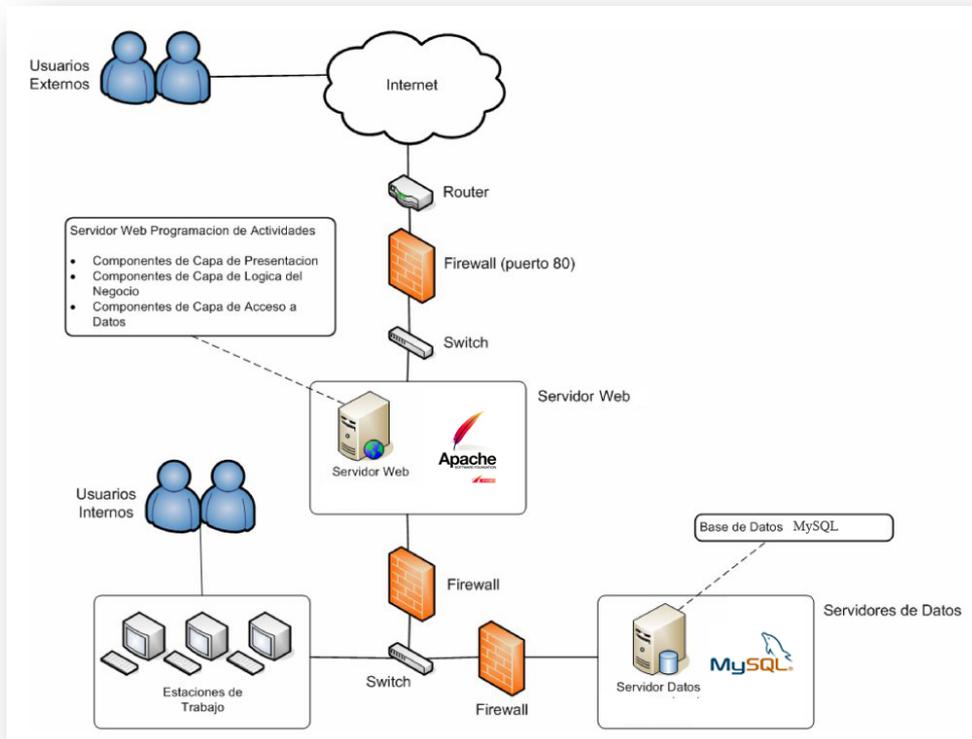
La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.



**Ilustración 6:** Arquitectura MVC

## 2.11 DESLIEGUE DEL PRODUCTO

En la siguiente imagen se muestran los distintos nodos que componen el sistema unidos por conexiones de comunicación. Es una visión dinámica del sistema, cada nodo representa un recurso de ejecución, tal como un computador, un dispositivo o memoria.



**Ilustración 7:** Despliegue de producto

Se puede observar que en la imagen existen dos servidores diferentes, uno es el servidor Web y otro el Servidor de datos. Se ha realizado así para separar todos los elementos del sistema, aunque el servidor de datos podría estar incluido dentro del servidor Web.

Los usuarios externos a la red local pueden acceder a los recursos disponiendo de una conexión a Internet, que podrá acceder a la red local a través del *router* y acceder al servidor para realizarle alguna petición.

Los usuarios internos (Red Local) pueden acceder directamente sin el uso de un *router*, accediendo de manera local mediante un navegador web que es el soporte que permite visualizar la aplicación. El navegador a través de Internet, gestiona una conexión que inicia una comunicación con el servidor mediante el protocolo HTTP.

El servidor esta compuesto por diferentes elementos que permiten al navegador mostrar la página. Dentro del servidor se encuentra una instancia de App Engine que permite generar los elementos necesarios para mostrar la página. Cuando se trata de lanzar algún *script* se ejecutara el interprete del navegador.

El servidor puede acceder a los servicios de la base de datos sin menor problema al estar situada en la misma red.

El último elemento que nos encontramos es la base de datos que almacena la información que la aplicación utiliza para generar las interfaces de usuario. Esta es modificable por medio de componentes funcionales de la aplicación.

### 3 PLANIFICACIÓN Y PRESUPUESTO

En los siguientes apartados se realiza una planificación para conocer la estimación de trabajo para el proyecto, obtener así una medida del tamaño previsto. Para ello realizaremos dos estudios de estimación basados en las características que requiere el sistema para su implementación.

#### 2.12 ESTIMACIÓN DE COSTES MEDIANTE PUNTOS DE FUNCIÓN (PFA)

La técnica de estimación de costes mediante puntos de función (PFA) proporciona una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la cantidad de funcionalidad requerida y entregada a los usuarios.

##### 3.1.1 EXPLICACIÓN DEL PROCESO

- Paso 1. Determinar el tipo de conteo.

Este paso consiste en definir el tipo de conteo entre desarrollo y mantenimiento. Esta es una forma de determinar el objetivo del conteo.

- Paso 2. Identificar los alcances de la medición y los límites de la aplicación.

El propósito de una medición consiste en dar una respuesta a un problema de negocio. El alcance de la medición define la funcionalidad que va a ser incluida en una medición específica y puede abarcar más de una aplicación

- Paso 3. Contar las funciones de datos.

Este paso consiste en identificar y contar la capacidad de almacenamiento de los datos. Se distinguen dos tipos de funciones de datos:

Archivo Lógico Interno – es un grupo de datos relacionados que el usuario identifica, cuyo propósito principal es almacenar datos mantenidos a través de alguna transacción que se está considerando en el conteo.

Archivo de Interfaz Externo - es un grupo de datos relacionados y referenciados pero no mantenido por alguna transacción dentro del conteo.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando principalmente el número de datos.

- Paso 4. Contar las funciones transaccionales.

Este paso consiste en identificar y contar la capacidad de realizar operaciones. Se distinguen tres tipos de funciones transaccionales:

Entrada Externa – es un proceso cuyo propósito principal es mantener uno más archivos lógicos internos.

Salida Externa – es un proceso cuyo propósito principal es presentar información al usuario mediante un proceso lógico diferente al de sólo recuperar los datos.

Consulta Externa – es un proceso cuyo propósito principal es presentar información al usuario leída de uno o más grupos de datos.

A cada componente identificado se le asigna una complejidad (bajo, medio o alto) considerando el número de datos utilizado en el proceso y los archivos referenciados.

- Paso 5. Determinar los puntos de función no ajustados.

Este paso consiste en sumar el número de componentes de cada tipo conforme a la complejidad asignada y sumar sus resultados para obtener el total.

- Paso 6. Determinar el valor del factor de ajuste.

El factor de ajuste se obtiene sumando 0.65 a la sumatoria de los grados de influencia de las 14 características generales del sistema, multiplicado por 0.01. Dentro de las características hay criterios como: complejidad del proceso, facilidad de instalación, entrada de datos en línea, etc.

- Paso 7. Determinar los puntos función ajustados.

Para determinar los puntos función ajustados se consideran los puntos función no ajustados por el factor de ajuste.

Estas son las 14 características generales de los sistemas que ayudan a obtener el factor de complejidad.

Comunicación de datos	Procesamiento distribuido
Rendimiento	Gran carga de trabajo
Volumen de transacciones	Entrada on-line de datos
Interface con el usuario	Actualización online
Procesamiento complejo	Utilización con otros sistemas
Facilidad de mantenimiento	Facilidad de operación
Instalación en múltiples lugares	Facilidad de cambios

**Tabla 1:** Tabla de factores de complejidad

### 3.1.2 ESTIMACIÓN POR PUNTOS DE FUNCIÓN

Hay que empezar obteniendo los parámetros que nos permitirán evaluar la funcionalidad del desarrollo software.

Parámetro significativo	Complejidad baja	Complejidad media	Complejidad alta
Entradas	x3	x4	x6
Salidas	x4	x5	x7
Ficheros internos	x7	x10	x15
Ficheros externos	x5	x7	x10
Consultas externas	x3	x4	x6

**Tabla 2:** Multiplicadores de complejidad

**Entradas** = 13 x complejidad baja.

**Salidas** = 2 x complejidad baja + 13 x complejidad media.

**Fichero internos** = 2 x complejidad baja + 1 x complejidad alta.

**Fichero externos** = 2 x complejidad media.

**Consultas externas** = 18 x complejidad media.

Para obtener los PFNA se realiza la suma de los productos de los parámetros de cada tipo.

$$PFNA = (13 \times 3) + (2 \times 4) + (13 \times 5) + (2 \times 7) + (1 \times 15) + (2 \times 7) + (18 \times 4) = 227$$

Una vez que hemos obtenido los PFNA, tenemos que calcular el factor de ajuste valorando los factores de complejidad.

Comunicación de datos	4	Procesamiento distribuido	0
Rendimiento	2	Gran carga de trabajo	2
Volumen de transacciones	2	Entrada <i>on-line</i> de datos	3
Interface con el usuario	3	Actualización online	3
Procesamiento complejo	1	Utilización con otros sistemas	0
Facilidad de mantenimiento	2	Facilidad de operación	1
Instalación en múltiples lugares	0	Facilidad de cambios	2

**Tabla 3:** Factores de complejidad

En la tabla anterior podemos deducir el siguiente sumatorio.

$$\sum FC = 25$$

Sacamos el factor de ajuste.

$$FA = (0.01 \times 25 + 0.65) = 0,9$$

Los puntos de función se obtienen con la siguiente ecuación.

$$PF = PFNA \times FA = 227 \times 0,9 = 204,3$$

Ahora calcularemos las líneas de código que se necesitan por cada punto de función. Para ellos sacamos la equivalencia de la tabla que realizó Casper Jones (Tabla 4). Como el lenguaje utilizado es PHP que no está en la tabla, pero sabemos que PHP se basa en librerías C, escogeremos este lenguaje como lenguaje.

Lenguaje	LDC/PF
Ensamblador	320
C	150
Cobol	106
Pascal	91
Basic	64
TCL	64
Java	53
C++	29

**Tabla 4:** Tabla de Casper Jones

1PF = 29 líneas de código C++.

LDC = 204,3 x 29 = 5924,7 líneas de código C++.

Tamaño PF	Rango de Esfuerzo en Horas	
	Mínimo	Máximo
50	152,66	231,88
100	299,86	455,49
137	407,46	618,93
150	445,07	676,06
200	589,01	894,70
300	874,24	1327,98
400	1156,97	1757,44
500	1437,85	2184,09
600	1717,26	2608,52
700	1995,46	3031,10
800	2272,62	3452,11
900	2548,88	3871,74
1000	2824,34	4290,17

**Tabla 5:** Relación entre puntos de función y esfuerzo en horas

Realizando la equivalencia obtenemos una relación de horas con respecto a puntos de función de 1200 horas.

### 2.13 ESTIMACIÓN COSTES MEDIANTE COCOMO

Una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos.

Este punto describe un modelo de estimación, propuesto por Barry Boehm, llamado COCOMO. Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto.

En la siguiente tabla muestra los tipos de desarrollos y los valores para las fórmulas de esfuerzo y tiempo.

Modo de desarrollo	A	B	C
Orgánico	3,20	1,05	0,38
Empotrado	3,00	1,12	0,35
Semi-Libre	2,80	1,20	0,32

**Tabla 6:** Modos de desarrollo de COCOMO

Fórmulas:

- Esfuerzo nominal (personas-mes) =  $A \times (KLDC)^B$
- Esfuerzo (personas-mes) = Esfuerzo Nominal  $\times \prod$  Factores de coste
- Tiempo de desarrollo(meses) =  $2,5 \times \text{Esfuerzo}^C$
- N° medio de personas(Personas) = Esfuerzo/Tiempo de desarrollo

En la siguiente imagen se muestra los factores de coste para el cálculo del esfuerzo.

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

**Tabla 7:** Factores de ajuste para el modelo COCOMO

Para el proyecto que se está realizando se va a escoger el modelo orgánico puesto que vamos a desarrollar para un entorno estable y se prevén unas decenas de miles de líneas de código. Por debajo de 50KLDC (Kilo Líneas De Código).

$$\text{Esfuerzo nominal} = 19,25$$

Esfuerzo (personas-mes) =  $19,25 \times 0,85$  (Complejidad del software)  $\times 0,86$  (Calidad de los analistas)  $\times 0,86$  (Calidad de los programadores)  $\times 0,82$  (Técnicas modernas de programación)  $\times 0,83$  (Empleo de herramientas) = 8,23 personas-mes.

$$\text{Tiempo de desarrollo} = 2.5 \times 8,23^{0,38} = 5,5 \text{ meses}$$

## 2.14 PLANIFICACIÓN

La planificación temporal para proyectos de desarrollo de software esta enfocada a que la fecha final de lanzamiento del sistema ya ha sido (irrevocablemente) establecida. La organización del proyecto software se ve forzada a distribuir el esfuerzo dentro de los plazos establecidos.

La planificación del proyecto se distribuye en tareas, cada tarea tiene una duración determinada y el inicio y el fin pueden estar condicionado por el resto de tareas. También observamos los predecesores en cada tarea, que significa que no se puede empezar una tarea hasta que se haya realizado la anterior. Si trabajaran mas personas en el proyecto podríamos iniciar tareas en paralelo.

		Nombre	Duración	Inicio	Terminado	Predecesores
1		Modelado de negocio	12 days	3/04/12 8:00	18/04/12 17:00	
2		Requisitos	9 days	19/04/12 8:00	1/05/12 17:00	1
3		Análisis	12 days	2/05/12 8:00	17/05/12 17:00	2
4		Diseño	13 days	18/05/12 8:00	5/06/12 17:00	3
5		Implementación	36 days	6/06/12 8:00	25/07/12 17:00	4
6		Pruebas	9 days	26/07/12 8:00	7/08/12 17:00	5
7		Despliegue	8 days	8/08/12 8:00	17/08/12 17:00	6

**Ilustración 8:** Calendarización del proyecto

### Modelado de negocio

Introducido para tener una perspectiva general del problema al que nos enfrentamos. La primera tarea consiste en definir los límites del problema y establecer pautas para acometer su estudio. Las siguientes tareas nos introducen en los conceptos básicos de las herramientas y tecnologías que vamos a utilizar para el desarrollo del proyecto.

### Requisitos:

En esta fase se ha realizado un estudio del problema con los requisitos generales que se han elicitado.

### Análisis:

Realizar un análisis del producto que se va a realizar, objetivos perseguidos, definir como se van a cumplir los requisitos y ver las funcionalidades básicas del sistema.

### Diseño

Diseñar la forma de actuar del sistema, como se va a comportar cuando interactúe con los distintos componentes que lo forman.

### Implementación

Con los documentos de análisis y diseño se empezará la implementación del sistema.

### Pruebas

Realizar pruebas de funcionamiento y visualización en distintos navegadores.

### Despliegue

Desplegar la aplicación en la nube de Google, definir tareas rutinarias de mantenimiento (cron), hacer pruebas de funcionamiento y de carga.

La duración del proyecto es de 99 días, empezando el 1 de Abril y terminando el 25 Agosto.

La siguiente imagen muestra un diagrama de GANTT donde se observan las dependencias entre las tareas y el tiempo de realización de cada una.

## 2.15 PRESUPUESTO

En el siguiente apartado se muestran los diferentes recursos, ya sean materiales o de trabajo, que el proyecto necesita para su realización. Vamos a exponer en términos monetarios el coste del proyecto.

### Recurso Materiales:

<i>Nombre</i>	<i>Uso(%)</i>	<i>Coste(€)</i>
Ordenador personal	80	900
Impresora	25	70
Internet ADSL	25	125
Xampp Server	100	0
Framework	100	0
StartUML 5.0	100	0
Open Office	100	0
Open Project 1.4	100	0
Adobe Suite (Demo)	100	0
MySQL Workbench	100	0
Material de oficina	100	50

**Tabla 8:** Recursos materiales

### Recursos de trabajo:

<i>Nombre</i>	<i>Precio por hora</i>
Jefe de Proyecto	20
Analista	15
Diseñador	15
Desarrollador	15
Documentalista	10
Ingeniero Técnico	15

**Tabla 9:** Recursos de trabajo

Los recursos de trabajo son las personas especializadas para cada parte del programa, pero el proyecto lo he realizado yo solo y he realizado todas las funciones bajo el nombre de ingeniero técnico.

## 4 CUESTIONES RESEÑABLES

- ¿Ayudas en el desarrollo?

Teniendo en cuenta el tiempo de desarrollo me decidí por buscar herramientas que me permitan desarrollar la aplicación de una manera mas rápida y eficiente. Encontré que para aplicaciones Web existen *frameworks* que permiten implementar el patrón Modelo-Vista-Controlador además de proveer de funcionalidades implementadas por el núcleo que ayudan al desarrollo de la aplicación.

- Depurador

Quiero destacar otro aspecto que me ha ayudado mucho en el desarrollo del sistema. Se trata del depurador. En un principio desconocía su existencia y al intentar localizar errores, la tarea se volvía muy complicada debido a que el lenguaje JavaScript no es compilado sino interpretado y puedes crear y lanzar la aplicación con errores. El navegador sólo te avisa de que existe un error en un archivo. Cuando descubrí esta herramienta, la tarea de localizar los errores en los scripts se torno más sencilla porque este depurador no sólo indica dónde está el error sino que también se pueden depurar todas las variables que maneja el sistema en ese momento.

- Interfaces

Cuando comienzas a diseñar una página Web utilizando HTML te das cuenta que sólo con eso no se va a poder realizar un diseño que impacte, que llame la atención y que produzca reacciones positivas en el usuario. Por lo que empiezo a documentarme en la manera de presentar la aplicación al usuario de una forma más elegante. La búsqueda me lleva a utilizar estilos en cascada, más conocidos como CSS. Lo que hacen los estilos en cascada es separar el diseño de la presentación del código HTML. Los estilos en cascada son una manera de organizar los elementos de la interfaz y de darle un diseño a ésta. Son usados de manera estática, definiendo las propiedades de los elementos que componen las interfaces.

Para cambiar el diseño dinámicamente, es decir, en respuesta a un evento se hace uso de la tecnología JavaScript que permite realizar modificaciones del diseño cuando algún evento es registrado. Con ello podemos interactuar con el usuario y que él se dé cuenta de que la página responde a las acciones que el realiza.

- AJAX

Cuando pensé en usar esta tecnología fue porque se me presentó el problema de estar en una página (interfaz) y tener que re-cargar la página entera cuando el cambio que se producía, solo afectaba a una pequeña parte de la página. Es evidente que re-cargar páginas no es agradable cuando se está navegando y menos re-cargar cada vez que se necesita algún recurso en un espacio corto de tiempo. Por esto, decidí empezar a documentarme sobre cómo funciona esta tecnología y me sorprende gratamente lo que es capaz de hacer. Mediante un objeto manejado por el navegador, pueden realizarse peticiones al servidor en segundo plano, con lo que se pueden realizar otras tareas, sin dejar al usuario con una página en blanco.

- Estándares

Un problema de magnitud mayor ha sido el tener que utilizar diferentes estándares a la hora de diseñar las interfaces mediante CSS. Debido a que no existe un estándar común, cada navegador posee sus propias instrucciones por lo que ha tenido que triplicarse el código y con ello, el aumento del tiempo invertido en la definición de estilos. Hay que destacar que para los navegadores Chrome, Mozilla y Opera as instrucciones son muy parecidas. El problema viene cuando se utiliza el navegador Explorer,

ya que no posee los mismos estándares y las instrucciones son muy diferentes. Por esta razón no está optimizado para Internet Explorer. Aparte de que las instrucciones que haya que incluir para optimizar en Internet Explorer, no puede interpretar tantas instrucciones como los demás navegadores, por lo que carece de funcionalidades de las que sería deseable disponer.



## 5 CONCLUSIONES Y POSIBLES AMPLIACIONES

### 2.16 CONCLUSIONES

Este proyecto ha supuesto una prueba para mí, tenía poco tiempo disponible para realizar un proyecto que preveía que me iba a dar trabajo. He tenido que ir avanzando sin entretenerme con cuestiones sin importancia y guiar el proceso de desarrollo en una línea que no podía desviarse de los objetivos lo más mínimo. El tener una idea clara de lo que quería hacer me ha servido mucho para saber cual era el producto final que quería llegar y en cierto modo lo he conseguido aunque como veremos dos apartados más abajo, todavía puedo alargar este proyecto durante unos meses.

Con este proyecto he aprendido a trabajar con los datos en la nube, ha interesarme por diferentes proveedores, por como configurar servidores de aplicaciones para un uso más eficiente. También he mejorado mi grado de conocimiento de Java y otras herramientas para el desarrollo de aplicaciones que me hace ver otras aplicaciones desde otro punto de vista.

Seguir avanzando en lenguajes que ya conocía como HTML, que en este año 2012, se ha producido una gran actualización a la versión 5 y he podido ver y utilizar estas nuevas especificaciones con una gran capacidad de mejora.

### 2.17 CONSECUCIÓN DE OBJETIVOS

Los objetivos que se presentaron han sido cubiertos con garantías, desarrollando un entorno para poder manejar la información perteneciente a un entorno para compartir contenido y poder mostrarla a los usuarios interesado por medio de Internet. Accesible por todas las personas que dispongan de una conexión a Internet.

Con la herramienta desarrollada los administradores son capaces de modificar la información que compone el dominio del problema y así mantener actualizada la página, sin necesidad de tener una persona especializada en programación Web.

Se mantiene los principios de sencillez, de manejo y claridad en la interfaz de usuario. Una herramienta sencilla permite un mejor uso de la aplicación por parte de los usuarios inexpertos con herramientas informáticas.

### 2.18 POSIBLE AMPLIACIONES

Utilizar el API de Facebook y Twitter para conectarle con sendos perfiles en redes sociales, es decir, que al actualizar información en cualquiera de los perfiles, ésta se viera reflejada en el sistema.

Crear una cuenta en con Google Analytics y enlazar con nuestro sitio para poder tener estadísticas sobre el uso de nuestra aplicación por parte de los usuarios, datos interesantes como duración de la sesión, lugar de conexión, número de clicks, etc.

## 6 BIBLIOGRAFÍA

- Rumbaugh, J., Jacobson, I., Booch, G (1999). *El lenguaje unificado de modelado. Manual de referencia*. Madrid: Pearsons Education, S.A., 2000.
- Introducción a CSS . Javier Eguíluz Pérez. Dirigido a programadores Web sin experiencia en CSS. Disponible en: <http://librosweb.es>
- Introducción a JavaScript . Javier Eguíluz Pérez. Dirigido a diseñadores Web. Disponible en: <http://librosweb.es>.
- Introducción a AJAX . Javier Eguíluz Pérez. Dirigido a programadores Web. Disponible en: <http://librosweb.es>
- Refsnes Data. XML DOM tutorial. Disponible en : <http://www.w3school.com/DOM/default.asp>
- Introducción a jQuery. Disponible en <http://librojquery.com>
- Consorcio del World Wide Web. Disponible en: [www.w3c.org](http://www.w3c.org)

## 7 ANEXOS

### 2.19 GLOSARIO DE TÉRMINOS

#### **AJAX**

Acrónimo de *Asynchronous JavaScript And XML*, es una técnica de desarrollo Web para crear aplicaciones interactivas

#### **BBDD**

Abreviatura de Base de Datos. Se define una base de datos como una colección estructurada de datos entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requisitos de información.

#### **COCOMO**

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés *CONstructive COst MObel*) es un modelo matemático de base empírica utilizado para estimación de costes de software.

#### **CSS**

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

#### **DHTML**

El HTML Dinámico o DHTML (*Dynamic HTML*) designa el conjunto de técnicas que permiten crear sitios Web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.

#### **DOM**

El *Document Object Model* o DOM es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

#### **HTML**

Siglas de *HyperText Markup Language* («lenguaje de marcado de hipertexto»), es el lenguaje de marcado predominante para la elaboración de páginas Web. Es usado para describir la estructura y el contenido en forma de texto.

#### **IU**

Siglas de interfaz de usuario, es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

#### **MVC**

Modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

#### **MySQL**

MySQL es un sistema de gestión de bases de datos relacional.

#### **PHP**

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Se usa principalmente para la interpretación del lado del servidor.

**PFA**

Siglas para la estimación de costes de un proyecto software denominado Puntos De Función.

**RSS**

RSS son las siglas de *Really Simple Syndication*, un formato XML para syndicar o compartir contenido en la Web.

**SQL**

El lenguaje de consulta estructurado o SQL, *structured query language*, es un lenguaje declarativo de acceso a bases de datos relacionales

**UML**

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad;

**URL**

Un localizador de recursos uniforme, más comúnmente denominado URL, *uniform resource locator*, es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet.

**XML**

Siglas en inglés de *eXtensible Markup Language* , es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

**XHTML**

Acrónimo en inglés de *eXtensible Hypertext Markup Language* es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.

## **BLOQUE II**

---

# **DOCUMENTACIÓN TÉCNICA**

---

## **PARTE I - ANÁLISIS**



**ÍNDICE: ANÁLISIS**

1	ANÁLISIS DEL SISTEMA.....	43
1.1	INTRODUCCIÓN .....	45
1.2	OBJETIVOS DEL SISTEMA.....	47
1.3	CATÁLOGO DE REQUISITOS DEL SISTEMA .....	49
	<i>1.3.1 REQUISITOS DE INFORMACIÓN</i> .....	49
	<i>1.3.2 REQUISITOS FUNCIONALES</i> .....	53
1.4	MATRIZ DE RASTREABILIDAD .....	67
2	RESUMEN.....	69
3	ÍNDICE DE TABLAS .....	70
4	INDICE DE ILUSTRACIONES .....	71



# **8 ANALISIS DEL SISTEMA**



## 8.1 INTRODUCCIÓN

Acordes con el tiempo con el que dispongo para realizar el trabajo fin de grado, no podía escoger un desarrollo de software que implicara el estudio de alguna tecnología nueva, tampoco tenía tiempo para enrolarme en un desarrollo de software largo y tedioso.

Tras estas observaciones, la idea es un desarrollo ágil y procurar perder poco tiempo en tareas comunes y rutinarias con el fin de poder emplear el tiempo en otras.

Se trata de desarrollar una plataforma web donde la gente pueda compartir conocimiento, pueda interactuar con otros usuarios con el fin de proporcionar un sitio de referencia en internet en cuanto a la difusión de conocimiento.

Para realizar esta tarea he pensado en utilizar una herramienta de Google que se denomina App Engine que nos proporciona un entorno local para poder desarrollar nuestras aplicaciones, poder hacer uso de su nube para almacenar datos propias, y un modo sencillo y rápido para el despliegue de la aplicación.

Para el desarrollo del sistema se van a utilizar diferentes tecnologías. Para programar el núcleo de la plataforma se ha optado por utilizar JAVA por su fácil uso, por que no es una tecnología nueva aunque si se va a profundizar más en el uso de librerías y plugins para que el núcleo tenga la funcionalidad necesaria para resolver nuestros problemas.

Para el desarrollo de las interfaces de usuario se ha optado por utilizar tecnologías web como HTML, CSS, JavaScript por ser los lenguajes de referencia en cuanto a entornos de aplicación web.

Un problema principal es el del almacenamiento y la manipulación de datos. Para ello vamos hacer uso del gran almacén que es como denomina Google a su gigantesca base de datos. Vamos a tener que resolver el problema que supone trabajar con una base de datos no relacional, por lo que tendremos que modificar un poco nuestra manera de pensar al realizar las consultas y definir la estructura de la base de datos.

Otro problema que no quería tener era el de tener que codificar la misma cosa multitud de veces, por eso he apoyado mi desarrollo en plantillas, con ellas dotamos a nuestra aplicación de mayor eficiencia y el proceso de desarrollo se convierte en una tarea menos tediosa y más ágil. Se ha implementado un sistema de plantillas que permiten reutilizar pequeñas partes de código.

La plataforma puede dividirse en dos grandes partes: la zona de usuarios y la zona de administración.

La zona de administración es la que permite manejar las entidades de datos pudiendo gestionar toda la información de la aplicación bajo una plataforma de desarrollo. Es complicada porque hemos hecho uso de librerías y plugins que en su conjunto podríamos dividir en:

Los siguientes apartados indican los puntos más importante de la aplicación.

1. Almacenamiento
2. Contenido:
  - Paginas
  - Estructura (XML): blog article
  - Plantillas de estructura: article, article-overview
3. Plantillas
4. Recursos
  - Carpetas
  - Imágenes
  - Archivos CSS y JS
5. Configuración
  - Configuración del sitio
  - Comentarios
  - Idiomas
  - Paquetes del mensaje
  - Usuarios
  - Grupos
  - Etiquetas
  - Atributos
6. Plugins
  - Configuración de plugins
  - Formularios
  - SEO urls,
7. Información Personal/ usuario
  - Perfil
  - Idioma
  - Soporte
8. Funciones básicas
  - Guardar, exportar, importar, crear índice de búsqueda, reinicializar, cargar sitio por defecto, limpiar cache, estadísticas. Añadir, eliminar.

## 8.2 OBJETIVOS DEL SISTEMA

<b>OBJ-01</b>	<b>Gestionar plantillas</b>
<b>Versión</b>	<i>0.1 (11/05/2012&gt;)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema deberá ser capaz de llevar a cabo una gestión de plantillas, funcionalidades básicas como son el alta, baja, modificación, consulta y baja . También se deberá llevar a cabo la carga de plantillas dinámicamente.
<b>Subobjetivos</b>	
<b>Importancia</b>	<i>Importante</i>
<b>Urgencia</b>	<i>Urgente</i>
<b>Estado</b>	<i>En construcción</i>
<b>Estabilidad</b>	<i>Alta</i>

**Tabla 10: Objetivo 01 - Gestión de plantillas.**

<b>OBJ-02</b>	<b>Gestionar permisos</b>
<b>Versión</b>	<i>0.1 (11/05/2012&gt;)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema debe ser capaz de ofrecer un sistema de permisos fiable, necesitamos conseguir tener integrado un sistema de autenticación de usuarios para separar la parte accesible por cualquier usuario de la parte de administración, accesible solo por personal del centro.
<b>Subobjetivos</b>	OBJ-01.01 Gestión de Grupos OBJ-01.02 Gestión de Usuarios
<b>Importancia</b>	<i>Importante</i>
<b>Urgencia</b>	<i>Hay presión</i>
<b>Estado</b>	<i>En construcción</i>
<b>Estabilidad</b>	<i>Alta</i>

**Tabla 11: Objetivo 02 - Gestionar permisos.**

<b>OBJ-02.01</b>	<b>Gestionar grupo</b>
<b>Versión</b>	<i>0.1 (11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema deberá ser capaz de llevar un control de los grupos de usuario disponibles en nuestra aplicación, se desea poder consultar, eliminar o modificar grupos existentes así como poder añadir nuevos grupos.
<b>Importancia</b>	Importante
<b>Urgencia</b>	Hay presión
<b>Estado</b>	En construcción
<b>Estabilidad</b>	Alta

**Tabla 12: Objetivo 02.01 - Gestionar grupos.**

<b>OBJ-02.02</b>	<b>Gestionar usuarios</b>
<b>Versión</b>	<i>0.1 (11/05/2012&gt;)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>

<b>Descripción</b>	El sistema deberá ser capaz de llevar un control de los usuarios que harán uso de nuestro sistema, se desea poder consultar, eliminar o modificar usuarios existentes así como poder añadir nuevos usuarios.
<b>Urgencia</b>	<i>Hay presión</i>
<b>Estado</b>	<i>En construcción</i>
<b>Estabilidad</b>	<i>Alta</i>

**Tabla 13: Objetivo 02.02 - Gestionar usuarios**

<b>OBJ-03</b>	<b>Gestionar contenido</b>
<b>Versión</b>	<i>0.1 (11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema debe ser capaz de poder llevar un control de los contenidos que va a tener la aplicación, como pueden ser los artículos, del blog, o partes de la estructura de las páginas o los comentarios. Gestionar de manera eficiente todo lo que el usuario va a poder visualizar cuando acceda al sitio.
<b>Importancia</b>	<i>Importante</i>
<b>Urgencia</b>	<i>Hay presión</i>
<b>Estado</b>	<i>En construcción</i>

**Tabla 14: Objetivo 03 – Gestionar contenido**

<b>OBJ-04</b>	<b>Gestionar recursos</b>
<b>Versión</b>	<i>0.1 (11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema deberá ser capaz de gestionar los recursos software que se necesiten. Podrán asociarse a diferentes contenidos. Podrán ser recursos como imágenes, hojas de estilo, scripts, etc.
<b>Importancia</b>	<i>Importante</i>
<b>Urgencia</b>	<i>Hay presión</i>
<b>Estabilidad</b>	<i>Alta</i>

**Tabla 15: Objetivo 04 - Gestionar recurso**

<b>OBJ-05</b>	<b>Gestionar comentario</b>
<b>Versión</b>	<i>0.1 (11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema deberá ser capaz de gestionar los comentarios hechos sobre los artículos.
<b>Importancia</b>	<i>Importante</i>
<b>Estado</b>	<i>En construcción</i>
<b>Estabilidad</b>	<i>Alta</i>

**Tabla 16: Objetivo 05 - Gestionar comentario**

## 8.3 CATÁLOGO DE REQUISITOS DEL SISTEMA

### 8.3.1 REQUISITOS DE INFORMACIÓN

<b>IRQ-01</b>	<b>Información sobre usuario</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-02 Gestionar permisos OBJ-02.01 Gestionar usuarios
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los usuarios.> En concreto:
<b>Datos específicos _</b>	Id Nombre Username Contraseña Descripción Creado
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.
<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 17: IRQ 01 – Información sobre usuario.**

<b>IRQ-02</b>	<b>Información sobre grupo</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-02 Gestionar permisos OBJ-02.02 Gestionar grupos
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los grupos.> En concreto:
<b>Datos específicos _</b>	Nombre Descripción
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.
<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 18: IRQ 02 – Información sobre grupo.**

<b>IRQ-03</b>	<b>Información sobre plantillas</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-01 Gestionar plantillas
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a las plantillas.> En concreto:

<b>Datos específicos _</b>	Autoguardado Cuerpo Nombre Título
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.
<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 19: IRQ 03 – Información sobre plantillas.**

<b>IRQ-04</b>	<b>Información sobre contenido</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al contenido. En concreto:
<b>Datos específicos _</b>	URL amigable Idioma Título Contenido Plantilla Fecha Incluir en búsquedas Caché Tipo Meta Keywords META Description Contenido en Head
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.
<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 20: IRQ 04 – Información sobre contenido.**

<b>IRQ-05</b>	<b>Información sobre recurso</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-01 Gestión de la Fundación Virtual.
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al contenido. En concreto:
<b>Datos específicos _</b>	Nombre Fichero Subcarpeta Tipo MIME Tamaño
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.

<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 21: IRQ 05 – Información sobre recurso.**

<b>IRQ-06</b>	<b>Información sobre comentarios</b>
<b>Versión</b>	<i>0.1(11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Objetivos asociados</b>	OBJ-05 Gestionar comentario
<b>Requisitos asociados</b>	
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los comentarios.> En concreto:
<b>Datos específicos _</b>	Id Autor Email Contenido Fecha
<b>Importancia</b>	Importante.
<b>Urgencia</b>	Hay presión.
<b>Estado</b>	Pendiente de validación.
<b>Estabilidad</b>	Alta.

**Tabla 22: IRQ 06 – Información sobre comentario.**



## 8.3.2 REQUISITOS FUNCIONALES

### 8.3.2.1 DEFINICIÓN DE ACTORES

<b>ACT-&lt;01</b>	<b>Administrador</b>
<b>Versión</b>	<i>0.1(11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	Este actor representa al administrador del sistema. Este actor tendrá privilegios especiales, que le permitirán hacer uso de todas las funcionalidades del sistema.
<b>Comentarios</b>	

**Tabla 23: Actor administrador.**

<b>ACT-&lt;02</b>	<b>No registrado</b>
<b>Versión</b>	<i>0.1(11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	Este actor representa al usuario que accede externamente al sistema. Este actor no tendrá privilegios y solo podrá acceder a métodos de carácter público de la aplicación.
<b>Comentarios</b>	

**Tabla 24: Actor usuario.**

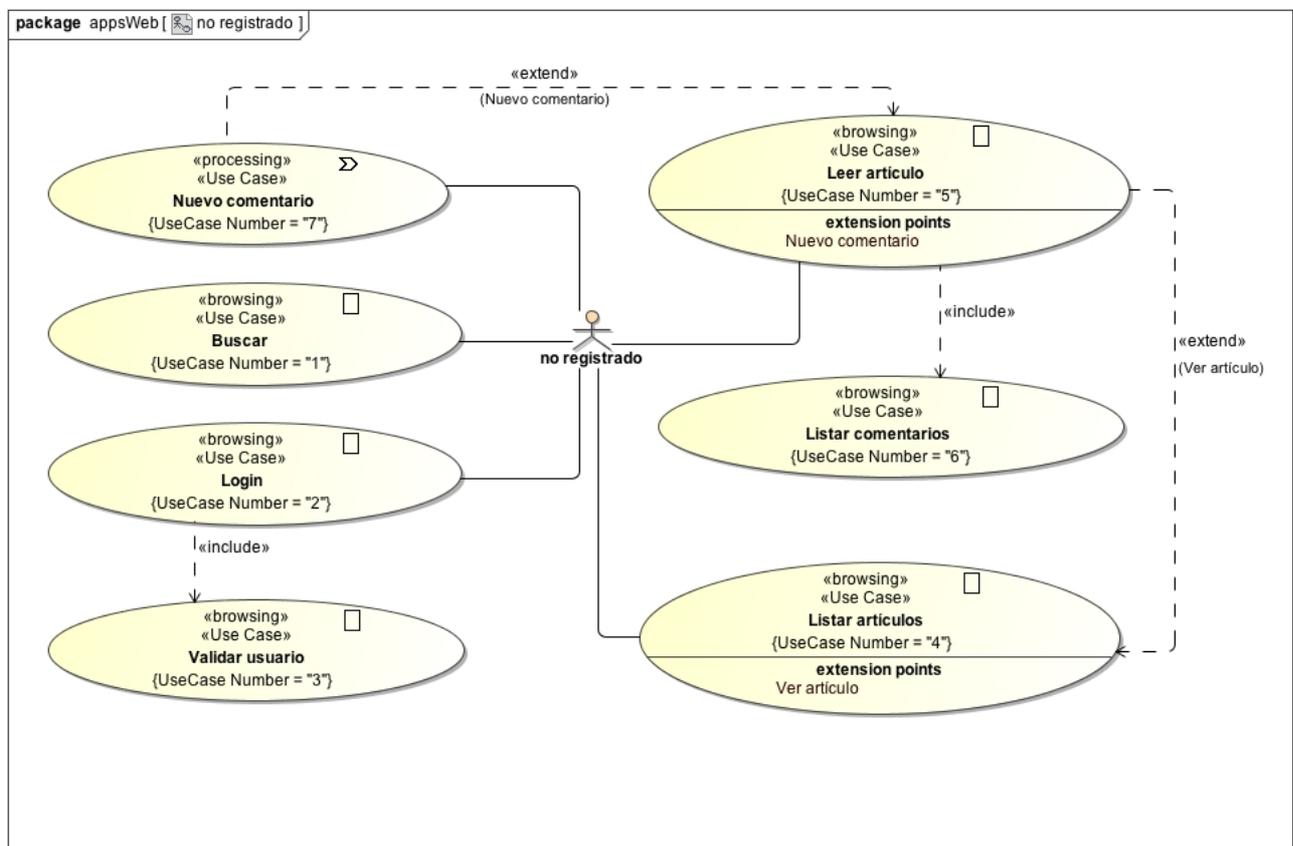


### 8.3.2.2 DIAGRAMAS DE CASOS DE USO

En la siguiente figura se plantea de forma general un diagrama de subsistemas que componen la aplicación. Este diagrama de subsistemas muestra las partes funcionales de las que se compone el sistema.

En primer lugar vamos a ver que acciones puede realizar un usuario no registrado, para ello hay que observar el siguiente diagrama de casos de uso. Este usuario podrá buscar parte del contenido del sistema, leer y listar los artículos, ver e insertar comentarios. También podrá realizar la acción de logearse para entrar como administrador.

La mayoría de las acciones que puede ejecutar un usuario no registrado, no implican una alteración de los valores de la base de datos, sino que para el son datos de lectura exclusivamente. La excepción se da al poder introducir comentario, que inserta una nueva fila en la tabla comentarios, alterando dicha tabla.



**Ilustración 9:** Diagrama de casos de uso del usuario no registrado

En este segundo diagrama vemos las acciones disponibles para un usuario administrador, podrá crear, modificar y borrar plantillas, podrá añadir, modificar y borrar páginas, por último podrá eliminar comentarios.

Todas las acciones que realiza un administrador suponen modificar, eliminar o añadir algún dato del almacén.

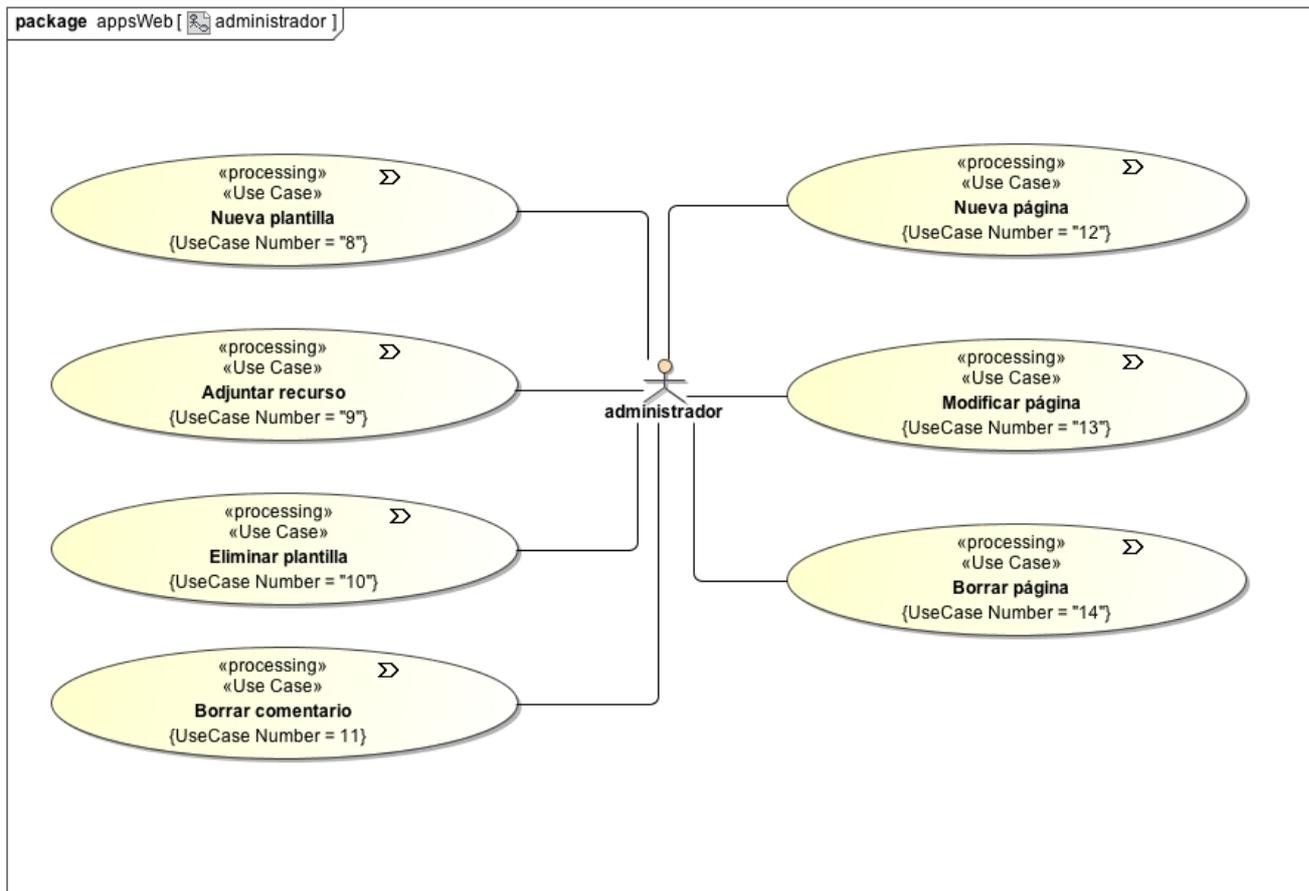


Ilustración 10: Diagrama de casos de uso del usuario administrador

### 8.3.2.3 Casos de uso para usuario “No registrado”

UC-01	Buscar	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido	
<b>Requisitos asociados</b>	IRQ-04 Información sobre contenido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando buscar.	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	Introducir criterios de búsqueda
	p2	Establecer conexión con la base de datos
	p3	Realizar consulta
	p4	Generar interface con los datos recibidos
	p5	El sistema muestra el contenido por pantalla
<b>Postcondición</b>	Resultados mostrados en pantalla	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p3	Si la consulta devuelve <i>null</i> se envía una notificación al usuario.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

Tabla 25: UC-01 Buscar

UC-02	Login	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-02 Gestionar permisos	
<b>Requisitos asociados</b>	IRQ-01 Información sobre usuario	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando logear un usuario.	
<b>Precondición</b>	El usuario no está logeado	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	Introducir datos
	p2	Validar datos. Ejecutar UC-03 Validar usuario
	p3	Si los datos son validados entonces se redirige a la zona de administración
<b>Postcondición</b>	Visualizar zona de administración	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si las datos introducidos no son validados, el caso de uso se vuelve a ejecutar
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

Tabla 26: UC-02 Login

UC-03	<b>Validar Usuario</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-02 Gestionar permisos	
<b>Requisitos asociados</b>	IRQ-01 Información sobre usuarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se valide un usuario.	
<b>Precondición</b>	Los datos de <i>username</i> y <i>contraseña</i> deben ser conocidos	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema establece conexión con la base de datos
	p2	El sistema realiza la consulta en la tabla <i>users</i> .
	p3	El sistema envía una respuesta confirmando los credenciales
<b>Postcondición</b>	Usuario logeado	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si la consulta devuelve un resultado vacío o null el caso de uso queda sin efecto y se devuelve una excepción.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 27: UC-03 Validar usuario**

UC-04	<b>Listar artículos</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido	
<b>Requisitos asociados</b>	IRQ-04 Información sobre contenido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se valide un usuario.	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema solicita al gestorBD un listado con todas los artículos
	p2	El sistema muestra por pantalla un listado con los artículos
	p3	Al mostrar por pantalla los artículos se podrá: <ul style="list-style-type: none"> <li>• Leer artículo. UC-05 Leer Artículo</li> </ul>
<b>Postcondición</b>	Listado de artículos	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p1	Si la consulta devuelve un resultado vacío o null el caso de uso queda sin efecto y se devuelve una excepción.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 28: UC-04 Listar artículos**

UC-05	<b>Leer artículo</b>
<b>Versión</b>	0.1(11/05/2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido

<b>Requisitos asociados</b>	IRQ-04 Información sobre contenido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera leer un artículo	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
g		gg
		p2 El sistema solicita al sistema gestorBD la información del artículo y lo muestra por pantalla
<b>Postcondición</b>	Se muestra el artículo por pantalla	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 29: UC-05 Leer artículo**

<b>UC-06</b>	<b>Listar comentarios</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-05 Gestionar comentarios	
<b>Requisitos asociados</b>	IRQ-06 Información sobre comentario	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se valide un usuario.	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
g		
		p2 El sistema muestra por pantalla un listado con los artículos
		p3 Al mostrar por pantalla los artículos se podrá: <ul style="list-style-type: none"> <li>• Nuevo comentario. UC-07 Nuevo comentario</li> </ul>
<b>Postcondición</b>	Usuario logeado	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p1	Si la consulta devuelve un resultado vacío o null el caso de uso queda sin efecto y se devuelve una excepción.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 30: UC-06 Listar comentarios**

<b>UC-07</b>	<b>Nuevo comentario</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-05 Gestionar comentarios	
<b>Requisitos asociados</b>	IRQ-10 Información sobre comentario	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera añadir un nuevo comentario	
<b>Precondición</b>	Identificador del artículo	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
<b>g</b>	p1	Introducir los datos, en concreto <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Email</li> <li>• Texto</li> <li>•</li> </ul>
	p2	El sistema verifica que los campos no están vacíos y el email es válido
	p3	El sistema solicita al gestorBD insertar un nuevo comentario en la base de datos para el artículo con id=identificador
	p4	El sistema devuelve un mensaje de éxito de la operación.
<b>Postcondición</b>	El artículo cuenta con un comentario más	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si alguno de los campos está vacío, se informa al usuario de que debe rellenarlos y se reinicia el caso de uso
	p3	Si la consulta falla se informa al usuario
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 31: UC-07 Nuevo comentario**

## 8.3.2.4 Casos de uso para usuario “Administrador”

UC-08	Nueva plantilla	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-02 Gestionar plantillas	
<b>Requisitos asociados</b>	IRQ-03 Información sobre plantilla	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera añadir una nueva plantilla	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
<b>g</b>	p1	Introducir los datos, en concreto <ul style="list-style-type: none"> <li>• Autoguardado</li> <li>• Contenido</li> <li>• Nombre</li> <li>• Título</li> </ul>
	p2	El sistema verifica que los campos no están vacíos
	p3	El sistema solicita al gestorBD insertar una nueva plantilla en la base de datos
	p4	El sistema devuelve un mensaje de éxito de la operación.
<b>Postcondición</b>	Nueva plantilla en el sistema	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si alguno de los campos está vacío, se informa al usuario de que debe rellenarlos y se reinicia el caso de uso
	p3	Si la consulta falla se informa al usuario
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

Tabla 32: UC-08 Nuevo comentario

UC-09	Adjuntar recurso	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-02 Gestionar plantillas	
<b>Requisitos asociados</b>	IRQ-03 Información sobre plantilla	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera añadir una nueva plantilla	
<b>Precondición</b>	Identificador de contenido al que se va adjuntar	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
<b>g</b>	p1	Introducir los datos, en concreto <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Fichero (URL)</li> </ul>
	p2	El sistema verifica que los campos no están vacíos
	p3	El sistema solicita al gestorBD insertar un nuevo recurso adjuntado al contenido solicitado.
	p4	El sistema devuelve un mensaje de éxito de la operación.
<b>Postcondición</b>	El contenido cuenta con un nuevo recurso	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

	p2	Si el alguno de los campos está vacío, se informa al usuario de que debe rellenarlos y se reinicia el caso de uso
<b>Importancia</b>	p3	Si la consulta falla se informa al usuario
<b>Estado</b>	Alta	
<b>Estabilidad</b>	Pendiente de validación	
	Alta	

**Tabla 33: UC-09 Adjuntar recurso**

<b>UC-10</b>	<b>Eliminar plantilla</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-02 Gestionar plantillas	
<b>Requisitos asociados</b>	IRQ-03 Información sobre plantilla	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera borrar una plantilla del sistema	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema solicita el identificador de la plantilla
	p2	El sistema pide confirmación del borrado mostrando su identificador.
	p3	El sistema solicita al gestorBD eliminar la plantilla de la base de datos.
	p4	El sistema devuelve un mensaje de éxito en la operación.
<b>Postcondición</b>	El sistema cuenta con una plantilla menos	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si no se confirma el borrado el caso de uso termina. A continuación, el caso de uso queda sin efecto.
	p4	Si la plantilla no puede ser borrada de la base de datos, se envía un mensaje de error al administrador. A continuación, este caso de uso queda sin efecto.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 34: UC-10 Eliminar plantilla**

<b>UC-11</b>	<b>Eliminar comentario</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-05 Gestionar comentarios	
<b>Requisitos asociados</b>	IRQ-06 Información sobre comentarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera borrar un comentario	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema solicita el identificador del comentario
	p2	El sistema pide confirmación del borrado mostrando su identificador.
	p3	El sistema solicita al gestorBD eliminar el comentario

		de la base de datos.
<b>Postcondición</b> <b>Excepciones</b>	p4	El sistema devuelve un mensaje de éxito en la operación.
		El artículo cuenta con un comentario menos
	<b>Paso</b> p2	<b>Acción</b> Si no se confirma el borrado el caso de uso termina. A continuación, el caso de uso queda sin efecto.
	p4	Si el comentario no puede ser borrado de la base de datos, se envía un mensaje de error al administrador. A continuación, este caso de uso queda sin efecto.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 35: UC-11 Eliminar comentario**

<b>UC-12</b>	<b>Nueva página</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido	
<b>Requisitos asociados</b>	IRQ-04 Información sobre contenido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera añadir una nueva página en el sistema	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	Introducir los datos, en concreto <ul style="list-style-type: none"> <li>• URL amigable</li> <li>• Idioma</li> <li>• Título</li> <li>• Contenido</li> <li>• Plantilla</li> <li>• Fecha</li> <li>• Incluir en búsquedas</li> <li>• Caché</li> <li>• Tipo</li> <li>• Meta Keywords</li> <li>• META Description</li> <li>• Contenido en Head</li> </ul>
	p2	El sistema verifica que los campos no están vacíos
	p3	El sistema solicita al gestorBD insertar una nueva página en la base de datos
	p4	El sistema devuelve un mensaje de éxito de la operación.
<b>Postcondición</b>	Nueva página en el sistema	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si alguno de los campos está vacío, se informa al usuario de que debe rellenarlos y se reinicia el caso de uso
	p3	Si la consulta falla se informa al usuario
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 36: UC-12 Nueva página**

<b>UC-13</b>	<b>Modificar página</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-03 Gestionar contenido	
<b>Requisitos asociados</b>	IRQ-04 Información sobre contenido	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera modificar una página del sistema	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	Introducir los datos que se quieren modificar, en concreto <ul style="list-style-type: none"> <li>• URL amigable</li> <li>• Idioma</li> <li>• Título</li> <li>• Contenido</li> <li>• Plantilla</li> <li>• Fecha</li> <li>• Incluir en búsquedas</li> <li>• Caché</li> <li>• Tipo</li> <li>• Meta Keywords</li> <li>• META Description</li> <li>• Contenido en Head</li> </ul>
	p2	El sistema toma los valores de los campos
	p3	El sistema solicita al gestorBD insertar una nueva página en la base de datos
	p4	El sistema devuelve un mensaje de éxito de la operación.
<b>Postcondición</b>	Nueva página en el sistema	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	p2	Si el alguno de los campos está vacío, se informa al usuario de que debe rellenarlos y se reinicia el caso de uso
	p3	Si la consulta falla se informa al usuario
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 37: UC-13 Modificar página**

<b>UC-14</b>	<b>Borrar página</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-05 Gestionar comentarios	
<b>Requisitos asociados</b>	IRQ-06 Información sobre comentarios	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera borrar una página	
<b>Precondición</b>		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema solicita el identificador de la página
	p2	El sistema pide confirmación del borrado mostrando su identificador
	p3	El sistema solicita al gestorBD eliminar la página de la

		base de datos
<b>Postcondición</b> <b>Excepciones</b>	p4	El sistema devuelve un mensaje de éxito en la operación.
		El sistema cuenta con una página menos
	<b>Paso</b> p2	<b>Acción</b> Si no se confirma el borrado el caso de uso termina. A continuación, el caso de uso queda sin efecto.
	p4	Si la página no puede ser borrado de la base de datos, se envía un mensaje de error al administrador. A continuación, este caso de uso queda sin efecto.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 38: UC-14 Borrar página**

<b>UC-15</b>	<b>Eliminar recurso</b>	
<b>Versión</b>	0.1(11/05/2012)	
<b>Autores</b>	Diego Yagüe Juárez	
<b>Objetivos asociados</b>	OBJ-04 Gestionar recursos	
<b>Requisitos asociados</b>	IRQ-05 Información sobre recursos	
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quiera borrar un recurso	
<b>Precondición</b>	Identificador del contenido	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	p1	El sistema solicita el identificador del recurso
	p2	El sistema pide confirmación del borrado mostrando su identificador
	p3	El sistema solicita al gestorBD eliminar el contenido de la base de datos
	p4	El sistema devuelve un mensaje de éxito en la operación.
<b>Postcondición</b> <b>Excepciones</b>		El sistema cuenta con un recurso menos
	<b>Paso</b> p2	<b>Acción</b> Si no se confirma el borrado el caso de uso termina. A continuación, el caso de uso queda sin efecto.
	p4	Si el recurso no puede ser borrado de la base de datos, se envía un mensaje de error al administrador. A continuación, este caso de uso queda sin efecto.
<b>Importancia</b>	Alta	
<b>Estado</b>	Pendiente de validación	
<b>Estabilidad</b>	Alta	

**Tabla 39: UC-15 Eliminar recurso**

### 8.3.2.5 REQUISITOS NO FUNCIONALES

<b>NFR-01</b>	<b>Flexibilidad</b>
<b>Versión</b>	<i>0.1(11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema debe ser diseñado y construido con los mayores niveles de flexibilidad en cuanto a la parametrización de los tipos de datos, de tal manera que la administración del sistema sea realizada por un administrador funcional del sistema.
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta

**Tabla 40: Requisito no funcional de flexibilidad.**

<b>NFR-&lt;02</b>	<b>Portabilidad</b>
<b>Versión</b>	<i>0.1(11/05/2012)</i>
<b>Autores</b>	<i>Diego Yagüe Juárez</i>
<b>Descripción</b>	El sistema debe de ser fácilmente portable a los navegadores Mozilla Firefox, Opera y Google Chrome.
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta

**Tabla 41: Requisito no funcional de portabilidad.**

## 2.20 MATRIZ DE RASTREABILIDAD

	<i>OBJ-01</i>	<i>OBJ-02</i>	<i>OBJ-03</i>	<i>OBJ-04</i>	<i>OBJ-05</i>
IRQ-01			•		
IRQ-02		•			
IRQ-03	•				
IRQ-04			•		
IRQ-05				•	
IRQ-06					•
UC-01			•		
UC-02		•			
UC-03		•			
UC-04			•		
UC-05			•		
UC-06					•
UC-07					•
UC-08	•				
UC-09				•	
UC-10	•				
UC-11					•
UC-12			•		
UC-13			•		
UC-14			•		
UC-15				•	

**Tabla 42: Matriz de rastreabilidad**



## 9 RESUMEN

TIPO	ID	Descripcion
<b>OBJETIVOS</b>	OBJ - 01	<i>Gestionar plantillas</i>
	OBJ –02	<i>Gestionar permisos</i>
	OBJ –03	<i>Gestionar contenido</i>
	OBJ –04	<i>Gestionar recursos</i>
	OBJ –05	<i>Gestionar comentarios</i>
<b>REQUISITOS INFORMACION</b>	IRQ - 01	<i>Información sobre usuarios</i>
	IRQ – 02	<i>Información sobre grupo</i>
	IRQ – 03	<i>Información sobre plantillas</i>
	IRQ – 04	<i>Información sobre contenidos</i>
	IRQ – 05	<i>Información sobre recursos</i>
	IRQ – 06	<i>Información sobre comentarios</i>
<b>REQUISITOS FUNCIONALES</b>	UC - 01	<i>Buscar</i>
	UC – 02	<i>Login</i>
	UC – 03	<i>Validar usuario</i>
	UC – 04	<i>Listar artículos</i>
	UC – 05	<i>Leer artículo</i>
	UC – 06	<i>Listar comentarios</i>
	UC – 07	<i>Nuevo comentario</i>
	UC – 08	<i>Nueva plantilla</i>
	UC – 09	<i>Adjuntar recurso</i>
	UC – 10	<i>Eliminar plantilla</i>
	UC – 11	<i>Eliminar comentario</i>
	UC – 12	<i>Nueva página</i>
	UC – 13	<i>Modificar página</i>
	UC – 14	<i>Eliminar página</i>
	UC – 15	<i>Eliminar recurso</i>
<b>REQUISITOS NO FUNCIONALES</b>	NFR - 01	<i>Flexibilidad</i>
	NFR – 02	<i>Portabilidad</i>

**Tabla 43: Tabla de resumen del análisis.**

## 10 ÍNDICE DE TABLAS

TABLA 1: OBJETIVO 01 - GESTIÓN DE PLANTILLAS.....	47
TABLA 2: OBJETIVO 02 - GESTIONAR PERMISOS. ....	47
TABLA 3: OBJETIVO 02.01 - GESTIONAR GRUPOS.....	47
TABLA 4: OBJETIVO 02.02 - GESTIONAR USUARIOS.....	48
TABLA 5: OBJETIVO 03 – GESTIONAR CONTENIDO .....	48
TABLA 6: OBJETIVO 04 - GESTIONAR RECURSO.....	48
TABLA 7: OBJETIVO 05 - GESTIONAR COMENTARIO .....	48
TABLA 8: IRQ 01 – INFORMACIÓN SOBRE USUARIO. ....	49
TABLA 9: IRQ 02 – INFORMACIÓN SOBRE GRUPO.....	49
TABLA 10: IRQ 03 – INFORMACIÓN SOBRE PLANTILLAS.....	50
TABLA 11: IRQ 04 – INFORMACIÓN SOBRE CONTENIDO. ....	50
TABLA 12: IRQ 05 – INFORMACIÓN SOBRE RECURSO.....	51
TABLA 13: IRQ 06 – INFORMACIÓN SOBRE COMENTARIO. ....	51
TABLA 14: ACTOR ADMINISTRADOR.....	53
TABLA 15: ACTOR USUARIO.....	53
TABLA 16: UC-01 BUSCAR .....	57
TABLA 17: UC-02 LOGIN .....	57
TABLA 18: UC-03 VALIDAR USUARIO .....	58
TABLA 19: UC-04 LISTAR ARTÍCULOS.....	58
TABLA 20: UC-05 LEER ARTÍCULO .....	59
TABLA 21: UC-06 LISTAR COMENTARIOS.....	59
TABLA 22: UC-07 NUEVO COMENTARIO.....	60
TABLA 23: UC-08 NUEVO COMENTARIO.....	61
TABLA 24: UC-09 ADJUNTAR RECURSO.....	62
TABLA 25: UC-10 ELIMINAR PLANTILLA.....	62
TABLA 26: UC-11 ELIMINAR COMENTARIO .....	63
TABLA 27: UC-12 NUEVA PÁGINA .....	63
TABLA 28: UC-13 MODIFICAR PÁGINA .....	64
TABLA 29: UC-14 BORRAR PÁGINA.....	65
TABLA 30: UC-15 ELIMINAR RECURSO.....	65
TABLA 31: REQUISITO NO FUNCIONAL DE FLEXIBILIDAD. ....	66
TABLA 32: REQUISITO NO FUNCIONAL DE PORTABILIDAD. ....	66
TABLA 33: MATRIZ DE RASTREABILIDAD .....	67
TABLA 34: TABLA DE RESUMEN DEL ANÁLISIS.....	69

## 11 INDICE DE ILUSTRACIONES

<b>ILUSTRACIÓN 1:</b> DIAGRAMA DE CASOS DE USO DEL USUARIO NO REGISTRADO.....	55
<b>ILUSTRACIÓN 2:</b> DIAGRAMA DE CASOS DE USO DEL USUARIO ADMINISTRADOR .....	56



**BLOQUE II**

---

**DOCUMENTACIÓN TÉCNICA**

---

**PARTE II - DISEÑO**



**ÍNDICE**

2	DISEÑO DEL SISTEMA	67
2.1	INTRODUCCIÓN.	69
2.2	GOOGLE APP ENGINE	69
2.3	DISEÑO DE LA ESTRUCTURA DE DATOS	70
2.4	MOTIVACIÓN EN LA PERSPECTIVA DE DISEÑO	77
2.4.1	SISTEMA GESTOR DE CONTENIDOS	78
2.5	MODELO DE COMPORTAMIENTO: DISEÑO ESTRUCTURAL	81
2.5.1	DIAGRAMA DE TIPOS DE OBJETOS	81
2.5.2	DIAGRAMA DE ACTIVIDADES	87
2.6	MODELO DE COMPORTAMIENTO: DISEÑO DINÁMICO	91
3	IMPLEMENTACIÓN	101
3.1	MODELO DE DESPLIEGUE DEL SISTEMA	103
3.2	TECNOLOGÍAS UTILIZADAS PARA LA IMPLEMENTACIÓN	104
4	PRUEBAS	107
5	RESUMEN	107
6	ÍNDICE DE TABLAS	110
7	ÍNDICE DE FIGURAS	111



# **2 DISEÑO DEL SISTEMA**

---



## 2.1 INTRODUCCIÓN.

Una vez realizado el análisis de requisitos de información y requisitos funcionales, descritos mediante casos de uso, se tiene que realizar un estudio mas detallado del funcionamiento del sistema, es decir, un análisis de diseño que tiene como objetivo ilustrar al lector cómo se comportan los elementos que componen la aplicación.

Se va a realizar un estudio en dos fases que componen el diseño del sistema. Por una parte se va analizar el comportamiento estático del sistema, y por otra parte se va a realizar un análisis del comportamiento dinámico del sistema.

El diseño del sistema esta condicionado por una herramienta llamada Google App Engine que se ha utilizado como apoyo para la aplicación.

En resumen, podemos elaborar una lista con los apartados mas importantes que componen este bloque.

- Google App Engine
- Diseño de la estructura de datos
- Modelo de comportamiento del sistema.
  - Comportamiento estático:
    - Diagramas de clases
    - Modelos de la base de datos
  - Comportamiento dinámico
    - Diagramas de secuencia
    - Diagramas de estado

## 2.2 GOOGLE APP ENGINE

Google App Engine permite ejecutar aplicaciones web en la infraestructura de Google.

Google App Engine admite aplicaciones escritas en varios lenguajes de programación. Gracias al entorno de tiempo de ejecución Java de App Engine, se puede crear una aplicación a través de tecnologías Java estándar, que incluyen JVM, servlets Java y el lenguaje de programación Java, o cualquier otro lenguaje que utilice un intérprete o compilador basado en JVM como, por ejemplo, JavaScript o Ruby. App Engine también ofrece un entorno de tiempo de ejecución Python, que incluye un rápido intérprete Python y la biblioteca estándar Python. Los entornos de tiempo de ejecución Java y Python se generan para garantizar que la aplicación se ejecute de forma rápida, segura y sin interferencias de otras aplicaciones en el sistema.

Para el desarrollo de nuestro sistema se ha optado por utilizar el entorno de ejecución Java porque se pueden utilizar herramientas de desarrollo web como Spring, JBoss, etc. y estándares del API conocidos. La aplicación interactúa con el entorno a través del estándar Java Servlet y puede utilizar tecnologías de aplicación Web conocidas como, por ejemplo, JavaServer Pages (JSP).

El almacén de datos de App Engine es un almacén de datos de objetos sin esquema, que dispone de un motor de consultas y transacciones atómicas. El SDK de Java incluye implementaciones de las interfaces de objetos de datos Java (JDO, Java Data Objects) y del API de persistencia Java (JPA, Java Persistence API), así como un API del almacén de datos de nivel inferior.

App Engine incluye las siguientes funcionalidades:

- Servidor web dinámico, totalmente compatible con las tecnologías web más comunes
- Almacenamiento permanente con funciones de consulta, clasificación y transacciones
- Escalado automático y distribución de carga
- API para autenticar usuarios y enviar correo electrónico a través de Google Accounts
- Un completo entorno de desarrollo local que simula Google App Engine en tu equipo
- Colas de tareas que realizan trabajos fuera del ámbito de una solicitud web
- Tareas programadas para activar eventos en momentos determinados y en intervalos regulares.

### 2.2.1 SISTEMA GESTOR DE BASES DE DATOS

*App Engine* proporciona un potente servicio de almacenamiento de datos distribuido que incluye un motor de búsqueda y transacciones. A medida que el servidor web distribuido crece con el tráfico, el almacén de datos distribuido crece con los datos.

El almacén de datos de *App Engine* no es como una base de datos relacional tradicional. Los objetos de datos, o "entidades", disponen de un tipo y un conjunto de propiedades. Las consultas pueden recuperar entidades de un tipo determinado filtradas y ordenadas según los valores de las propiedades.

Las entidades del almacén de datos carecen de esquema. El código de aplicación se encarga de proporcionar y de aplicar la estructura de las entidades de datos.

A continuación explicaremos la especificación de la estructura de la base de datos, con ello se consigue que el lector tenga una idea de los elementos que formaran la base de datos, incluyendo una meta descripción de cada elemento, se informa de las relaciones que existen entre los elementos y como están agrupados.

Una primera aproximación, con la que tener una idea, de cómo va a ser nuestro modelo de datos, es realizando un esquema entidad/relación compuesto de entidades y relaciones, acompañado de un diccionario de datos.

## 2.3 DISEÑO DE LA ESTRUCTURA DE DATOS

En esta sección explicaremos los elementos que componen el modelo de datos. Con ello pretendemos dar a conocer el aspecto organizativo de las fuentes de información que tendremos que desarrollar.

Se va a realizar un diseño conceptual del sistema incluyendo los elementos y las relaciones que van a formar nuestro entorno para tener una idea global de su funcionamiento.

Para su realización se ha tenido en cuenta la fase de análisis realizada previamente, sujeto a sus restricciones y requisitos.

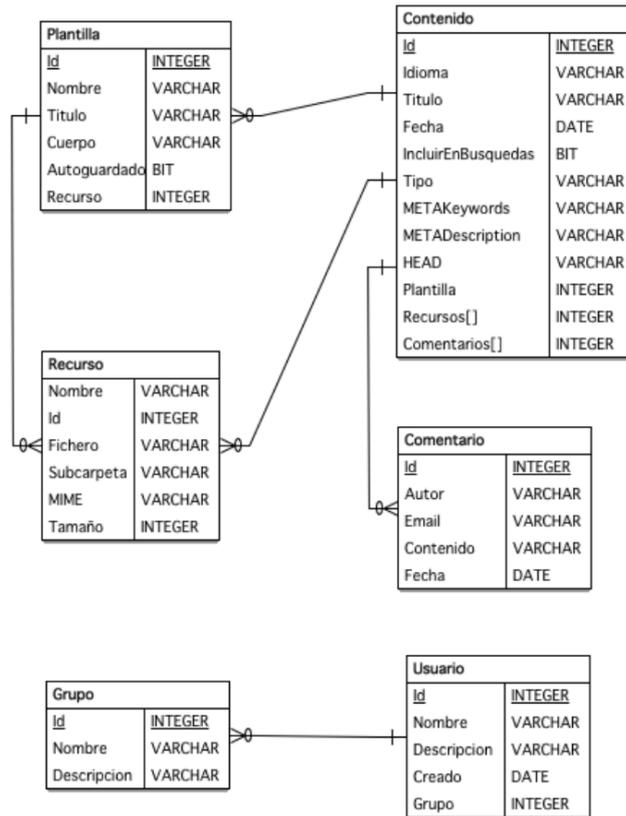


Ilustración 11: Modelo de la base de datos

### 2.3.1 DICcionario DE DATOS

#### 2.3.1.1 ENTIDADES

##### Entidad: Plantilla

Descripción: Representa las plantillas que se pueden crear en el sistema, para que sirvan como elemento base para el contenido.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Nombre	dNombre	Nombre de la plantilla		Único
Título	dCadena	Título de la plantilla	Desconocido	No nulo
Cuerpo	dDescripción	Cuerpo de la plantilla	NULL	
Autoguardado	dBoolean	Boolean que define si se realiza autoguardado	NULL	

Tabla 44: Atributos de la entidad plantilla

**Entidad: Contenido**

Descripción: Representa el contenido que se puede agregar al sistema, puede representar cualquier tipo de contenido.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Título	dTitulo	Título del contenido	NULL	
Idioma	dNombre	Nombre del contenido		Único
Descripción	dDescripcion	Descripción detallada	Desconocido	No nulo
Fecha	dFecha	Fecha de creación	Desconocido	No nulo
IncluirEnBusqueda	dBoolean	Indica si se incluye en busquedas	No	
Tipo	dCadena	Tipo de plantilla	Desconocido	No nulo
METAKeywords	dCadena	Palabras clave	Desconocido	No nulo
METADescripcion	dCadena	Descripción más detallada	Desconocido	No nulo
HEAD	dCadena	Texto a incluir en el HEAD	Desconocido	No nulo

**Tabla 45: Atributos de la entidad contenido**

**Entidad: Recurso**

Descripción: Representa un recurso que se sube desde el PC local.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Nombre	dNombre	Nombre de la plantilla		Único
Fichero	dCadena	Dirección donde se encuentra el fichero	Desconocido	No nulo
Subcarpeta	dCadena	Subcarpeta donde se incluye el fichero	Desconocido	
MIME	dCadena	Tipo MIME del fichero	Desconocido	No nulo
Tamaño	dEntero	Tamaño del fichero	Desconocido	

**Tabla 46: Atributos de la entidad condición**

**Entidad: Comentario**

Descripción: Representa los comentarios que pueden realizar los usuarios.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Autor	dCadena	Autor comentario		No nulo
Email	dEmail	Descripcion detallada	Desconocido	No nulo
Contenido	dCadena	Contenido del comentario		

**Tabla 47:** Atributos de la entidad usuarios

**Entidad: Usuario**

Descripción: Representa los usuarios del sistema.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Nombre	dCadena	Nombre del usuario		No nulo
Contraseña	dpassword	Hora en la que se cierra el museo y el horno	15:00	No nulo
Creado	dFecha	Indica si permanecera cerrado el museo y el horno	No	

**Tabla 48:** Atributos de la entidad usuario

**Entidad: Grupo**

Descripción: Representa los grupos de usuarios del sistema.

<i>ATRIBUTOS</i>	<i>DOMINIO</i>	<i>DESCRIPCIÓN</i>	<i>DEFECTO</i>	<i>RESTRICCIONES</i>
Identificador	dIdentificador	Identificador único de la tupla en la entidad	Auto-Increment	Clave principal
Nombre	dCadena	Nombre del grupo	Desconocido	
Descripción	dDescripcion	Descripción del grupo		No nulo

**Tabla 49:** Atributos de la entidad grupo

### 2.3.1.2 Relaciones

#### Relación: Adjuntar

- **Descripción:** Describe la relación existente entre plantilla y recurso. Expresa que una plantilla puede adjuntar recursos.
- **Restricciones:** Una plantilla puede adjuntar muchos recursos, y un recurso puede ser adjuntado a varias plantillas.

#### Relación: Adjuntar2

- **Descripción:** Describe la relación existente entre contenido y recurso. Expresa que un contenido puede adjuntar recursos
- **Restricciones:** Un contenido puede adjuntar muchos recursos, y un recurso puede ser adjuntado a varios contenidos.

#### Relación: Incluye

- **Descripción:** Describe la relación existente entre contenido y comentario. Expresa que el contenido puede incluir comentarios.
- **Restricciones:** El contenido puede tener muchos comentarios, y un comentario solo es incluido en un contenido.

#### Relación: Pertenece

- **Descripción:** Describe la relación existente entre usuario y grupo. Expresa el concepto de tener un usuario inscrito en grupo.
- **Restricciones:** Un usuario solo puede inscribirse en un grupo, y en grupo se pueden inscribir uno o mas usuarios.

#### Dominios

<i>DOMINIO</i>	<i>TIPO</i>	<i>FORMATO</i>	<i>VALORES</i>	<i>DESCRIPCIÓN</i>
dIdentificador	Entero	Int	Entero [1..429.496.265]	Identificador de la ocurrencia.
dDescripción	Texto	Text	Cadena de 0 hasta 65535 caracteres.	Descripción de la ocurrencia
dCadena	Cadena	Varchar(255)	Cadena de 0 hasta 255 caracteres	
dFecha	Fecha	Time		Fecha(A/M/D)
dEntero	Entero	Int	Entero [1..429.496.265]	Se corresponde con los enteros.
dPassword	Cadena	Varchar(9)	Cadena de 0 hasta 9 caracteres	Campo utilizado para almacenar la contraseña de usuario.
dEmail	Cadena	Varchar(20)	Cadena de 0 hasta 20 caracteres	Email de contacto

**Tabla 50:** Dominios de los tipos de atributos

## **2.4 DISEÑO DE LA APLICACIÓN**



## 2.4.1 MOTIVACIÓN EN LA PERSPECTIVA DE DISEÑO

- Lo primero a lo que debo hacer referencia es que el sistema desarrollado esta enfocado a ser una aplicación de gestión, y estas aplicaciones fundamentalmente se basan en: alta, baja, modificación, consulta y listado de las entidades de información manejadas por el sistema.

Esto ocasiona que a nivel conceptual sea un diseño muy repetitivo, con lo que carece de sentido llenar la documentación con diagramas similares sin utilidad para la implementación.

- Diseño bajo el patrón Modelo Vista Controlador (MVC)

El diseño del sistema impone el uso del patrón MVC (Modelo-Vista-Controlador) y proporcionando una serie de clases básicas, obligando a extender esta funcionalidad básica de una forma determinada.

El patrón trata de desacoplar los objetos del dominio (Modelo), las ventanas (Vista) y los manejadores (Controlador), a fin de tratar que los objetos del dominio sean utilizados lo menos posible y reducir el impacto que los cambios en la interfaz tienen con respecto a ellos.

Así definiremos los modelos para que no exista acoplamiento y tenga mínima visibilidad con respecto a las vistas.

Los controladores son definidos para que manejen los eventos o peticiones al sistema, y que redirija el procesamiento de datos y la visualización de resultado



**Ilustración 12:** Esquema general del patrón Modelo-Vista-Controlador

- Un controlador se usa para manejar la lógica de cierta sección de su aplicación. Comúnmente, los controladores son usados para manejar la lógica de un solo modelo.

Los controladores pueden tener cualquier cantidad de métodos a los que normalmente se les llama acciones. Las acciones son métodos de los controladores para mostrar vistas. Una acción es un único método de un controlador.

- Un Modelo representa un modelo de datos y, en programación orientada a objetos, es un objeto que representa una "cosa".
- La capa vista es cómo hablas a los usuarios. La mayor parte del tiempo las vistas estarán mostrando documentos (X)HTML a los navegadores.

### 2.4.2 Sistema gestor de contenidos

Se ha implementado un sistema de gestión de contenidos (CMS) que permite construir sitios web y aplicaciones en línea en la plataforma Google App Engine para Java.

Entre las características de un CMS se incluyen internacionalización, URLs amigables, intercambio de páginas, comentar (con moderación), control de versiones de contenido, las formas de entrada, código de la imagen, Google Analytics, plug-ins opcionales, seguridad y gestión de grupos, estructura de páginas padre/hijo, plantillas programables, gestión de recursos, múltiples modos de edición (visual, la sintaxis wiki, texto plano) y mucho más, además de alojamiento gratuito en la nube de Google.

¿Qué es un sistema de gestión de contenidos (CMS)?

Un sistema de gestión de contenidos (CMS) es un conjunto de procedimientos utilizados para administrar el flujo de trabajo en un entorno de colaboración. Un CMS Web es un software que puede rastrear todo el contenido y los recursos de un sitio web - al igual que una biblioteca pública lleva un seguimiento de todos los libros y las grabaciones en las estanterías. Los componentes pueden ser de contenido como texto, imágenes, scripts y hojas de estilo, o plantillas que el contenido en formato de pantalla.

Una gran ventaja de usar un CMS para mantener un sitio es que las páginas se pueden editar con el mismo navegador utilizado para ver el sitio. Se pueden hacer cambios y revisar instantáneamente, con un built-in, basada en menú, editor visual. Desde la CMS administra todo el contenido en un solo lugar, todos los activos de su sitio es fácil de encontrar y actualizar, sin tener que aprender las tecnologías especiales en la Web.

Características principales:

- URL amigables
- Seguimiento de publicaciones por fecha
- CKEditor 3.0
- Moderación y manejo de comentarios
- Versionado
- Previsualización de páginas
- Gestión de páginas hijas de un padre.
- Plantillas
- Subida y exportación de recursos

- Configuración para Google Analytics
- Soporte para capucha
- Soporte de plugins
- RSS

## CONCEPTOS GENERALES

- Contenido de páginas:

Área de contenido CMS. El sistema le pedirá que inicie sesión, y luego poder acceder a la página de bienvenida en el área de trabajo. Para editar el contenido, hay que seleccionar las páginas de contenido de la página de bienvenida o en el enlace contenido en el encabezado.

-Árbol de páginas:

Una página es un fragmento de contenido. El contenido está enlazado a una plantilla para crear una página web completa que puede ser interpretada por un navegador.

El CMS organiza las páginas usando una jerarquía padre/hijo. En el área de trabajo, la jerarquía se representa como un esquema de árbol. Inicialmente, el árbol está contraído.

Para abrir cada rama del árbol, seleccione el botón [+] control.

Para agregar una carpeta de la vista de árbol, pulse el botón [+] control.

Para abrir una página para la edición, seleccione el enlace de nombre de la página.

-Área de trabajo :

El área de trabajo de las páginas se organiza en cinco pestañas: Página, contenido, páginas hijas, comentarios y seguridad.

-Ficha

La ficha de página establece los atributos de título, URL amigable, plantilla, la fecha de publicación, habilitar los comentarios, y el tipo de página.

-Título

El atributo *Title* se muestra en el árbol de páginas, así como en la historia del navegador. Puede o no puede coincidir con el título H1.

-Plantilla

La plantilla proporciona el formato Web que rodea el contenido de la página. Cada página debe estar asociada con una plantilla. Al diseñar un sitio se puede proporcionar una plantilla personalizada que satisfaga las necesidades específicas de su sitio.

### -Habilitar comentarios

Los comentarios pueden ser habilitado para cualquier página. La plantilla de página debe ser compatible con los comentarios y los comentarios deben estar habilitadas para cada página. El marcado de bloque de comentario se puede personalizar en la configuración del sitio.

### -Tipo de página

El sistema es compatible con dos tipos de páginas, sencilla y estructurada. Las páginas simples tienen un único campo de contenido. Las páginas estructuradas pueden utilizar varios campos para almacenar contenido. La plantilla se puede colocar campos de contenido individuales en lugares específicos, para crear fácilmente un diseño uniforme.

### -Ficha Contenido

La pestaña Contenido permite la edición y aprobación del contenido de la página.

### -Contenido idioma

Si la aplicación se ha configurado para utilizar varios idiomas, una visión diferente de una página se puede crear para cada idioma soportado. Las vistas pueden compartir la página y la configuración de seguridad, pero cada uno tiene un área de contenido independiente.

El lenguaje de presentar detecta automáticamente la configuración de idioma de un navegador Web. El ajuste de idioma se mantendrá durante la sesión del navegador. (Cerrar el navegador o eliminar la cookie JSESSIONID.) Si una vista no está disponible para el idioma preferido, la vista idioma por defecto se presentará en su lugar.

### -Editor de páginas

El CMS utiliza un plugin popular, CKEdit, para crear y mantener contenido de la página. La interfaz es similar a otros productos y debe ser intuitiva para la mayoría de los usuarios.

### -Imagen

El cuadro de diálogo de imagen incluye una acción de servidor de búsqueda que muestra el árbol de recursos. Para enlazar a un recurso, selecciónelo en el árbol, y el editor va a generar la referencia adecuada.

### -Los controles de formulario

Para crear un formulario de entrada de datos, utilice el complemento del form, en lugar de los controles del editor de formularios.

## DISEÑO

### -Diseño de plantillas

Las plantillas controlan como la página se muestra, incluida la presentación gráfica, disposición y colores. Un plantilla de diseño consiste en un componente de contenido que define el HTML de salida de las páginas del sitio, con uno o más ficheros CSS que definen la disposición, fuentes, colores y otros estilos.

### -Recursos estáticos

Los recursos son cualquier contenido estático usado por las plantillas de diseño o por las páginas del sitio. Los recursos puede ser imágenes, archivos CSS, archivos JavaScript, archivos XML, PDFs, Excel, Word, video y cualquier otro recurso. Están organizados dentro de carpetas en un típico sistema de archivos. Para referenciar un recurso deber ir precedido en la URL de “/file” seguido del path de la carpeta y l nombre de archivo.

### -Estructuras de contenido

Las estructuras están formadas para usar un conjunto de elementos consistente. Un página simple recoge información como texto en un campo de entrada. Una página estructurada puede recoger multitud de elementos de varios tipos.

### 2.4.3 MODELO DE COMPORTAMIENTO: DISEÑO ESTRUCTURAL

#### 2.4.3.1 DIAGRAMA DE TIPOS DE OBJETOS

La siguiente ilustración es un diagrama de las principales clases que forman el sistema. Primero queremos disponer de plantillas conteniendo el código básico de la estructura para una página. Además esta plantilla necesitará de elementos auxiliares por lo que se define la clase recurso para poder almacenar estos elementos en forma de ficheros.

La clase página se integra dentro de una plantilla en el espacio reservado para ella. Una página solo puede integrarse con una plantilla. Las paginas pueden ser comentadas por los usuarios, para guardar estos datos se ha creado la clase comentario.

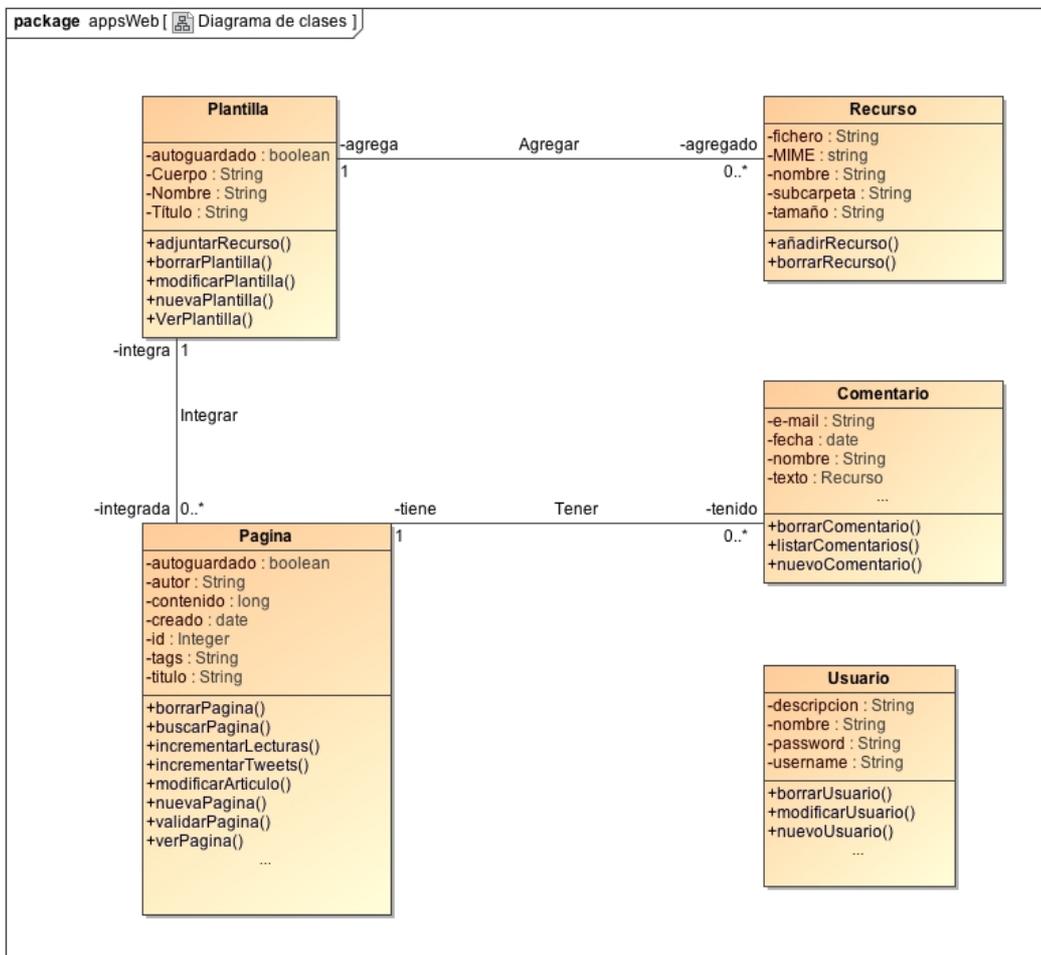


Ilustración 13: Diagrama de clases

### 2.4.3.2 Tipo de objeto Plantilla

#### Descripción del tipo de Objeto Plantilla

<i>Tipo</i>	<i>Plantilla</i>
<b>Versión</b>	1.0(25-05-2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Requisitos</b>	IRQ-03 Información sobre plantilla
<b>Asociados</b>	
<b>Descripción</b>	Este tipo concreto representa a las plantillas almacenadas en el sistema

**Tabla 51:** Tipo de objeto Plantilla

#### Atributos del tipo de Objeto Plantilla

<i>Atributo variable</i>	<i>plantilla::autoguardado</i>
<b>Descripción</b>	Atributo que indica si se quiere escoger la opción de autoguardado mientras se edita una plantilla
<b>Tipo OCL</b>	Boolean
<b>Valor Inicial</b>	0

**Tabla 52:** Atributo autoguardado del objeto Plantilla

<i>Atributo variable</i>	<i>plantilla::cuerpo</i>
<b>Descripción</b>	Atributo donde se almacena todo el código de la plantilla
<b>Tipo OCL</b>	Long text
<b>Valor Inicial</b>	

**Tabla 53:** Atributo cuerpo del objeto Plantilla

<i>Atributo variable</i>	<i>plantilla::nombre</i>
<b>Descripción</b>	Atributo donde se almacena el nombre para identificarla dentro del árbol de archivos
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 54:** Atributo nombre del objeto Plantilla

<i>Atributo variable</i>	<i>plantilla::titulo</i>
<b>Descripción</b>	Atributo que indica el título de la plantilla
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 55:** Atributo titulo del objeto Plantilla

### 2.4.3.3 Tipo de objeto Recurso

#### Descripción del tipo de Objeto Recurso

<i>Tipo</i>	<i>Recurso</i>
<b>Versión</b>	1.0(25-05-2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Requisitos</b>	IRQ-05 Información sobre Recurso
<b>Asociados</b>	
<b>Descripción</b>	Este tipo concreto representa a los recursos almacenados en el sistema

**Tabla 56:** Tipo de objeto Recurso

#### Atributos del tipo de Objeto Recurso

<i>Atributo variable</i>	<i>recurso::fichero</i>
<b>Descripción</b>	Atributo que indica la ruta donde se encuentra el fichero
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 57:** Atributo fichero del objeto Recurso

<i>Atributo variable</i>	<i>recurso::MIME</i>
<b>Descripción</b>	Atributo que señala el tipo de MIME que tiene el archivo
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 58:** Atributo MIME del objeto Recurso

<i>Atributo variable</i>	<i>recurso::nombre</i>
<b>Descripción</b>	Atributo donde se almacena el nombre para identificar al recurso dentro del árbol de archivos
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 59:** Atributo nombre del objeto Recurso

<i>Atributo variable</i>	<i>recurso::subcarpeta</i>
<b>Descripción</b>	Atributo que indica en que subcarpetas se encuentra almacenado el recurso
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 60:** Atributo subcarpeta del objeto Recurso

<i>Atributo variable</i>	<i>recurso::tamano</i>
<b>Descripción</b>	Atributo que indica el tamaño total del archivo en bytes
<b>Tipo OCL</b>	Float
<b>Valor Inicial</b>	

**Tabla 61:** Atributo tamaño del objeto Recurso

### 2.4.3.4 Tipo de objeto Comentario

#### Descripción del tipo de Objeto Comentario

<i>Tipo</i>	<i>Comentario</i>
<b>Versión</b>	1.0(25-05-2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Requisitos</b>	IRQ-06 Información sobre Comentario
<b>Asociados</b>	
<b>Descripción</b>	Este tipo concreto representa a los comentarios almacenados en el sistema

**Tabla 62:** Tipo de objeto Comentario

#### Atributos del tipo de Objeto Comentario

<i>Atributo variable</i>	<i>comentario::email</i>
<b>Descripción</b>	Atributo que indica el correo electrónico del autor del comentario
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 63:** Atributo email del objeto Comentario

<i>Atributo variable</i>	<i>comentario::fecha</i>
<b>Descripción</b>	Atributo que guarda la fecha de publicación del comentario
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	

**Tabla 64:** Atributo fecha del objeto Comentario

<i>Atributo variable</i>	<i>comentario::nombre</i>
<b>Descripción</b>	Atributo donde se almacena el nombre para identificar al comentario dentro del árbol de archivos
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 65:** Atributo nombre del objeto Comentario

<i>Atributo variable</i>	<i>comentario::texto</i>
<b>Descripción</b>	Atributo donde se almacena el input que escribe el usuario
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 66:** Atributo texto del objeto Comentario

### 2.4.3.5 Tipo de objeto Usuario

#### Descripción del tipo de Objeto Usuario

<i>Tipo</i>	<i>Usuario</i>
<b>Versión</b>	1.0(25-05-2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Requisitos</b>	IRQ-01 Información sobre Usuario
<b>Asociados</b>	
<b>Descripción</b>	Este tipo concreto representa a los usuarios almacenados en el sistema

**Tabla 67:** Tipo de objeto Usuario

#### Atributos del tipo de Objeto Usuario

<i>Atributo variable</i>	<i>usuario::descripcion</i>
<b>Descripción</b>	Atributo que indica una breve descripción sobre el usuario
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 68:** Atributo descripción del objeto Usuario

<i>Atributo variable</i>	<i>usuario::password</i>
<b>Descripción</b>	Atributo que guarda la fecha de publicación del usuario
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	

**Tabla 69:** Atributo fecha del objeto Usuario

<i>Atributo variable</i>	<i>usuario::nombre</i>
<b>Descripción</b>	Atributo donde se almacena el nombre para identificar al usuario dentro del árbol de archivos
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 70:** Atributo nombre del objeto Usuario

<i>Atributo variable</i>	<i>usuario::texto</i>
<b>Descripción</b>	Atributo donde se almacena el input que escribe el usuario
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 71:** Atributo texto del objeto Usuario

### 2.4.3.6 Tipo de objeto Página

#### Descripción del tipo de Objeto Página

<i>Tipo</i>	<i>Página</i>
<b>Versión</b>	1.0(25-05-2012)
<b>Autores</b>	Diego Yagüe Juárez
<b>Requisitos</b>	IRQ-04 Información sobre Contenido
<b>Asociados</b>	
<b>Descripción</b>	Este tipo concreto representa a las páginas almacenadas en el sistema

**Tabla 72:** Tipo de objeto Página

#### Atributos del tipo de Objeto Página

<i>Atributo variable</i>	<i>pagina::autoguardado</i>
<b>Descripción</b>	Atributo que se corresponde con la opción de autoguardado en modo edición
<b>Tipo OCL</b>	Boolean
<b>Valor Inicial</b>	0

**Tabla 73:** Atributo autoguardado del objeto Página

<i>Atributo variable</i>	<i>pagina::autor</i>
<b>Descripción</b>	Atributo que señala el autor de la página
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 74:** Atributo autor del objeto Página

<i>Atributo variable</i>	<i>página::contenido</i>
<b>Descripción</b>	Atributo donde se almacena el cuerpo de la página
<b>Tipo OCL</b>	Longtext
<b>Valor Inicial</b>	

**Tabla 75:** Atributo contenido del objeto Página

<i>Atributo variable</i>	<i>página::creado</i>
<b>Descripción</b>	Atributo que indica la fecha de creación de la página
<b>Tipo OCL</b>	Date
<b>Valor Inicial</b>	

**Tabla 76:** Atributo creado del objeto Página

<i>Atributo variable</i>	<i>página::tag</i>
<b>Descripción</b>	Palabras representativas que describan el contenido de la página
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 77:** Atributo tag del objeto Página

<i>Atributo variable</i>	<i>página::titulo</i>
<b>Descripción</b>	Título que va a tener la página cuando la veamos desde un navegador
<b>Tipo OCL</b>	String
<b>Valor Inicial</b>	

**Tabla 78:** Atributo título del objeto Página

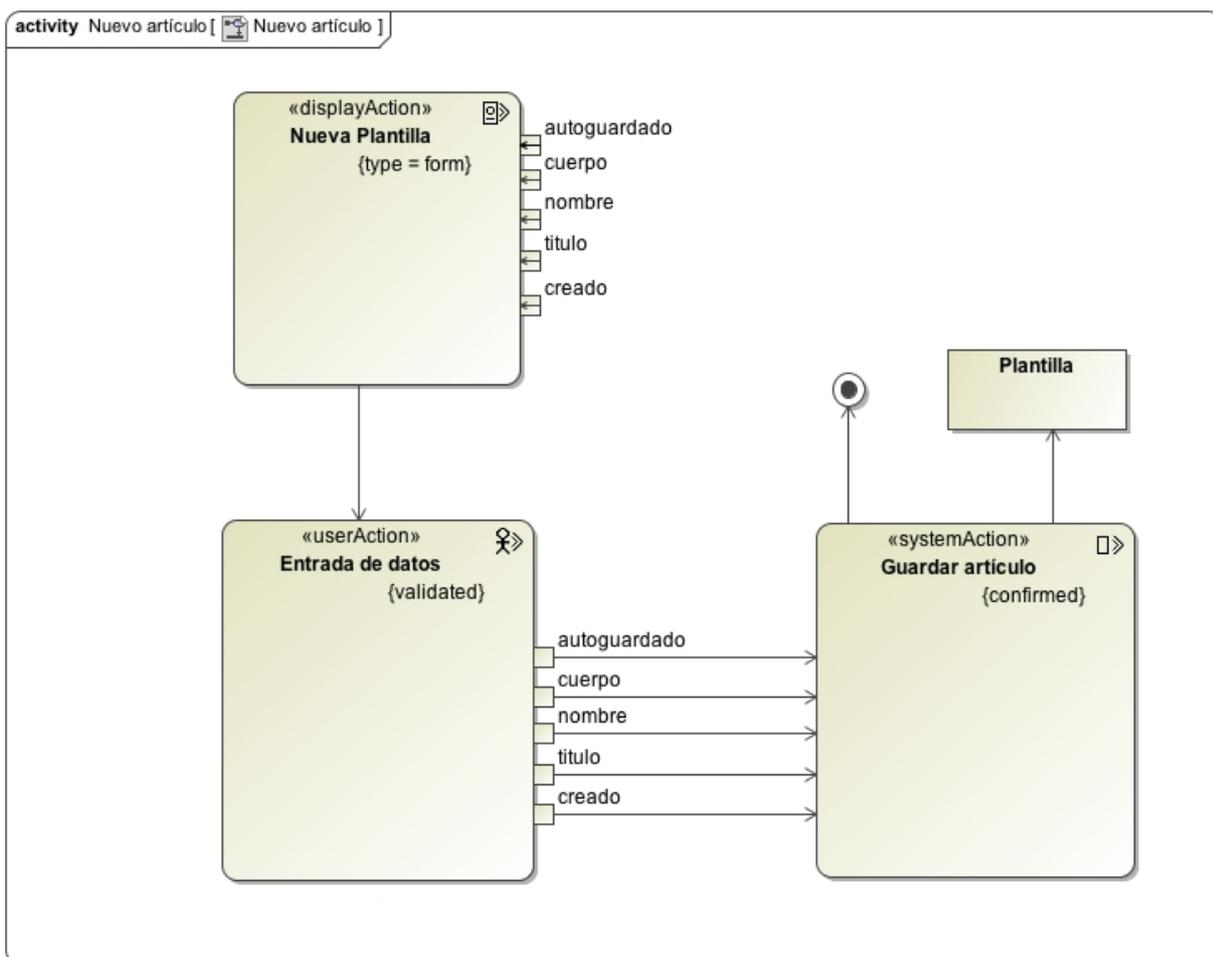
## DIAGRAMA DE ACTIVIDADES

Como con casos de uso solamente es posible capturar poca información, cada caso de uso puede ser descrito más detalladamente mediante un proceso. Es decir, las acciones que son parte de un caso de uso así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades.

Los dos estereotipos «user Action» y «system Action» pueden ser usados análogamente al flujo de procesos. El estereotipo «user Action» es usado para indicar interacciones de usuario en la página web iniciando un proceso o respondiendo a un requisito de información. Por lo contrario, «system Action» describe acciones que son ejecutados por el sistema.

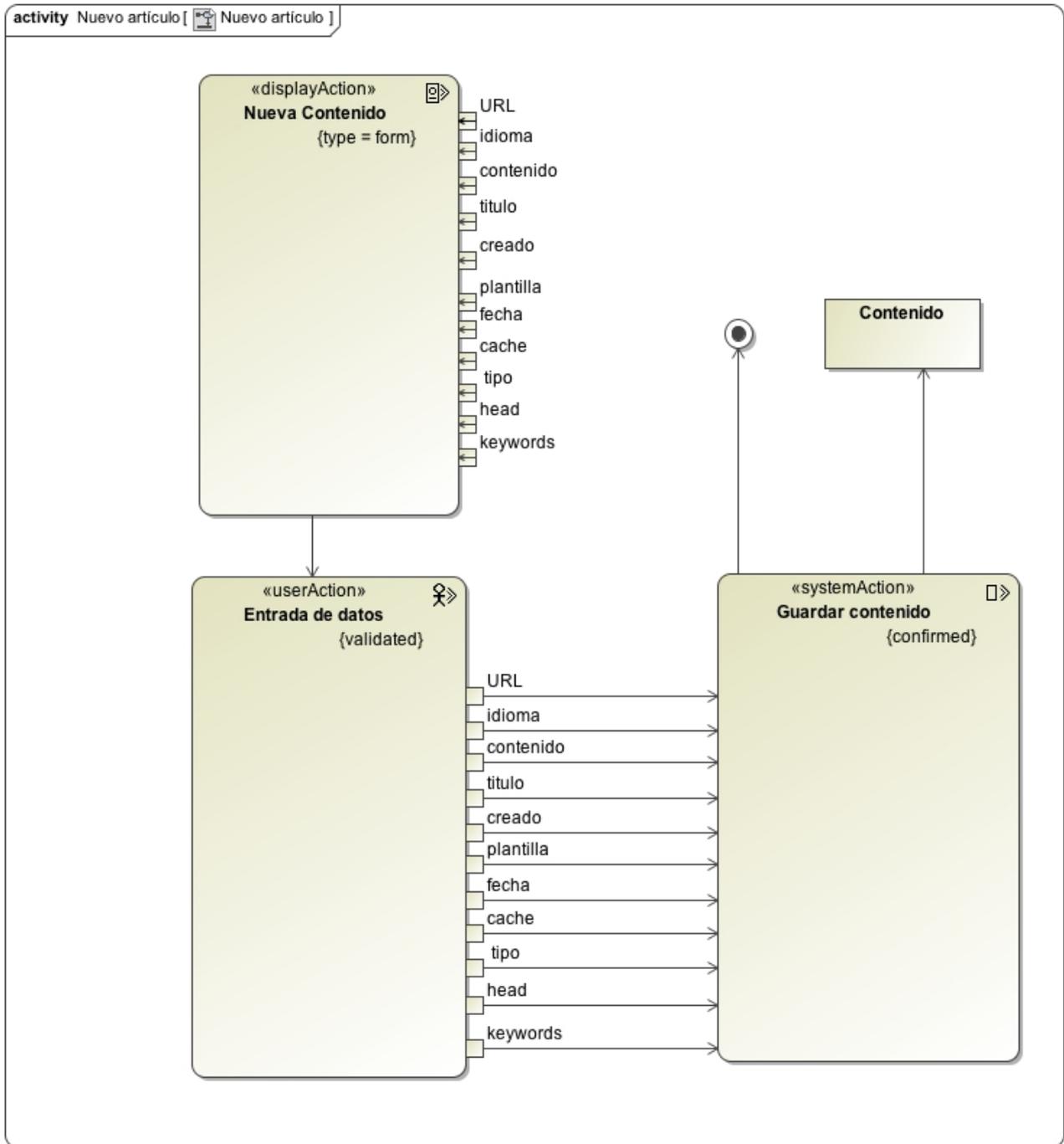
Para ejemplificar modelamos varias actividades. Primero, una actividad para el caso de uso "CrearPlantilla". El mismo muestra un formulario que permite al usuario entrar su nombre, título y demás inputs. Los datos deben ser validados durante la entrada de datos. El formulario completado por el usuario es finalmente salvado en la base de datos de la aplicación.

### 2.4.3.7 Diagrama de actividad: Nuevo artículo



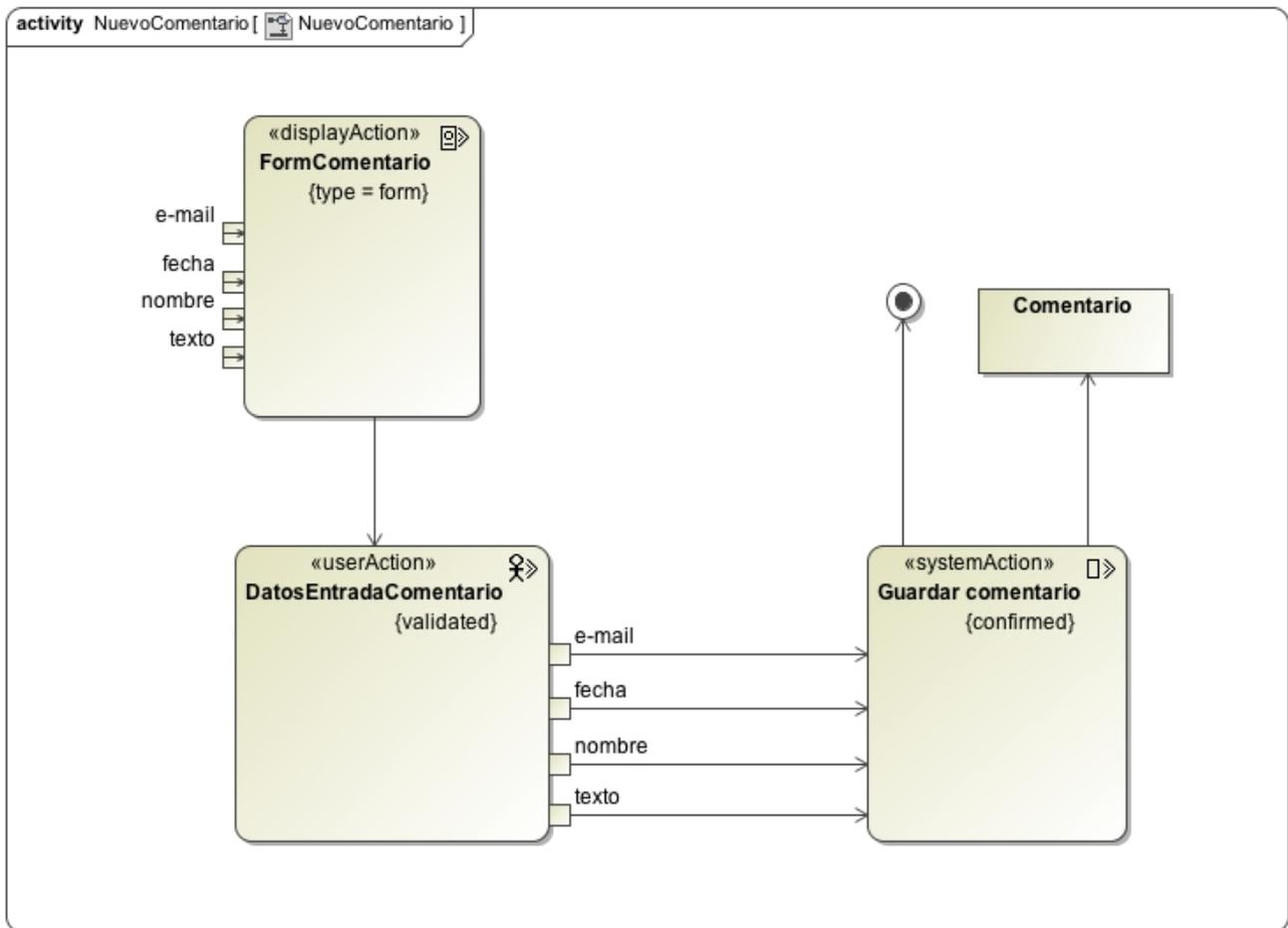
**Ilustración 14:** Diagrama de actividad: Nuevo artículo

### 2.4.3.8 Diagrama de actividad: Nuevo contenido/pagina



**Ilustración 15:** Diagrama de actividad: Nuevo contenido

### 2.4.3.9 Diagrama de actividad: Nuevo comentario



**Ilustración 16:** Diagrama de actividad: Nuevo comentario



## 2.4.4 MODELO DE COMPORTAMIENTO: DISEÑO DINÁMICO

### 2.4.4.1 DIAGRAMAS DE SECUENCIA DEL SISTEMA (DSS)

En los siguiente diagramas de secuencia, solo se documenta a modo de ejemplo y por no extender innecesariamente la documentación, las operaciones insertar, modificar y borrar correspondientes las entidades, pues gracias al diseño siguiendo el patrón, los diagramas son iguales para cualquier combinación de operaciones y entidades.

#### DIAGRAMA DE SECUENCIA INSERTAR

En el siguiente diagrama se puede ver una secuencia de acciones que se intercambian, por una parte el administrador y el controlador, y por la otra el controlador y el modelo.

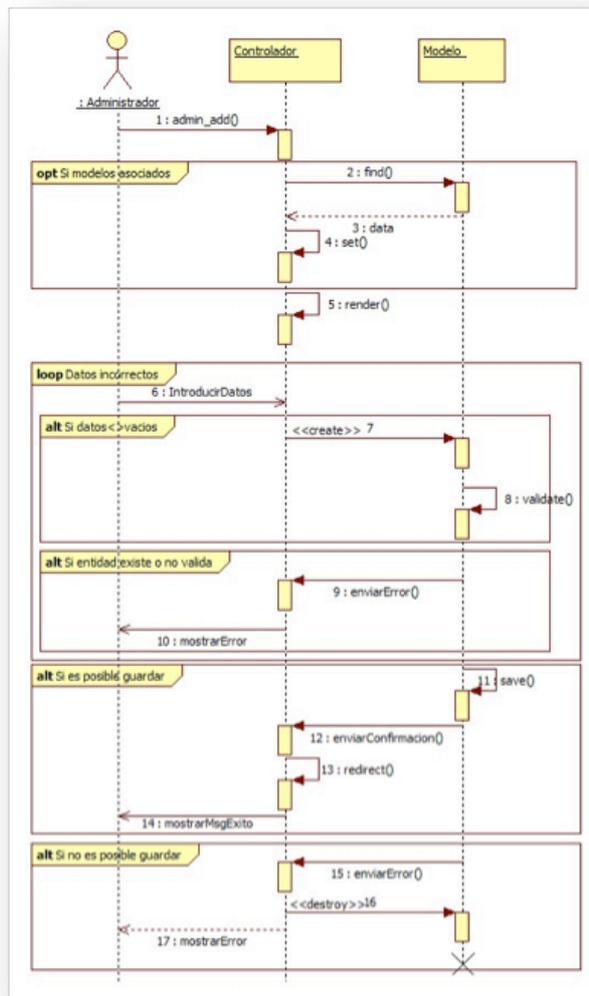
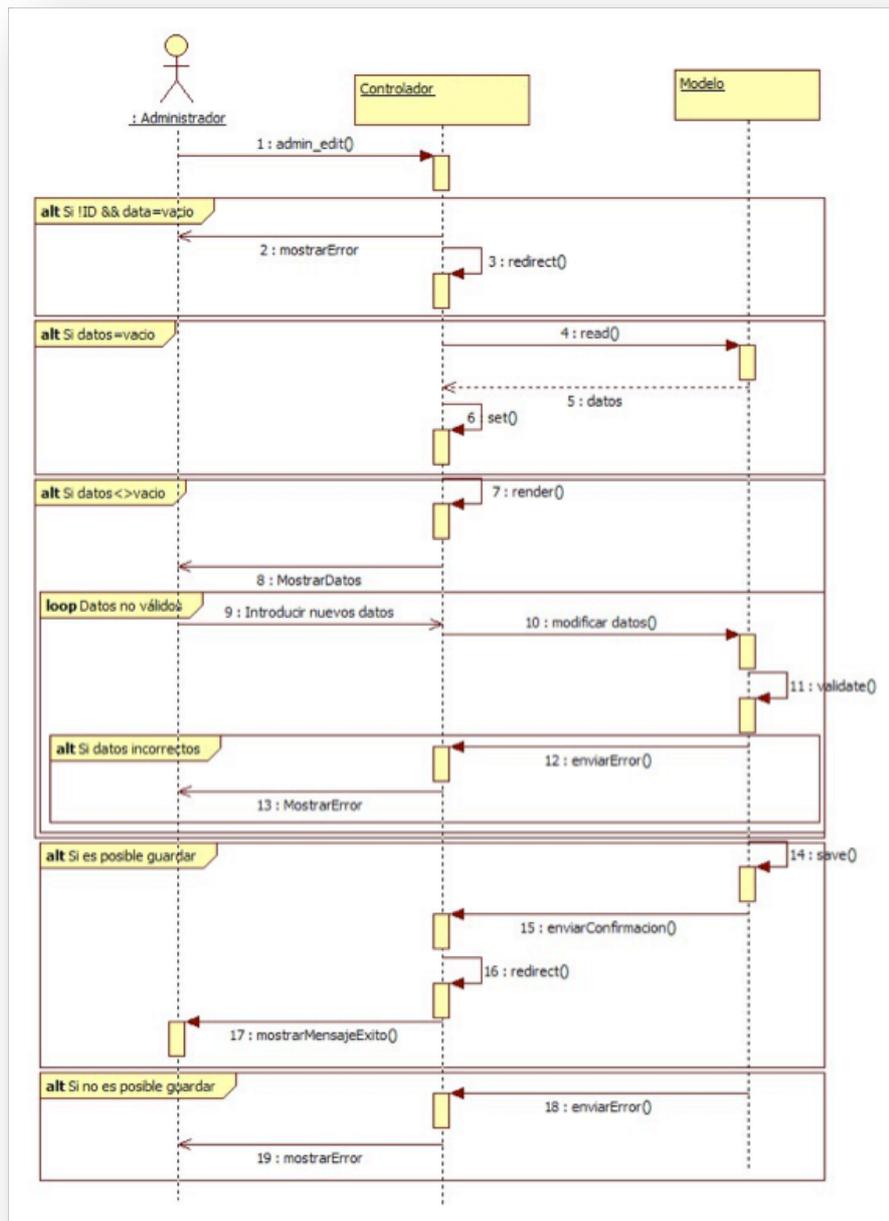


Ilustración 17: Diagrama de secuencia de insertar entidad

## DIAGRAMA DE SECUENCIA MODIFICAR

En el siguiente diagrama se puede ver la secuencia de acciones que se intercambian ,por una parte el administrador y el controlador, y por la otra el controlador y el modelo.

Este diagrama ilustra la forma de modificar cualquier tipo de fuente información existente en el sistema. Esta funcionalidad vale igual para cualquier entidad que pertenezca a la fuente de información que gestionamos en la aplicación.

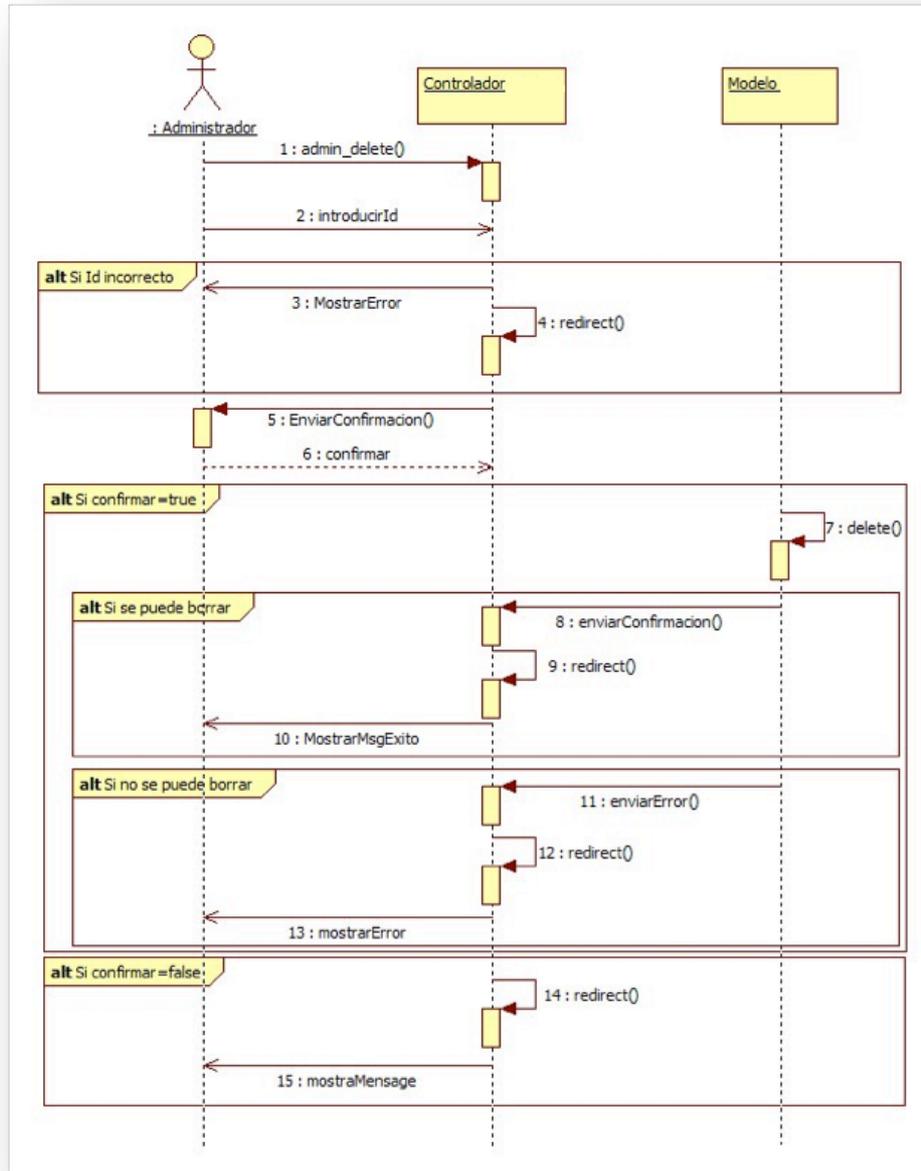


**Ilustración 18:** Diagrama de secuencia de modificar una entidad

## DIAGRAMA DE SECUENCIA BORRAR ENTIDAD

En el siguiente diagrama se puede ver una secuencia de acciones que se intercambian por una parte el administrador y el controlador, y por la otra el controlador y el modelo.

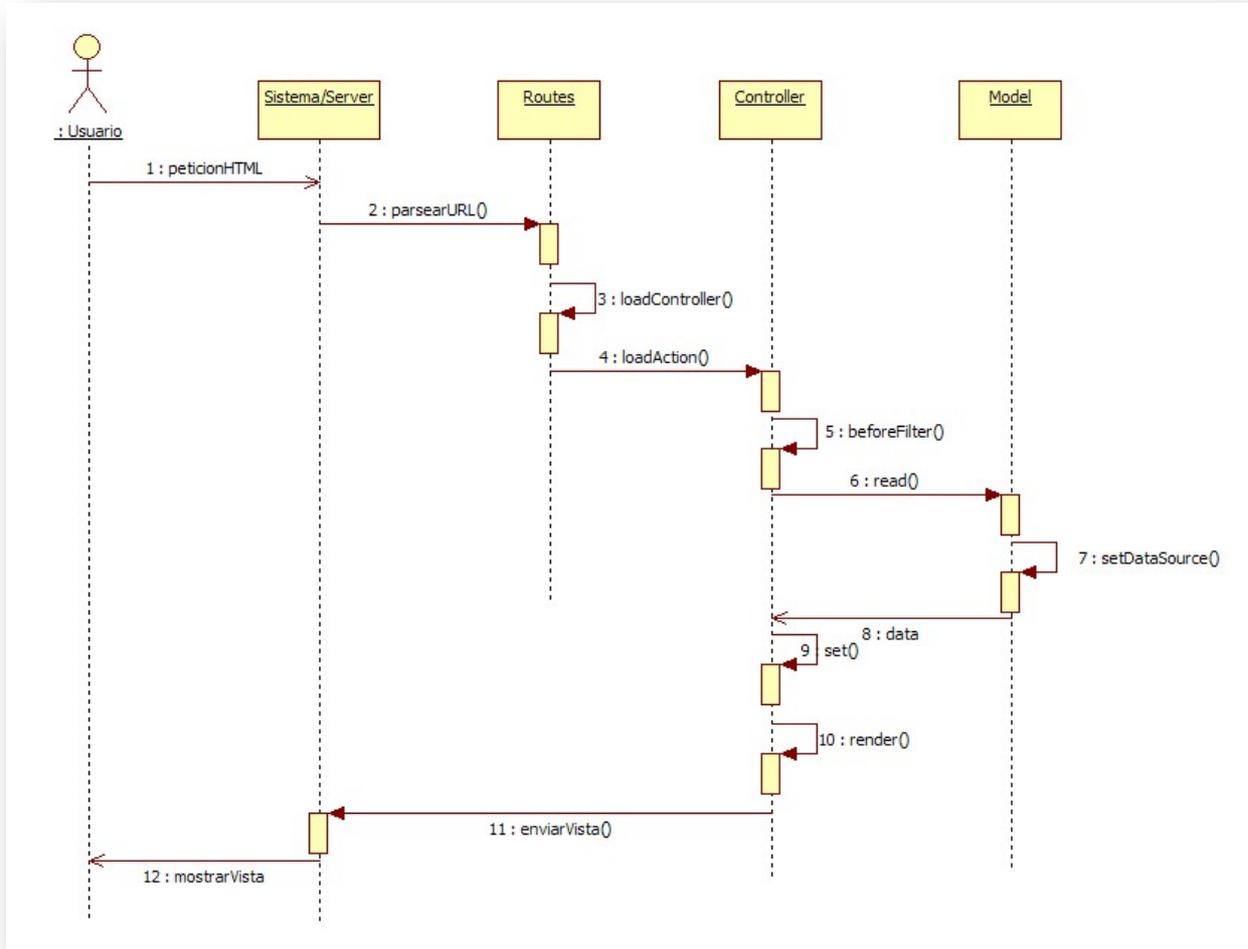
Este diagrama ilustra la forma de borrar cualquier tipo de fuente información existente en el sistema.



**Ilustración 19:** Diagrama de secuencia de borrar una entidad

## DIAGRAMA DE SECUENCIA DE UNA PETICIÓN HTML

En el siguiente diagrama se muestra la secuencia de acciones que se intercambian los distintos objetos cuando un usuario realiza una petición al servidor.

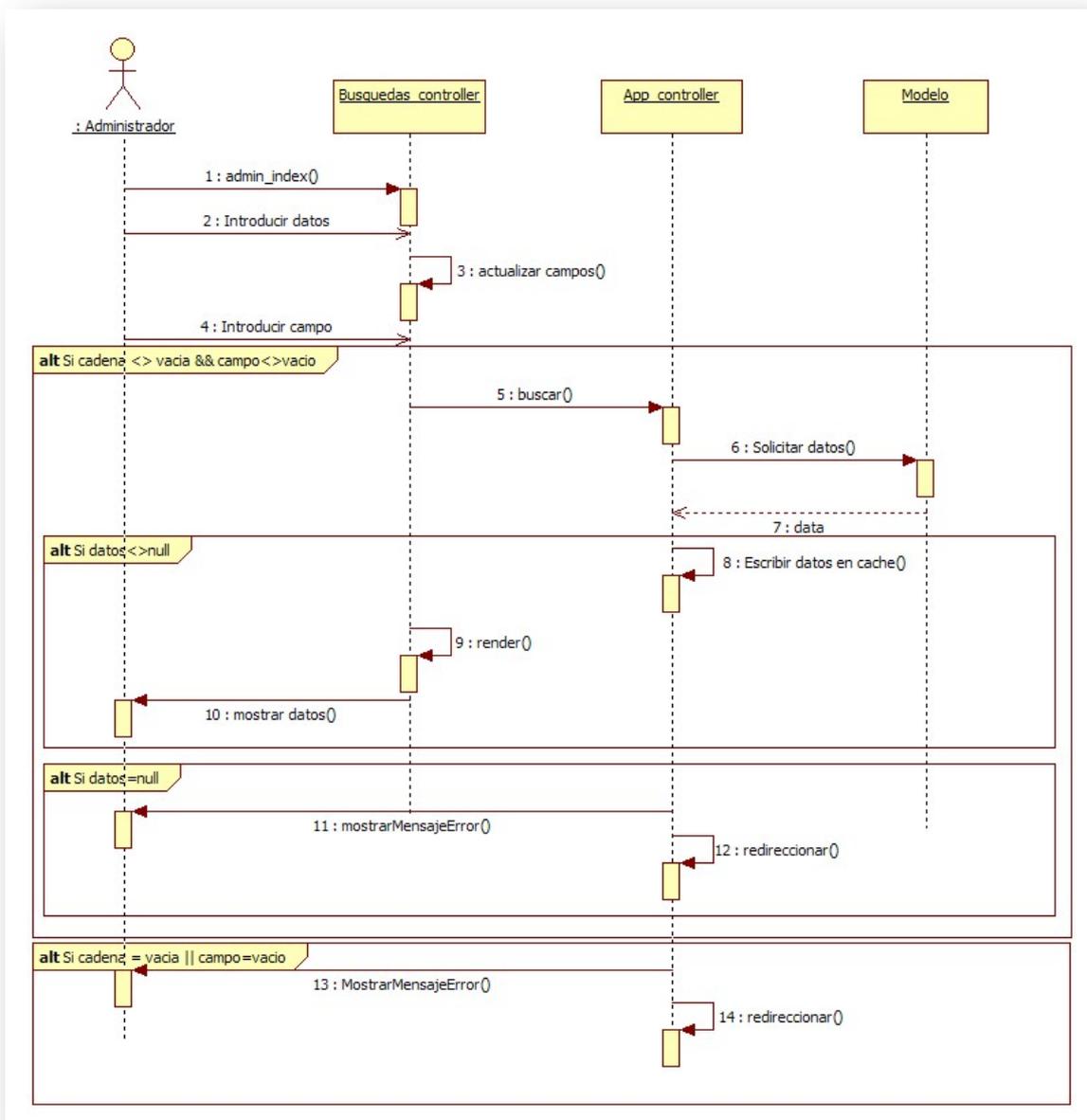


**Ilustración 20:** Diagrama de secuencia de una petición HTML

## DIAGRAMA DE SECUENCIA REALIZAR BUSQUEDA

En el siguiente diagrama se muestra la secuencia de acciones que se intercambian los distintos objetos del sistema cuando se realiza un operación de búsqueda.

La búsqueda puede ser realizada genéricamente sobre cualquier tabla que contenga información y se podrá seleccionar el campo sobre el que realizar la búsqueda de la cadena.



**Ilustración 21:** Diagrama de secuencia de una petición de búsqueda

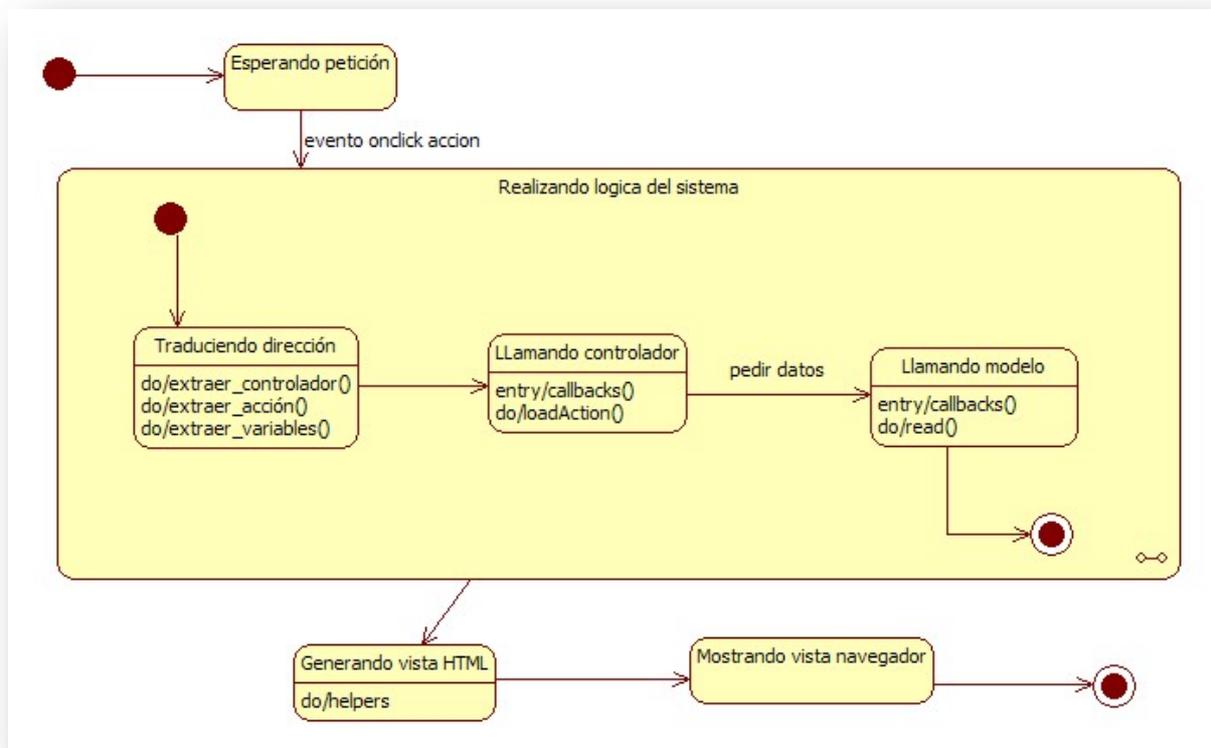


## DIAGRAMAS DE ESTADOS DEL SISTEMA (DES)

En los siguientes diagramas se ilustran los estados por los que pasa un objeto cuando se produce algún evento externo el cual modifique el flujo normal del sistema. En respuesta a estos eventos se realizan una serie de acciones. También ilustran que eventos cambian el estado del sistema.

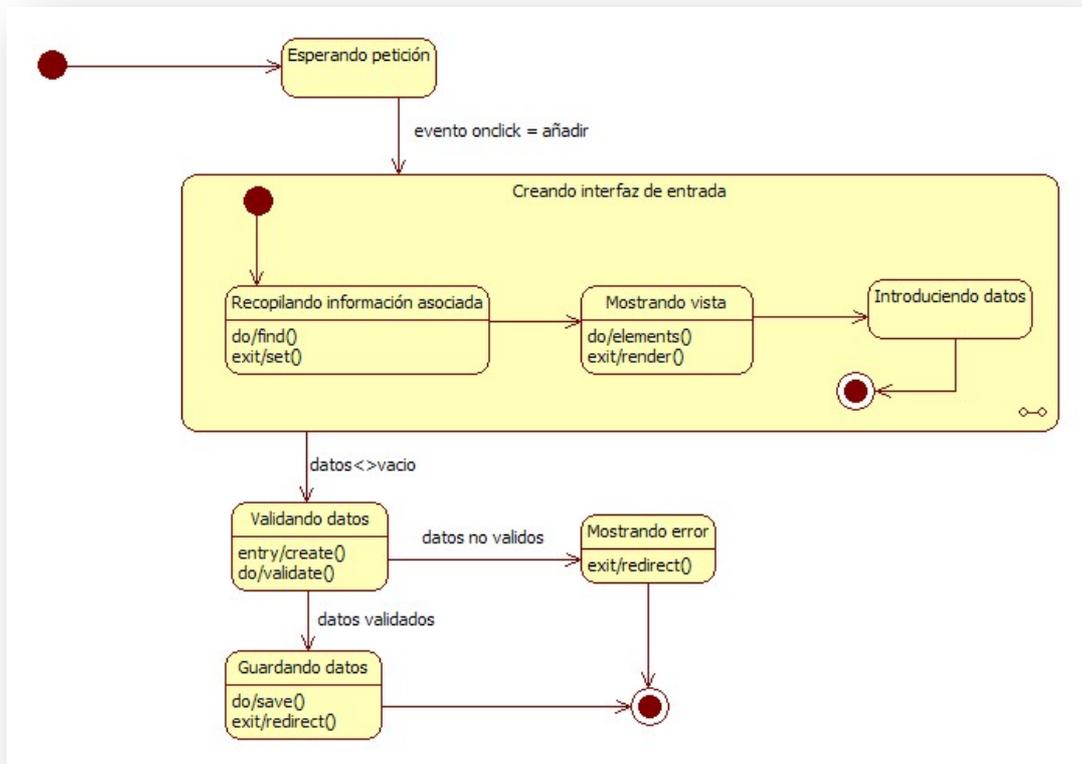
En los diagramas de estados solo se representan los relevantes de cara al modelo de negocio. Los incluidos no son relevantes de cara al proceso de negocio,

Diagrama de estados de una petición a un objeto: el comportamiento que realiza una entidad puede abarcar a las demás entidades, ya que se comportan de manera similar. El siguiente diagrama muestra las secuencias de estados por las que atraviesa el sistema cuando se recibe una petición que necesita utilizar un objeto del sistema para manipular la información asociada en la entidad.



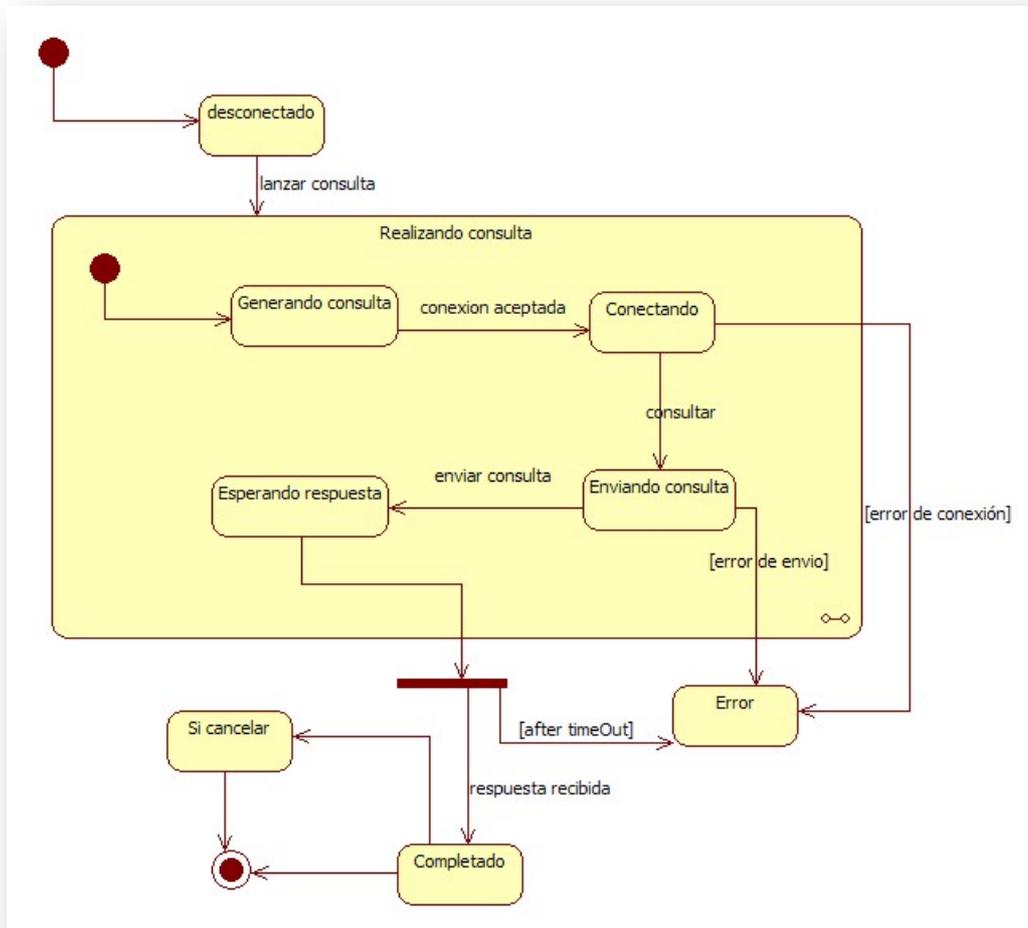
**Ilustración 22:** Diagrama de estados de una petición a un objeto del modelo

El diagrama de estados esta enfocado hacia un evento que produzca cambios, ya sean al crear nuevos objetos o modificar los existentes, y así comprobar los estados existentes y eventos que producen que exista un cambio de un estado a otro.



**Ilustración 23:** Diagrama de estados insertar/modificar entidad

Diagrama de estado de una consulta: el siguiente diagrama muestra la secuencia de estados que se producen al realizar, desde el modelo de alguna entidad, una consulta a la base de datos.



**Ilustración 24:** Diagrama de estados de una petición a la base de datos



# **3. IMPLEMENTACIÓN**

---

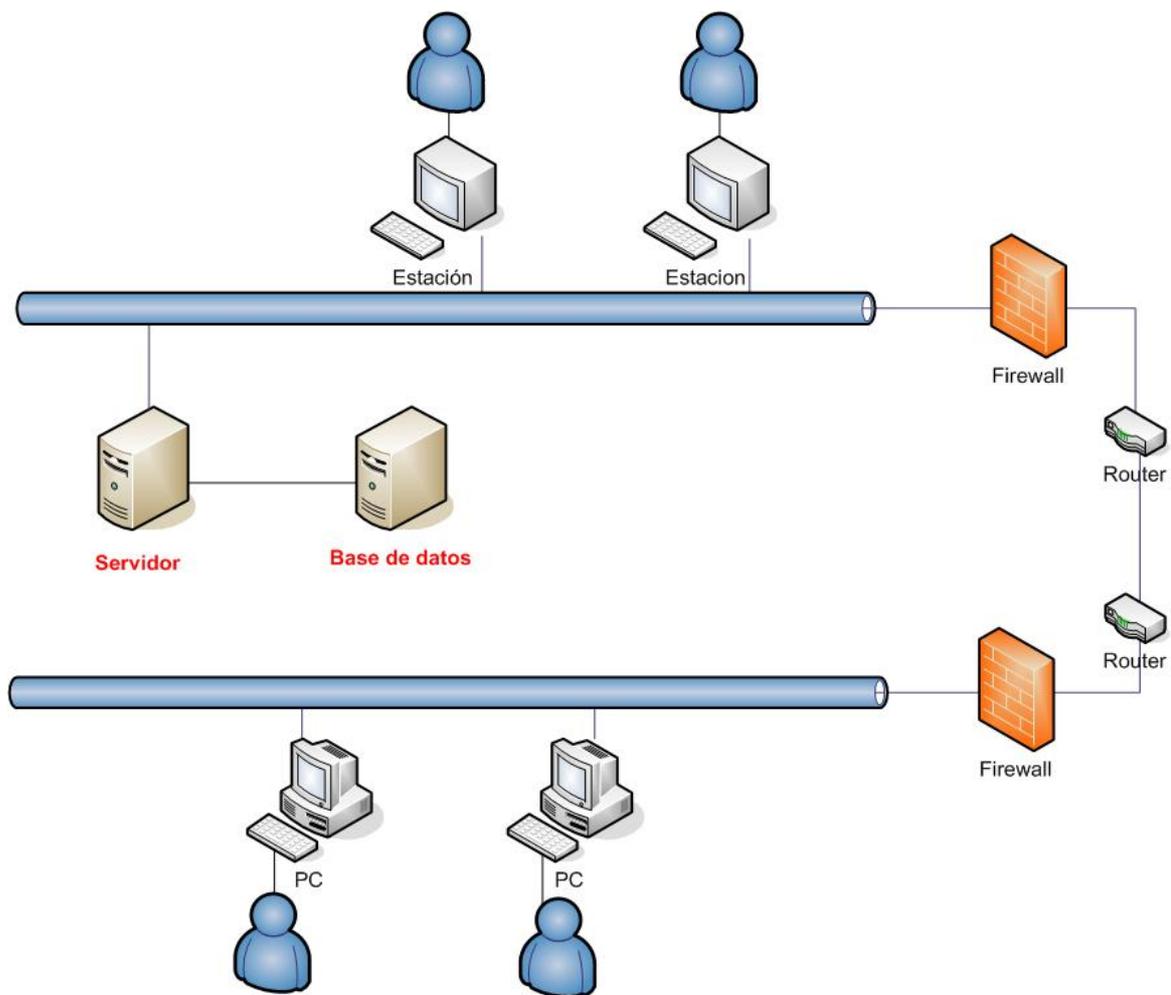


### 3.2 MODELO DE DESPLIEGUE DEL SISTEMA

En la siguiente ilustración se muestra un modelo de despliegue que nos ayuda a entender como se relaciona la implementación de nuestro sistema con los distintos componentes hardware. Modela la arquitectura en tiempo de ejecución del sistema.

La arquitectura utilizada se basa en un sistemas cliente-servidor: Los sistemas cliente-servidor son un extremo del espectro de los sistemas distribuidos y requieren tomar decisiones sobre la conectividad de red de los clientes a los servidores y sobre la distribución física de los componentes software del sistema a través de nodos.

El servidor esta alojado en la red local, no se ha utilizado un servidor externo por que se cuenta con un equipo que hace las funciones de servidor. El mismo servidor despliega los artefactos App y lib que contiene las librerias necesarias para hacer funcionar de manera correcta todo el entorno. Se puede acceder al recurso mediante las computadoras de la misma red y tambien a traves de internet mediante el uso de un modem o router que diriga la comunicación entre el servidor y la computadora del usuario.



**Ilustración 25:** Diagrama de despliegue del sistema

### 3.3 TECNOLOGÍAS UTILIZADAS PARA LA IMPLEMENTACIÓN

Las herramientas que se han utilizado para apoyar el desarrollo del sistema han sido: Java , JavaScript, Motor Ajax, librería JQuery y CSS.

El estándar Java Servlet para aplicaciones web. Proporciona clases de servlets de tu aplicación, páginas del servidor Java (JSP), archivos estáticos y archivos de datos, junto con el descriptor de implementación (el archivo web.xml) y otros archivos de configuración en una estructura de directorio WAR estándar. App Engine muestra solicitudes a través de la invocación de servlets según el descriptor de implementación.

El JVM se ejecuta en un entorno seguro de espacio aislado para aislar tu aplicación por servicio y seguridad. El espacio aislado garantiza que la aplicación solo pueda realizar acciones que no interfieran con el rendimiento ni con la escalabilidad de otras aplicaciones. Por ejemplo, una aplicación no puede generar cadenas, escribir datos en el sistema de archivos local ni establecer conexiones de red arbitrarias. Una aplicación tampoco puede utilizar JNI ni ningún otro código nativo. El JVM puede ejecutar cualquier código de bytes de Java que opere dentro de las restricciones del espacio aislado.

¿Qué es **DHTML**?

El HTML Dinámico o DHTML (*Dynamic HTML*) designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM.

- **JAVASCRIPT** es un lenguaje de programación interpretado. Se utiliza normalmente en el lado del cliente permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model*

.

- **DOM** o *Document Object Model* es un conjunto de utilidades específicamente diseñadas para manipular documentos XML. Por extensión, DOM también se puede utilizar para manipular documentos XHTML y HTML. Técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente.

Desafortunadamente, las posibilidades teóricas de DOM son mucho más avanzadas de las que se pueden utilizar en la práctica para desarrollar aplicaciones web. El motivo es que el uso de DOM siempre está limitado por las posibilidades que ofrece cada navegador. Mientras que algunos navegadores como Firefox y Safari implementan DOM de nivel 1 y 2 (y parte del 3), otros navegadores como Internet Explorer (versión 7 y anteriores) ni siquiera son capaces de ofrecer una implementación completa de DOM nivel 1.

- **AJAX**, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De

esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**AJAX** es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. **JAVASCRIPT** es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante `XMLHttpRequest`, objeto disponible en los navegadores actuales.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

- **JQUERY** es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.
- **CSS** (*Cascading Style Sheets*) u hojas de estilo en cascada es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

La idea que se esconde detrás de las hojas de estilo es separar la estructura de un documento de su presentación.

Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.

Los navegadores permiten a los usuarios especificar su propia hoja de estilo local, que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad.

El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

## ¿Qué es HTML5?

HTML5 introduce muchas características de vanguardia que permiten a los desarrolladores crear aplicaciones y sitios web con la funcionalidad, velocidad, rendimiento y experiencia de aplicaciones de escritorio. Pero a diferencia de las aplicaciones de escritorio, aplicaciones basadas en la plataforma web puede llegar a un público mucho más amplio con una gama más amplia de dispositivos. HTML5 acelera el ritmo de la innovación y le permite rodar sin problemas hacia fuera su último trabajo a todos los usuarios al mismo tiempo.

HTML5 no es una sola cosa o una tecnología monolítica. Se trata de un conjunto de características, las tecnologías y las API que trae el poder del escritorio y la vitalidad de la experiencia multimedia a la web, mientras que la amplificación de los puntos fuertes de la web núcleo de interactividad y conectividad.

HTML5 incluye la quinta revisión del lenguaje de marcado HTML, CSS3, y una serie de APIs JavaScript. Juntas, estas tecnologías permiten crear aplicaciones complejas que antes podían ser creados sólo para plataformas de escritorio.

La versión anterior y más usada de HTML, HTML4, carece de características necesarias para la creación de aplicaciones modernas basadas en un navegador. El uso fuerte de Javascript ha ayudado a mejorar esto, gracias a frameworks como jQuery, jQuery UI, Sproutcore, entre otros.

Flash en especial ha sido usado en reemplazo de HTML para desarrollar web apps que superaran las habilidades de un navegador: Audio, video, webcams, micrófonos, datos binarios, animaciones vectoriales, componentes de interfaz complejos, entre muchas otras cosas. Ahora HTML5 es capaz de hacer esto sin necesidad de plugins y con una gran compatibilidad entre navegadores.

HTML4 y HTML5 son 100% compatibles entre sí. Todo el código que tienes en HTML normal seguirá funcionando sin problemas en HTML5.

Internet Explorer, claro, es también la razón de la poca adopción de HTML5. Sin embargo, hay formas de que las etiquetas semánticas de HTML5 y atributos de CSS3 funcionen en IE.

## Nuevas capacidades de JavaScript

### Web Storage

Una cookie es la forma más casposa de guardar información en el lado del cliente. También es la única forma. Las cookies no pueden guardar más de 4KB por cookie, 100KB por dominio. Muy poco. Pero al mismo tiempo, todo el contenido de las cookies va pegado a cualquier petición HTTP que hagas al servidor. Lo que significa que por cada vez que el usuario recarga la página o baja una imagen, tiene que subir los KB que pesan todas las cookies que le hemos dado.

Web Storage soluciona este problema. Son variables que puedes guardar en el disco del usuario, con soporte en todos los navegadores (incluyendo IE8), puedes guardar hasta 5MB y no sólo texto. Cualquier tipo de datos cabe en un Storage.

### Web Workers

El problema de JavaScript es que solo puede hacer una cosa al mismo tiempo. Gran parte de la razón por la que Web falló y las web apps son simples es porque la multitarea es imposible. Web Workers soluciona eso. Web Workers permite tener multiples archivo .js corriendo en paralelo en una misma página. Haciendo tareas complejas más veloces gracias al multithreading.

### Web Sockets

Igual que XMLSockets en Actionscript, Web Sockets permite hacer aplicaciones multiusuario en tiempo real, como juegos, chats, notificaciones, etc. Si el navegador no tiene soporte de Web Sockets, es posible usar implementaciones multiuser en Javascript como PubSubHubBub.

### Arrastrar y soltar

En Gmail, crear un email e intenta arrastrar un archivo del explorador de archivos al mail. Verás que es posible adjuntarlo con sólo arrastrarlo. El gesto de arrastrar y soltar ahora es posible gracias a HTML5. Puedes traer trozos de datos o archivos enteros.

### Fuera de línea

"Web" y "offline" son dos términos que muchas personas no se asocian entre sí. Pero pronto lo harán, con HTML5 APIs que permiten crear aplicaciones que funcionan incluso cuando no estés conectado. Caché de aplicaciones, localStorage, sessionStorage, IndexedDB, sistema de archivos y eventos en línea y fuera de línea permiten que las aplicaciones "amphibiously" trabajar con o sin conexión a Internet

### 3 PRUEBAS

La elaboración de una batería de pruebas para una aplicación web es indispensable para que la aplicación presente un buen rendimiento. Se van a realizar dos tipos de pruebas: una en el lado del cliente donde se ejecuta una parte de la aplicación y la otra en el lado del servidor.

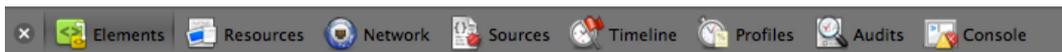
Una prueba se enfoca sobre la lógica interna del programa y sobre las funciones externas. Con estas pruebas se desvelan posibles errores cometidos en la elaboración del sistema. Un buen proceso de prueba es aquel que tiene una alta probabilidad de encontrar un error no descubierto hasta entonces.

Los objetivos de la fase de pruebas son los siguientes:

- Encontrar y documentar defectos que pueda tener la aplicación web.
- Validar que funciona para lo que ha sido diseñado.
- Verificar requisitos que debe de cumplir el software.
- Validar interacción e integración de los componentes.
- Asegurar que los defectos encontrados se han corregido antes de la entrega al cliente.

En cuanto a la realización de las pruebas se basan en testeos a diferentes niveles, se necesita probar si cada unidad funciona correctamente, luego es necesario probar si los distintos módulos encajan entre sí y por último pruebas al proyecto web globales.

La herramienta utilizada para realizar las pruebas en la parte cliente de la aplicación ha sido las herramientas de desarrollo de Google para su navegador Chrome. Es un plugin que se instala como complemento en el navegador y dispone de las siguientes características.



**Ilustración 26:** Ventana de desarrollo Chrome

En la primera pestaña se encuentran los elementos, es decir, podemos visualizar toda la estructura HTML, junto con sus estilos asociados, pudiendo diferenciar las partes que componen la estructura de la página.

La siguiente opción es la de *resources* que son los recursos que esta utilizando la página, aquí englobamos todos los ficheros como imágenes, .html, .css, .js, etc. Aquí podemos descargarnos los ficheros a local, así como las últimas versiones modificadas.

Después encontramos la opción de *network* donde podemos ver en tiempo real los elementos que se van descargando del servidor, así como sus tiempo de carga y espera. También podemos ver lo que pesa cada fichero para intentar que los ficheros más pesados se carguen al final del resto de los elementos.

Se puede acceder a los scripts de la página para poder insertar *breakpoints* y así ver el estado de la ejecución en un momento concreto para validar su correcto funcionamiento.

Con el *timeline* se pueden realizar grabaciones en tiempo real de los elementos que se van cargando a nivel de DOM, así como los nodos que componen la página, peticiones con AJAX, en definitiva el monitoreo de todas las interacciones con el servidor.

Los perfiles sirven para medir la eficiencia de los ficheros JavaScript de la página, mide tiempos de ejecución, porcentajes de uso, funciones utilizadas, etc.

En la opción *Audit* podemos realizar una auditoria de la página que nos mostrará la utilización de red y el rendimiento de la página. Nos aconsejara cambiar las reglas que no sean eficientes y eliminar las que no se utilizen. Nos mostrará los detalles como el tamaño habilitado para las cookies, contenido cacheable, etc.

En la opción de *Console* podemos interactuar con la página y ejecutar funciones para ver el funcionamiento de los elementos.

Para depurar la parte del servidor y verificar que todo funciona correctamente, se ha realizado un testeo en todas las funciones disponibles, dando diferentes datos de entrada y verificando que el resultado era el apropiado.

Se han insertado registros para cada una de las entidades, modificando los valores de entrada, y así verificar el correcto funcionamiento de la aplicación, desde que el usuario introduce los parámetros de entrada, hasta que llegan a la base de datos.

Al ejecutar la aplicación desde un plugin del eclipse se puede testear el lado del servidor utilizando la perspectiva de *debug* y ver los logs, añadir puntos de ruptura, observar variables, ver el código línea por línea, como si de una aplicación normal se tratara.

Se han añadido puntos de interrupción en la creación de los objetos, en la introducción de registros en la base de datos, en los datos devueltos por una consulta o en la destrucción de objetos, para ver el resultado de las operaciones en puntos críticos de la ejecución.

Se puede visualizar la pila para ver que partes del programa se están ejecutando en un momento determinado y como se relacionan entre si

## 4 RESUMEN

<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>TIPOS DE OBJETOS</b>	Object
	Controller
	App_controller
	Servicio_controller
	Decoraciones_controller
	Familias_controller
	Categorias_controller
	Piezas_controller
	Overloadable
	Model
	App_model
	Servicio
	Decoracion
	Familia
	Categoria
	Pieza
	Helper
	Text_helper
	Form_helper
	Html_helper
	Helper
	MeioUpload
	ModelBehaviors
	Imagen
	Catalogo
App_model	
<b>ASOCIACIONES</b>	Incluye
	Ofrece
	Pertenece
	Incluida
	Utiliza
<b>DIAGRAMAS DE SECUENCIA</b>	Diagrama de secuencia insertar pieza
	Diagrama de secuencia modificar decoracion
	Diagrama de secuencia consultar categoría
	Diagrama de secuencia borrar familia
	Diagrama de secuencia listar servicios
	Diagrama de secuencia de una petición HTML
	Diagrama de secuencia realizar búsqueda
<b>DIAGRAMAS DE ESTADOS</b>	Diagrama de estados petición a objeto
	Diagrama de estados insertar/modificar entidad
	Diagrama de estados consultar base de datos

## 5 Índice de Tablas

TABLA 1: ATRIBUTOS DE LA ENTIDAD PLANTILLA .....	81
TABLA 2: ATRIBUTOS DE LA ENTIDAD CONTENIDO.....	82
TABLA 3: ATRIBUTOS DE LA ENTIDAD CONDICIÓN.....	82
TABLA 4: ATRIBUTOS DE LA ENTIDAD USUARIOS .....	83
TABLA 5: ATRIBUTOS DE LA ENTIDAD USUARIO .....	83
TABLA 6: ATRIBUTOS DE LA ENTIDAD GRUPO .....	83
TABLA 7: DOMINIOS DE LOS TIPOS DE ATRIBUTOS .....	84
TABLA 8: TIPO DE OBJETO PLANTILLA .....	93
TABLA 9: ATRIBUTO AUTOGUARDADO DEL OBJETO PLANTILLA .....	93
TABLA 10: ATRIBUTO CUERPO DEL OBJETO PLANTILLA .....	93
TABLA 11: ATRIBUTO NOMBRE DEL OBJETO PLANTILLA .....	93
TABLA 12: ATRIBUTO TITULO DEL OBJETO PLANTILLA.....	93
TABLA 13: TIPO DE OBJETO RECURSO .....	94
TABLA 14: ATRIBUTO FICHERO DEL OBJETO RECURSO .....	94
TABLA 15: ATRIBUTO MIME DEL OBJETO RECURSO.....	94
TABLA 16: ATRIBUTO NOMBRE DEL OBJETO RECURSO .....	94
TABLA 17: ATRIBUTO SUBCARPETA DEL OBJETO RECURSO.....	94
TABLA 18: ATRIBUTO TAMANO DEL OBJETO RECURSO .....	94
TABLA 19: TIPO DE OBJETO COMENTARIO .....	95
TABLA 20: ATRIBUTO EMAIL DEL OBJETO COMENTARIO.....	95
TABLA 21: ATRIBUTO FECHA DEL OBJETO COMENTARIO .....	95
TABLA 22: ATRIBUTO NOMBRE DEL OBJETO COMENTARIO.....	95
TABLA 23: ATRIBUTO TEXTO DEL OBJETO COMENTARIO .....	95
TABLA 24: TIPO DE OBJETO USUARIO .....	96
TABLA 25: ATRIBUTO DESCRIPCIÓN DEL OBJETO USUARIO .....	96
TABLA 26: ATRIBUTO FECHA DEL OBJETO USUARIO .....	96
TABLA 27: ATRIBUTO NOMBRE DEL OBJETO USUARIO .....	96
TABLA 28: ATRIBUTO TEXTO DEL OBJETO USUARIO.....	96
TABLA 29: TIPO DE OBJETO PÁGINA.....	97
TABLA 30: ATRIBUTO AUTOGUARDADO DEL OBJETO PÁGINA.....	97
TABLA 31: ATRIBUTO AUTOR DEL OBJETO PÁGINA .....	97
TABLA 32: ATRIBUTO CONTENIDO DEL OBJETO PÁGINA .....	97
TABLA 33: ATRIBUTO CREADO DEL OBJETO PÁGINA .....	97
TABLA 34: ATRIBUTO TAG DEL OBJETO PÁGINA .....	97
TABLA 35: ATRIBUTO TÍTULO DEL OBJETO PÁGINA .....	98

## 6 Índice de Figuras

ILUSTRACIÓN 1: MODELO DE LA BASE DE DATOS .....	81
<b>ILUSTRACIÓN 2: ESQUEMA GENERAL DEL PATRÓN MODELO-VISTA-CONTROLADOR.....</b>	<b>87</b>
<b>ILUSTRACIÓN 3: DIAGRAMA DE CLASES .....</b>	<b>92</b>
<b>ILUSTRACIÓN 4: DIAGRAMA DE ACTIVIDAD: NUEVO ARTÍCULO.....</b>	<b>99</b>
<b>ILUSTRACIÓN 5: DIAGRAMA DE ACTIVIDAD: NUEVO CONTENIDO.....</b>	<b>100</b>
<b>ILUSTRACIÓN 6: DIAGRAMA DE ACTIVIDAD: NUEVO COMENTARIO.....</b>	<b>101</b>
<b>ILUSTRACIÓN 7: DIAGRAMA DE SECUENCIA INSERTAR ENTIDAD .....</b>	<b>¡ERROR!MARCADOR NO DEFINIDO.</b>
<b>ILUSTRACIÓN 8: DIAGRAMA DE SECUENCIA DE MODIFICAR UNA ENTIDAD.....</b>	<b>104</b>
<b>ILUSTRACIÓN 9: DIAGRAMA DE SECUENCIA DE BORRAR UNA ENTIDAD .....</b>	<b>105</b>
<b>ILUSTRACIÓN 10: DIAGRAMA DE SECUENCIA DE UNA PETICIÓN HTML.....</b>	<b>106</b>
<b>ILUSTRACIÓN 11: DIAGRAMA DE SECUENCIA DE UNA PETICIÓN DE BUSQUEDA .....</b>	<b>107</b>
<b>ILUSTRACIÓN 12: DIAGRAMA DE ESTADOS DE UNA PETICIÓN A UN OBJETO DEL MODELO .....</b>	<b>109</b>
<b>ILUSTRACIÓN 13: DIAGRAMA DE ESTADOS INSERTAR/MODIFICAR ENTIDAD.....</b>	<b>110</b>
<b>ILUSTRACIÓN 14: DIAGRAMA DE ESTADOS DE UNA PETICIÓN A LA BASE DE DATOS.....</b>	<b>111</b>
<b>ILUSTRACIÓN 15: DIAGRAMA DE DESPLIEGUE DEL SISTEMA.....</b>	<b>115</b>
<b>ILUSTRACIÓN 17: VENTANA DE DESARROLLO CHROME.....</b>	<b>119</b>



**BLOQUE III**

---

**MANUAL DE USUARIO**

---



## 1. ÍNDICE

1. ÍNDICE .....	127
2. MANUAL DE INSTALACIÓN .....	128
2.1 INSTALAR NAVEGADOR: .....	128
2.2 INSTALAR ECLIPSE:.....	131
2.3 INSTALACIÓN DE GOOGLE APP ENGINE .....	131
2.4 INSTALACIÓN DE MAVEN .....	132
2.5 CREAR NUEVO PROYECTO .....	133
2.6 MANUAL DE USUARIO.....	137
3. ÍNDICE DE ILUSTRACIONES .....	151

## 2. MANUAL DE INSTALACIÓN

Para la ejecución de la aplicación se necesitan los siguientes programas.

Eclipse – Nos proporciona el entorno para ejecutar la aplicación.

Google App Engine Eclipse - se trata de un plugin realizado por Google para que podamos desarrollar aplicaciones con App Engine en Eclipse. Proporciona un servidor con las mismas características que tendríamos si desplegáramos la aplicación en su nube.

Maven – Es otro plugin para eclipse que nos permite generar los archivos necesarios para ejecutar la aplicación.

Java JDK 6 – Librerías Java.

Navegador Google Chrome/ Mozilla Firefox para poder visualizar la aplicación.

Código fuente – Código necesario para generar los archivos que componen la aplicación web.

### 2.1 INSTALAR NAVEGADOR:

Un navegador es necesario para poder interactuar con las interfaces que se han realizado, el navegador es el que interpreta el código HTML y genera las vistas, también cuentan con paquetes que pueden ejecutar código JavaScript, interpretar PHP, representar CSS, etc.

En este manual se ha escogido *Google Chrome* como navegador preferido. Se expresan las razones, como adquirir una versión actual, la que se adjunta en el paquete de archivos es:

#### 3.4 Como instalar Google Chrome

Google Chrome es un navegador Web gratuito que se instala en tan sólo unos minutos. Está disponible en versiones para Windows, Mac y Linux. En este artículo te guiaremos por todos los pasos del proceso de instalación.

#### 3.5 Como instalar Google Chrome en Windows

Para instalar Google Chrome utilizando tu cuenta de usuario, sigue los pasos descritos a continuación.

- 1 Ve a <http://www.google.es/chrome> .
- 2 En el menú desplegable situado encima del botón de descarga, comprueba el idioma del navegador.



**Ilustración 27:** Seleccionar idioma de descarga de Google Chrome Windows

- 3 Haz clic en Descargar Google Chrome. Al descargar el archivo de instalación en este sitio, te aseguras de que Google Chrome esté actualizado con las últimas funciones y parches de seguridad.



**Ilustración 28:** Seleccionar enlace de descarga

- 4 Consulta las Condiciones del servicio y haz clic en el botón Aceptar e instalar para continuar.

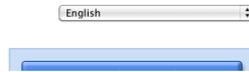


**Ilustración 29:** Aceptar condiciones Google Chrome

- 5 Tras descargar el archivo del instalador, aparece el cuadro de diálogo "Bienvenido a Google Chrome". En él se te pide que elijas el motor de búsqueda que te gustaría usar de forma predeterminada en el navegador.

### 3.6 Como instalar Google Chrome en Mac

- 3 Accede a la página <http://www.google.com/chrome/?hl=es>.



#### **Ilustración 30:** Seleccionar idioma de descarga Mac

- 4 En el menú desplegable situado encima del botón de descarga, comprueba el idioma del navegador.
- 5 Haz clic en Descargar Google Chrome.



#### **Ilustración 31:** Seleccionar enlace de descarga

- 6 Consulta las Condiciones del servicio y, a continuación, haz clic en el botón Aceptar e instalar para continuar.
- 7 Una vez que hayas descargado el archivo "Google Chrome.dmg", ábrelo.
- 8 En la ventana que se abrirá, arrastra el icono de Google Chrome hasta la carpeta "Aplicaciones". Google Chrome se instalará para todas las cuentas de usuario de tu equipo.
- 9 La primera vez que abras el navegador, el buscador te solicitará que confirmes que deseas abrir la aplicación descargada de Internet. Haz clic en Abrir para continuar.
- 10 Aparecerá el cuadro de diálogo "Bienvenido a Google Chrome" solicitándote que elijas el motor de búsqueda que te gustaría usar de forma predeterminada en el navegador.

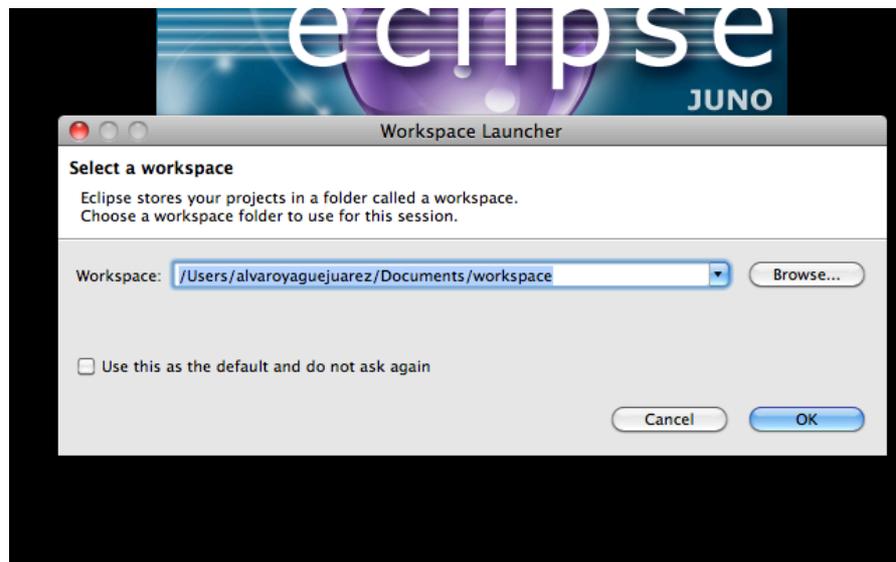
## 2.2 INSTALAR ECLIPSE:

Para la instalación de Eclipse mire en el CD proporcionado y bajo la carpeta software se encuentra un archivo comprimido llamado eclipse, descomprímalo en su escritorio.

En la carpeta donde se ha descomprimido el ZIP busque un ejecutable llamado eclipse.exe haga doble clic y ya tendrá eclipse funcionando.

Haga doble click y siga las instrucciones.

Cuando inicie el programa se le pedirá que introduzca la ruta de un directorio donde se guardarán los proyectos que genere con Eclipse. Esto es conocido como *workspace*.



**Ilustración 32:** Workspace Eclipse

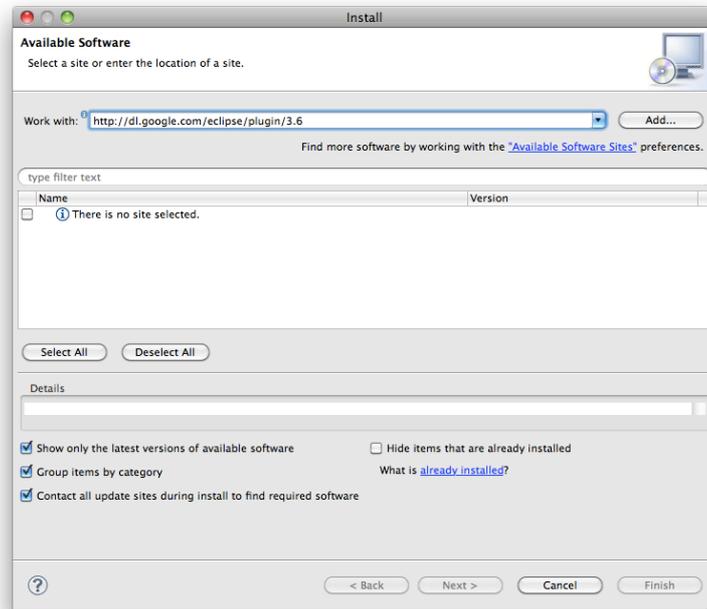
## 2.3 INSTALACIÓN DE GOOGLE APP ENGINE

1. Selecciona el menú Help > Install New Software... (Ayuda > Instalar software nuevo).
2. En el cuadro de texto Work with (Trabajar con), introduce:  
<http://dl.google.com/eclipse/plugin/3.6>

Haz clic en el botón Add... (Añadir). En el cuadro de diálogo que aparece, haz clic en OK (Aceptar). Deja el nombre en blanco; el sistema lo obtendrá del sitio actualizado.

3. Haz clic en el triángulo situado junto a "Plugin" (Complemento) y "SDKs" (SDK). Marca las casillas de verificación situadas junto a "Google Plugin for Eclipse 3.6" y "Google App Engine Java SDK" (SDK Java de Google App Engine). También puedes seleccionar "Google Web Toolkit SDK" (SDK de Google Web Toolkit) si quieres utilizar Google Web Toolkit con tus aplicaciones.
4. Haz clic en el botón Next (Siguiente). Sigue las indicaciones que aparecen para aceptar las condiciones del servicio e instalar el complemento.

- Al finalizar la instalación, aparece un mensaje de Eclipse solicitando que reinicies el sistema. Haz clic en Yes (Sí). Eclipse se reinicia y el complemento ya está instalado.



**Ilustración 33:** Instalando App Engine

## 2.4 INSTALACIÓN DE MAVEN

m2e es un plugin que nos permite integrar Maven dentro de Eclipse, de forma que podamos trabajar fácilmente con proyectos Java gestionados por Maven.

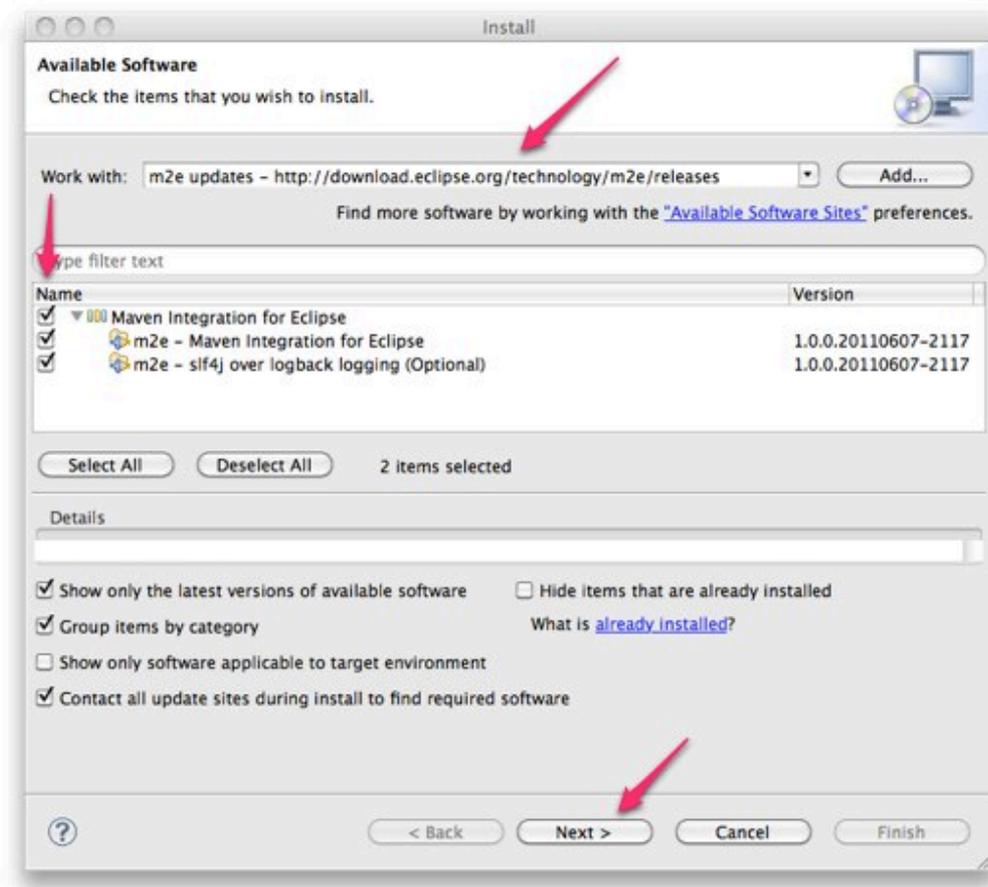
La instalación la podemos hacer por el Marketplace, o de la forma tradicional proporcionando nosotros la URL. Vamos a elegir este segundo método para tener un poco más de control sobre lo que vamos a instalar.

En la página de documentación de m2e nos indica dos URL de instalación, una para las releases oficiales, y otra para las builds estables de desarrollo. Vamos a escoger la primera ya que la segunda parece no funcionar.

Así que la URL que vamos a usar es: <http://download.eclipse.org/technology/m2e/releases>

Para hacer la instalación seleccionamos en el menú de Eclipse Help --> Install New Software...

Ahora sobre el botón Add..., y añadimos la URL.



**Ilustración 34:** Instalación de Maven

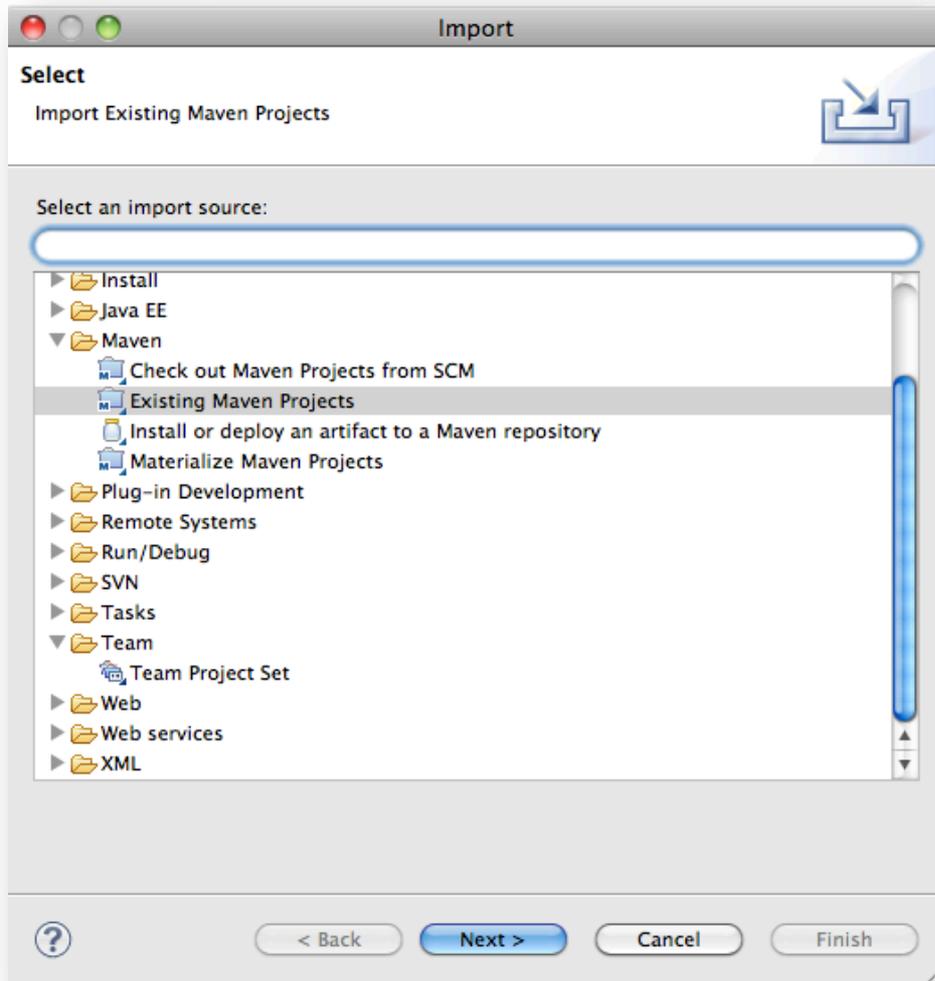
## 2.5 CREAR NUEVO PROYECTO

### PASO 1:

Antes de crear el proyecto, debes copiar los archivos que se encuentran en la carpeta `/cms/` del CD al *workspace* que hayas definido previamente.

### PASO 2:

Ejecutar Eclipse y en la pestaña de “Archivo” seleccionar la opción “Importar“. Seleccionar importar proyecto Maven existente desde el *workspace* bajo la pestaña de Maven.



**Ilustración 35:** Importar proyecto Maven

Olvidate de los errores y advertencias, antes de preocuparse por ellos tenemos que arreglar algo más.

En la pestaña de configuración hay un campo donde indica el directorio de origen. Selecciona el proyecto que acabas de copiar del CD al Workspace. Dando a siguiente deberías ver la configuración de los archivos POM.

Eclipse comenzará a importar proyectos Maven y comenzará la descarga de las dependencias necesarias definidas dentro de los POM. Esta tarea llevará un tiempo al ordenador.

Una vez terminado el proceso tendremos nuestro proyecto en el espacio de trabajo y podremos ver en la pestaña “problemas“ y podremos ver los errores y advertencias.

### PASO 3: Solucionar problemas

En primer lugar un aviso: "Problemas Java Build Path" . Para resolver esta advertencia, hay que cambiar el J2SE JRE-1.5 biblioteca de la ruta de compilación y reemplazarla con la versión 6 de Java JRE biblioteca.

Dentro del Explorador de proyectos ,haz click derecho sobre el proyecto y pincha en la opción Propiedades. En la sección Java Build Path, seleccione librerías y haga doble clic en Sistema de

Bibliotecas JRE [J2SE-1.5], cambielo por la versión 1.6. Haga clic en Finish y Eclipse actualizará la ruta de compilación y compilar el proyecto de nuevo.

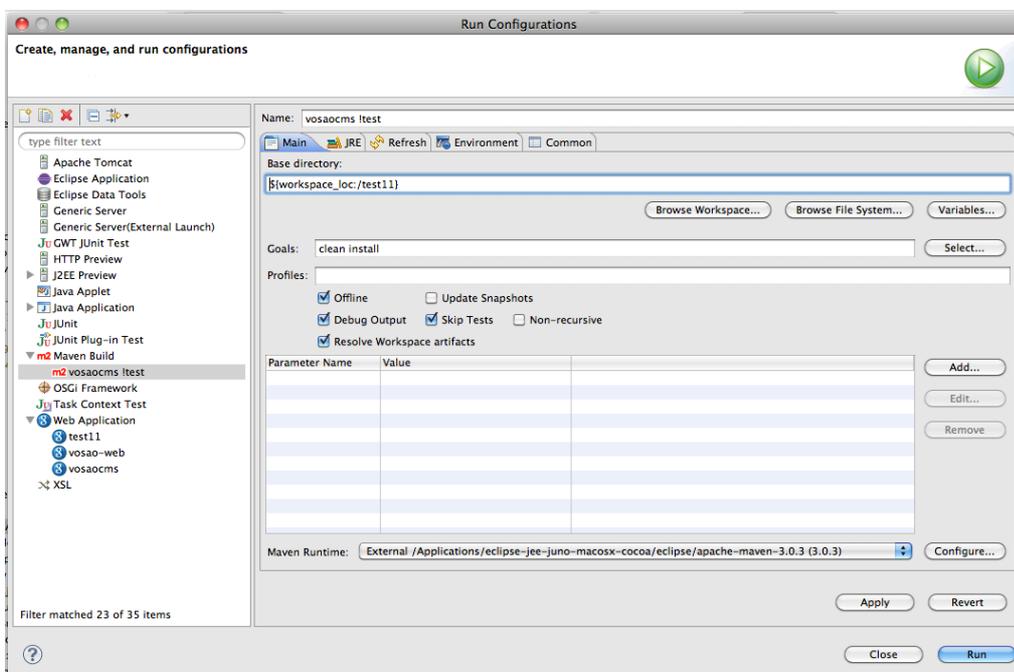
### Project Facet

Siguiente en la lista: "El nivel de compilador de Java no coincide con la versión de la *facet project* instalado de Java." . Vaya al Explorador de proyectos, abra la perspectiva Java EE, haga clic en el proyecto y seleccione Propiedades ... . Seleccione el "Project facets" en la ventana de propiedades y cambie el valor de Java a 1,6.

### PASO 4: Compilar con Maven

Cuando haya resuelto los errores podrá ejecutar el proyecto para generar el código necesario que necesite para ejecutar la aplicación bajo App Engine.

Click con el botón secundario sobre el proyecto y en "ejecutar como" seleccionar como maven install.



**Ilustración 36:** Compilar con Maven

### PASO 5: Ejecutar

Ahora ya tiene el código necesario para poder ejecutar el proyecto como una aplicación web. Seleccione ejecutar como aplicación web.

En su navegador introduzca la url "http://localhost:8888". Si es la primera vez que accede al sitio, se encontrará con una página que le ayudará a configurar el cms.

## CARACTERÍSTICAS

Contenido: Page tree:

Una página en Vosao CMS es un fragmento de contenido. El contenido está enlazado a una plantilla para crear una página web completa que pueda ser interpretada por el navegador.

El CMS organiza las páginas usando una jerarquía padre / hijo. En el área de trabajo, la jerarquía se representa como un esquema o control de árbol. Inicialmente, el árbol está contraído.

Para abrir cada rama del árbol, seleccione el botón [+] control.

Para agregar una carpeta de la vista de árbol, pulse el botón [+] control.

Para abrir una página para la edición, seleccione el enlace de nombre de la página.

El área de la página organiza acciones en cinco pestañas: Página, contenido, páginas hijas, comentarios y de seguridad.

Un URL amigables es definido por una URL de la página que es única para un sitio web. Una URL amigable consta del nombre de la página principal y de una subpágina.

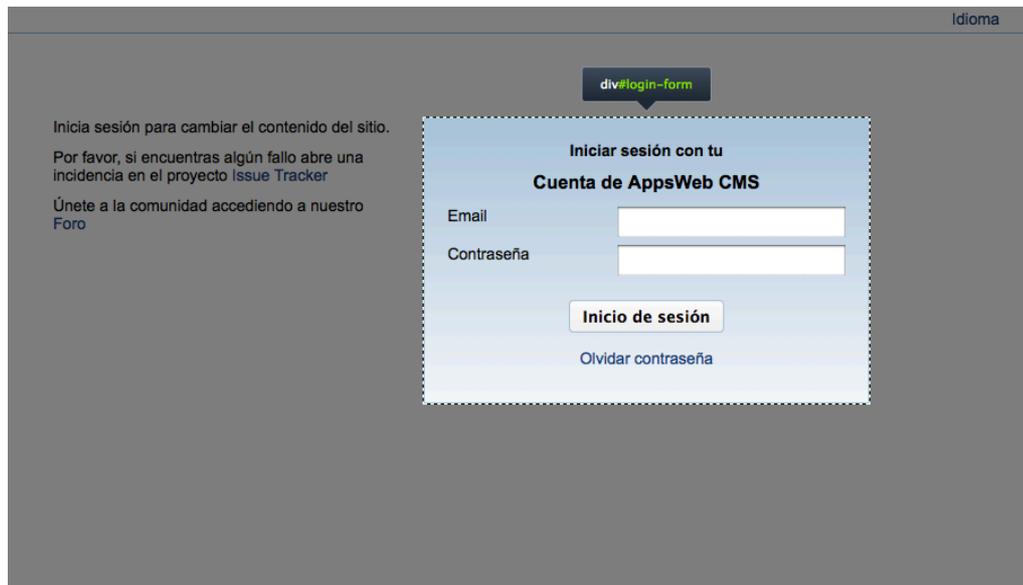
Direcciones reservadas:

- /ah
- /cms
- /plugin
- /static
- /login.jsp
- /file
- /setup
- /update
- /hotCron
- /JSON-RPC

## 2.6 MANUAL DE USUARIO

A continuación se muestran los pasos necesarios para configurar el CMS. Introduzca la siguiente url en el navegador: “http://localhost:8888/cms/”.

Se le dirigirá a la página para logearse en el cms y empezar una sesión como administrador.



**Ilustración 37:** Página de *login*

Si es la primera vez accede al sitio introduzca los siguientes credenciales:

Email: admin@test.com

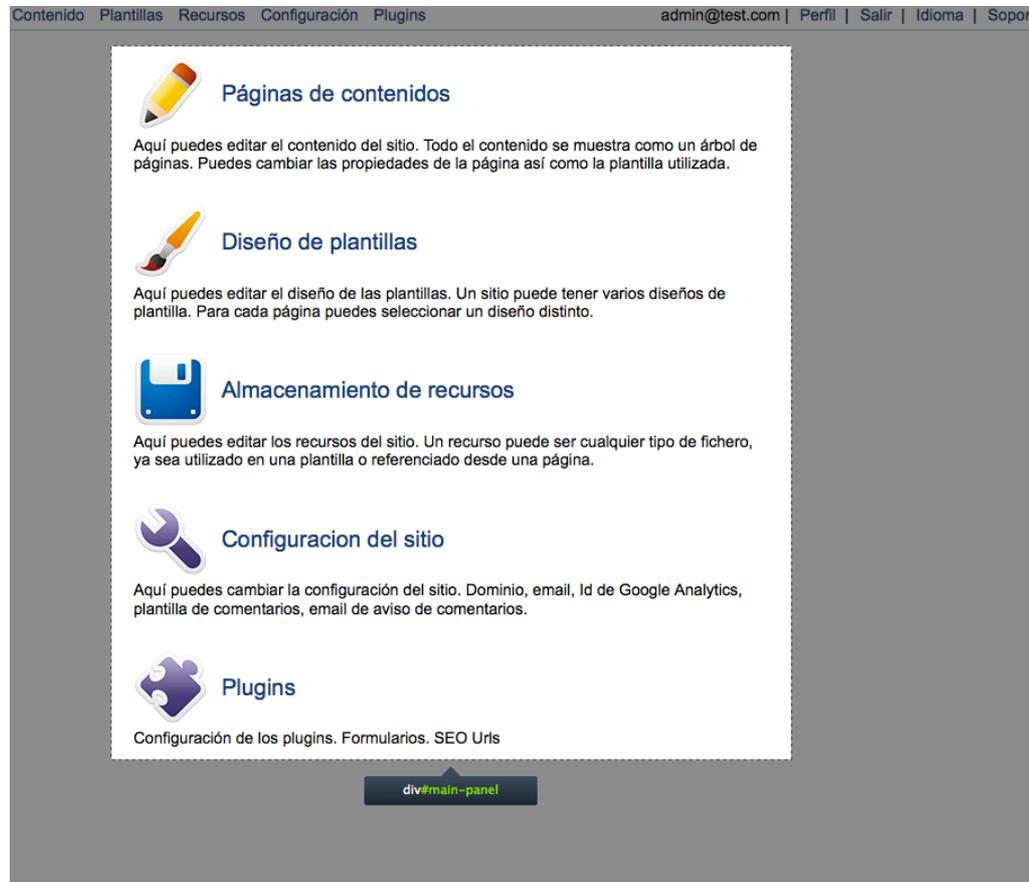
Contraseña: admin



**Ilustración 38:** Credenciales

## 2.6.1 Área de trabajo

Es la página que se muestra inmediatamente después de logearse. En ella se encuentran las opciones principales para manejar la información del sistema.



**Ilustración 39:** Página principal administrador

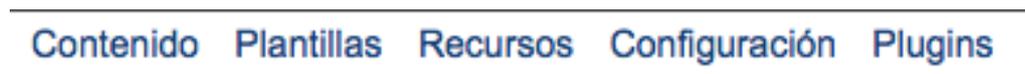
Esas opciones son:

- 1 Página de contenidos: Aquí puedes editar el contenido del sitio. Todo el contenido se muestra como un árbol de páginas. Puedes cambiar las propiedades de la página así como la plantilla utilizada.
- 2 Diseño de plantillas: Aquí puedes editar el diseño de las plantillas. Un sitio puede tener varios diseños de plantilla. Para cada página puedes seleccionar un diseño distinto.
- 3 Almacenamiento de recursos: Aquí puedes editar los recursos del sitio. Un recurso puede ser cualquier tipo de fichero, ya sea utilizado en una plantilla o referenciado desde una página.
- 4 Configuración del sitio: Aquí puedes cambiar la configuración del sitio. Dominio, email, Id de Google Analytics, plantilla de comentarios, email de aviso de comentarios.
- 5 Plugins: Configuración de los plugins. Formularios. SEO Urls

## 2.6.2 Navegación

La zona de administración cuenta con una menú principal en la parte superior dividido en dos partes. La primera indica las opciones principales para gestionar la información de las distintas entidades del sistema.

En la parte derecha de la barra de menú están las opciones de sesión del sistema. Existe información sobre quién está logeado, el perfil de ese usuario, el idioma configurado en el sistema, y la página de soporte del sistema.



**Ilustración 40:** Barra de menú izquierda

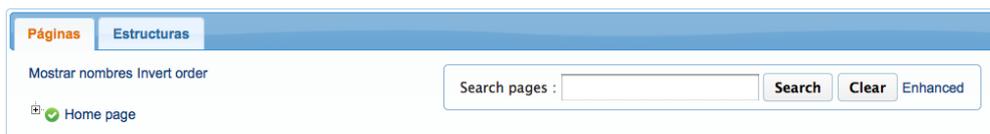


**Ilustración 41:** Barra de menú derecha

Ahora veremos las opciones disponibles una por una.

## 2.6.3 Contenido

La pestaña Contenido permite la edición y aprobación del contenido de la página.

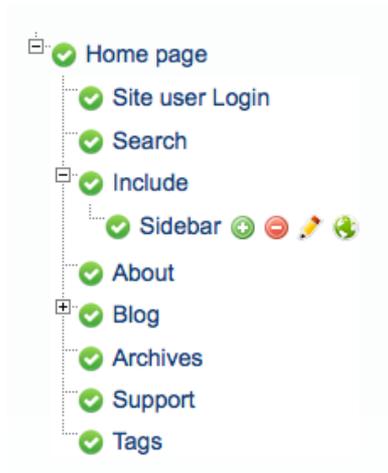


**Ilustración 42:** Contenido

Cuando seguimos el enlace de contenido nos encontramos un panel como muestra la imagen anterior. Podemos ver tres elementos.

Primero las pestañas que muestran las páginas y su estructura. En segundo lugar se puede ver el contenido de cada pestaña, en este caso el árbol principal sin expandir. Y por último una caja para realizar búsquedas de contenido en nuestro sistema.

Si expandimos el árbol de directorios tendremos algo parecido a lo siguiente. Un directorio principal *Home Page* y páginas que cuelgan de él.



**Ilustración 43:** Árbol de contenido

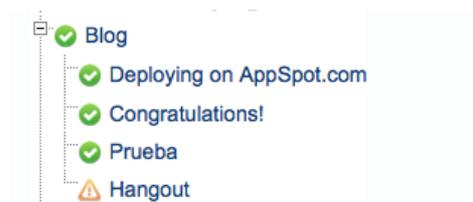
Puede ver la información que gestiona el sistema. Se pueden expandir los apartados que dispongan de subpáginas. Cada apartado cuenta con cuatro opciones disponibles representadas por cuatro iconos.

El icono añadir página gestiona el alta de una nueva página asociada al apartado que se ha pulsado, genera el siguiente cuadro de diálogo.



**Ilustración 44:** Nueva página contenido

Automáticamente se actualizará el árbol de directorios y mostrará una página más con un símbolo que indica que todavía no ha sido aprobada.



**Ilustración 45:** Página sin aprobar

Para aprobarla hay que situarse sobre el nombre de la página y pulsar el icono de editar página para continuar rellenando campos imprescindibles para la publicación de contenido.

Lo primero que nos encontramos al pulsar el icono editar página es lo siguiente.

Contenido: / blog / hangout

**Página** Contenido Subpáginas Comentarios Security Recursos Atributos

URL "amigable" /blog/

Restful

Template

Fecha de publicación (dd.mm.yyyy) 24-horas   -

Enable comments

Incluir en los resultados de búsqueda

Procesar con Velocity

Procesar sintaxis Mediawiki

Activar CKEditor

No post-procesar la página

Cache activada

URL de las dependencias del cache de página (lista separada por comas)

Tipo de página

Estructura

Plantilla de la Estructura

Tags [Añadir etiqueta](#)

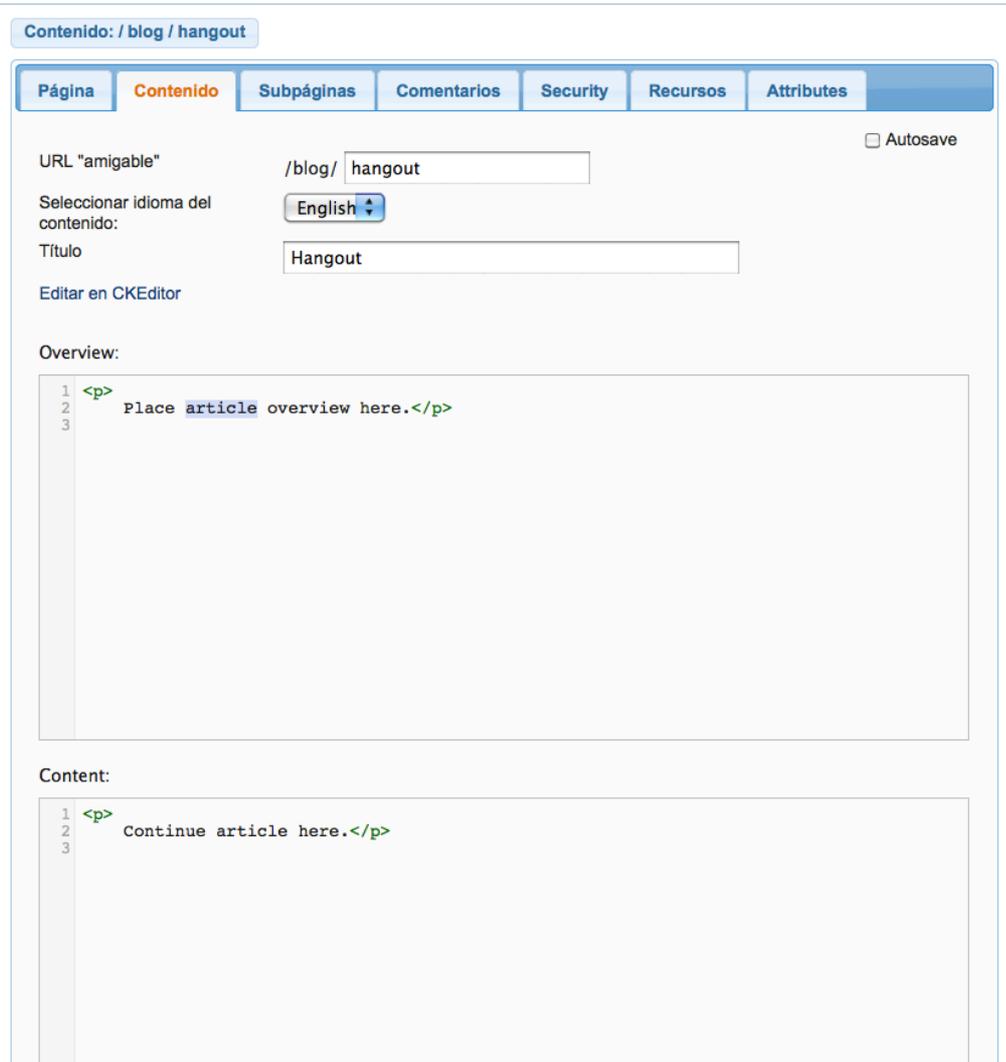
Metadata & Header

Aprobar al guardar

### Ilustración 46: Editar página

En este apartado hay campos que definen como se va a configurar la página. También palabras que se incluyen en el *header* de la página.

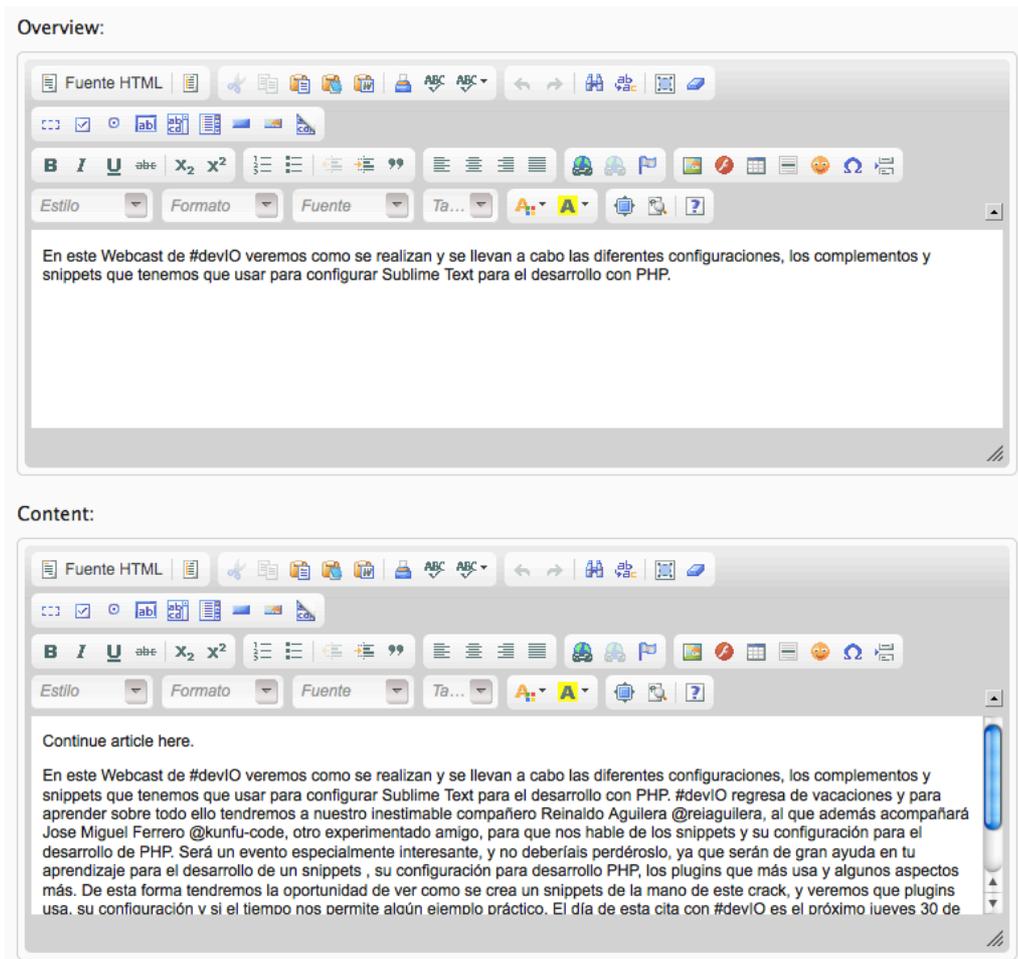
La siguiente pestaña muestra el contenido y la vista previa, en este caso de un artículo para el blog. Seleccionar el idioma para mostrar y la URL.



**Ilustración 47:** Editar página – Contenido

Podremos ver y editar el contenido como código, como muestra la imagen anterior o podemos hacer uso del editor CKEditor. Sirve para mostrar como se vería en el navegador lo que estás escribiendo.

Viene equipado con varias paletas para poder editar el estilo del contenido como si de un editor de textos se tratase.



**Ilustración 48:** CKEditor

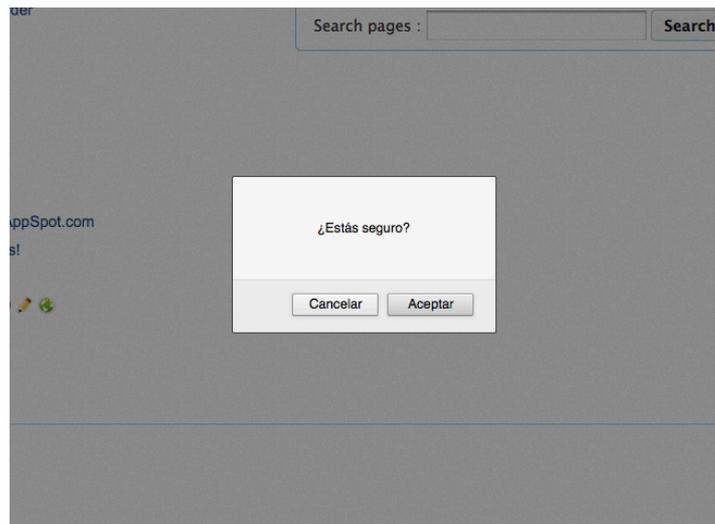
En todas las pestañas tenemos asociado un menú en la parte inferior con diferentes acciones como el siguiente.



**Ilustración 49:** Acciones

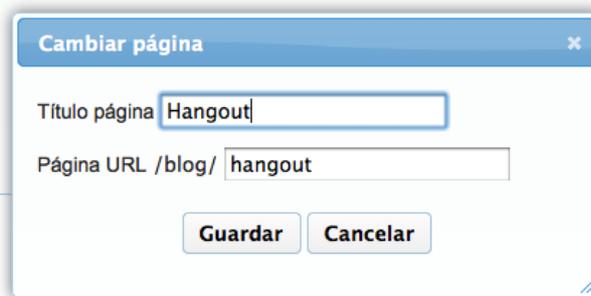
Para que el contenido pueda ser visualizado desde la parte externa de la zona de administración es necesario que se marque la opción Aprobar contenido.

El segundo icono de los elementos es para eliminar una página del árbol de directorios.



**Ilustración 50:** Eliminar página

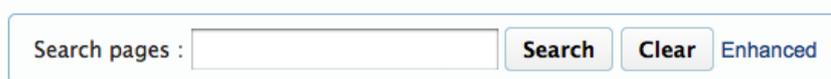
El último icono sirve para modificar los datos de título y URL de la página.



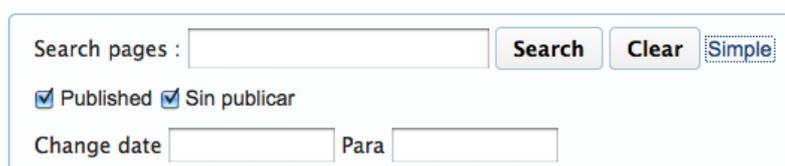
**Ilustración 51:** Modificar página

## 2.6.4 Búsquedas

En la zona de administración siempre hay habilitado un buscador para realizar cualquier tipo de búsqueda. Podemos utilizar el buscador en su versión sencilla o en su versión extendida.



**Ilustración 52:** Búsqueda simple



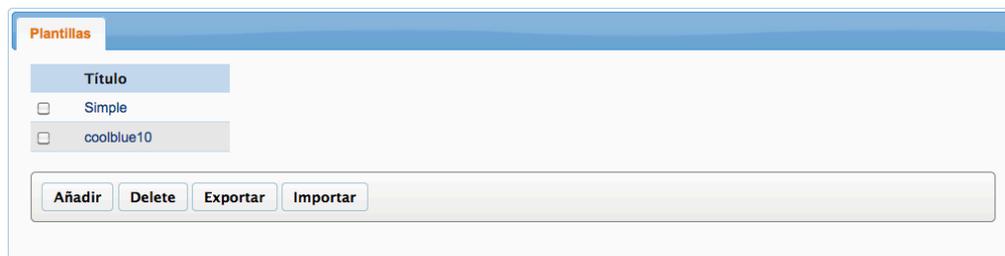
**Ilustración 53:** Búsqueda avanzada

## 2.6.5 Plantillas

La plantilla proporciona el formato Web que rodea el contenido de la página. Cada página debe estar asociado con una plantilla. Una plantilla simple se proporciona en la configuración de la aplicación por defecto. También se pueden diseñar plantillas con un diseño personalizado que satisfagan las necesidades personales.

El sistema es compatible con dos tipos de páginas, sencilla y estructurada. Sencillo páginas de contenido tienda en un campo de texto único por superficie. Páginas estructuradas pueden utilizar varios campos para almacenar contenido. La plantilla se puede colocar campos de contenido individuales en lugares específicos, para crear fácilmente un diseño uniforme.

El CMS utiliza un plugin popular, CKEdit, para crear y mantener contenido de la página. La interfaz es similar a otros productos y debe ser intuitiva para la mayoría de los usuarios.

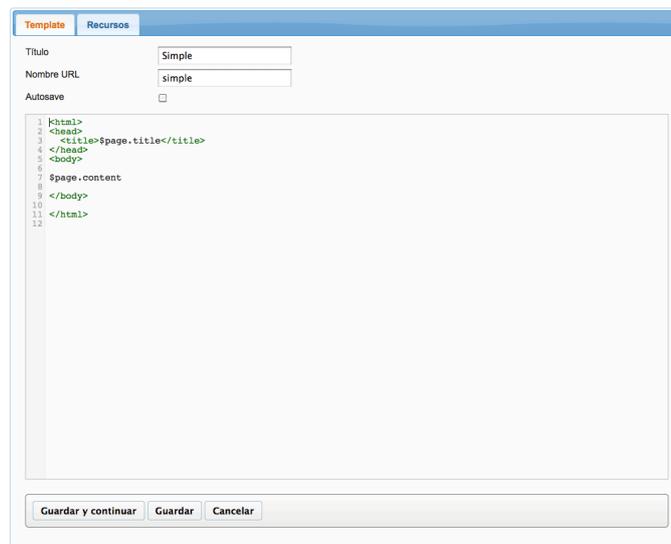


**Ilustración 54:** Plantillas

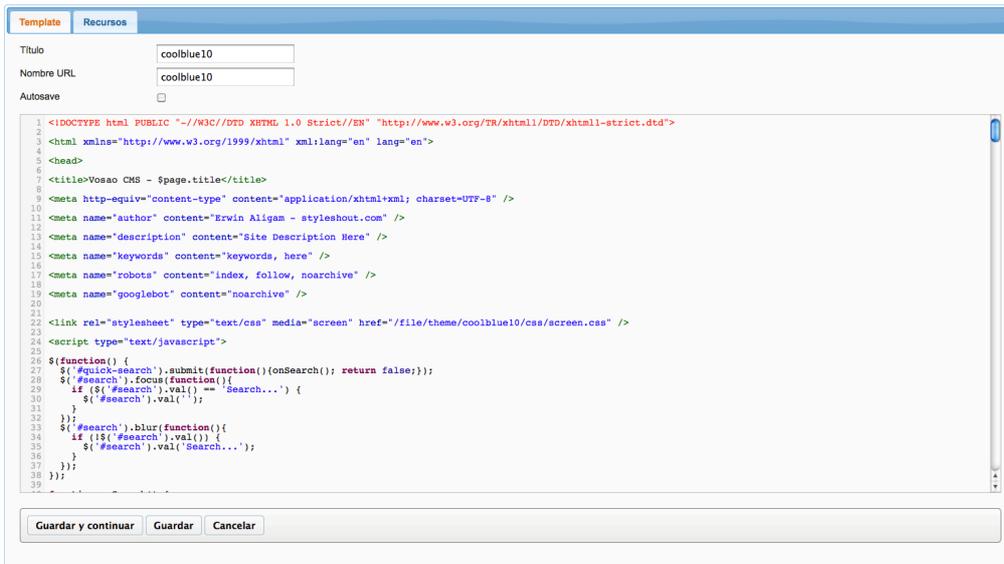
Hay cuatro opciones disponibles: añadir, eliminar, exportar e importar.

Cuando pulsamos sobre el nombre de alguna plantilla se muestran dos subconjuntos de opciones. Uno para su código y otro para recursos asociados.

Pueden ser plantillas de lo más sencillo a lo más complejo.

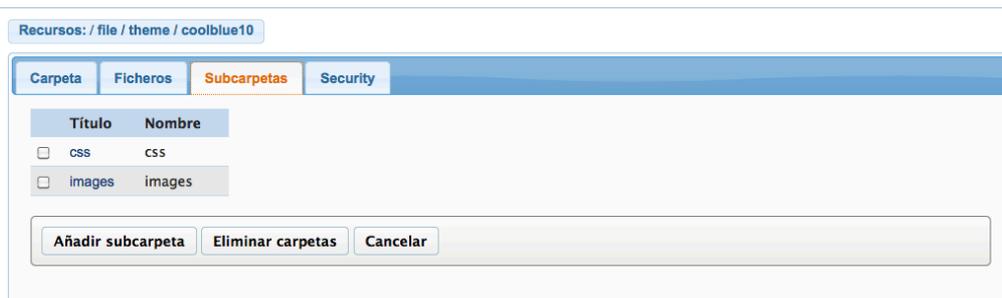


**Ilustración 55:** Plantilla simple



**Ilustración 56:** Plantilla compleja

Bajo la opción de recursos podemos ver ficheros asociados de tipo css, js, jpg, etc.



**Ilustración 57:** Recursos asociados a plantillas

## 2.6.6 Recursos

La opción recursos muestra el árbol de contenidos de los todos los recursos disponibles en el sistema. Se puede ver a que página pertenece cada uno de los recursos.

El cuadro de diálogo de imagen incluye una acción de búsqueda que muestra el árbol de recursos del CMS. Para enlazar a un recurso, selecciónelo en el árbol, y el editor va a generar la referencia adecuada. Para enlazar a una página, introduzca la referencia correspondiente.

La acción de nueva imagen tratará de subir un archivo nuevo en el área de recursos. Sobre la base de localización de la página, el sistema creará automáticamente una estructura de carpetas que coincide en el área de recursos bajo la carpeta de recursos / página.



**Ilustración 58:** Árbol de recursos

Pulsando sobre uno de los recursos nos aparece la siguiente información más detallada sobre el directorio o recurso que ha sido elegido.



**Ilustración 59:** Árbol de recursos

### 2.6.7 Configuración

Estas opciones permiten tener control sobre variables de configuración útiles y sobre la gestión de usuarios, comentarios y tags.

Los comentarios pueden ser habilitado para que cualquier página. La plantilla de página debe ser compatible con los comentarios y los comentarios deben estar habilitadas para cada página. El marcado de bloque de comentario se puede personalizar en la configuración del sitio.

Configuración del sitio | Comentarios | Idiomas | Paquete de mensajes | Usuarios | Grupos | Tags | Attributes

Versión del esquema de Base de Datos  
Google Analytics ID  
Email del propietario del sitio: admin@test.com  
**¡Atención!** La dirección de email debe pertenecer a un desarrollador de la aplicación en GAE.  
Site domain: test.com  
Activar el uso de reCaptcha en el sitio:   
Extensiones de ficheros editables (lista separada por comas): css,js,xml  
URL de la página de inicio de sesión: /login  
URL de la página 404  
Activar CKEditor:   
Zona horaria: ACT  
Seleccionar idioma del contenido: English  
Activar soporte para Picasa:

Guardar | Exportar | Importar | Crear índice de búsqueda | Reinicializar | Cargar sitio por defecto | Limpiar cache | Estadísticas

**Ilustración 60:** Árbol de recursos

La gestión de usuarios del sistema se gestiona desde una de las pestañas de configuración. Desde aquí se podrán cambiar los roles de usuario.

Nombre	Email	Rol	Acceso
<input type="checkbox"/> admin	admin@test.com	Administrador	Enabled
<input type="checkbox"/> Diego	dyaguejuarez@gmail.com	Administrador	Enabled

Añadir | Eliminar

**Ilustración 61:** Gestión de usuarios

## 2.6.8 Plugins

Sitio para configurar plugins adicionales para que el sistema aumente su funcionalidad. Existen dos plugins instalados, uno para formularios y otro para URLs.



### Configuración de Plugins

Puedes instalar y configurar los plugins disponibles.



### Forms

El plugin Forms te facilita la creación de distintos tipos de formularios. La información introducida por el usuario será enviada a un dirección concreta.



### SEO URLs

Este plugin facilita la indexación SEO y las campañas de anuncios. Aquí puede introducir una lista de alias de URLs que se asignarán a una URL del sitio. Si estas migrando desde otro sitio, las URL de SEO pueden ayudarte a mantener la valoración del sitio y preservar las inversiones en SEO.

## Ilustración 62: Plugins

## 2.6.9 Perfil, salir, idioma



## Ilustración 63: Idiomas

The image shows a web form titled "Perfil usuario" with a blue header bar. The form contains the following fields and controls:

- User name:** A text input field containing the text "Diego".
- Email usuario:** A text input field containing the email address "dyaguejuarez@gmail.com".
- Contraseña:** A password input field with a vertical line indicating the cursor position.
- Reescribe la contraseña:** An empty text input field for re-entering the password.
- Zona horaria:** A dropdown menu with "Europe/Madrid" selected and a blue arrow icon on the right.

At the bottom of the form is a large, light-colored button labeled "Guardar".

**Ilustración 64:** Perfil de usuario

### 3. ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: SELECCIONAR IDIOMA DE DESCARGA DE GOOGLE CHROME WINDOWS	128
ILUSTRACIÓN 2: SELECCIONAR ENLACE DE DESCARGA.....	129
ILUSTRACIÓN 3: ACEPTAR CONDICIONES GOOGLE CHROME .....	129
ILUSTRACIÓN 4: SELECCIONAR IDIOMA DE DESCARGA MAC.....	130
ILUSTRACIÓN 5: SELECCIONAR ENLACE DE DESCARGA.....	130
ILUSTRACIÓN 6: WORKSPACE ECLIPSE .....	131
ILUSTRACIÓN 7: INSTALANDO APP ENGINE.....	132
ILUSTRACIÓN 8: INSTALACIÓN DE MAVEN.....	133
ILUSTRACIÓN 9: IMPORTAR PROYECTO MAVEN .....	134
ILUSTRACIÓN 10: COMPILAR CON MAVEN.....	135
ILUSTRACIÓN 11: PÁGINA DE <i>LOGIN</i> .....	137
ILUSTRACIÓN 12: CREDENCIALES .....	137
ILUSTRACIÓN 13: BARRA DE MENÚ IZQUIERDA .....	139
ILUSTRACIÓN 14: BARRA DE MENÚ DERECHA .....	139
ILUSTRACIÓN 15: CONTENIDO.....	139
ILUSTRACIÓN 16: ÁRBOL DE CONTENIDO .....	140
ILUSTRACIÓN 17: NUEVA PÁGINA CONTENIDO .....	140
ILUSTRACIÓN 18: PÁGINA SIN APROBAR.....	140
ILUSTRACIÓN 19: EDITAR PÁGINA .....	141
ILUSTRACIÓN 20: EDITAR PÁGINA – CONTENIDO .....	142
ILUSTRACIÓN 21: CKEDITOR .....	143
ILUSTRACIÓN 22: ACCIONES .....	143
ILUSTRACIÓN 23: ELIMINAR PÁGINA .....	144
ILUSTRACIÓN 24: MODIFICAR PÁGINA .....	144
ILUSTRACIÓN 25: BÚSQUEDA SIMPLE.....	144
ILUSTRACIÓN 26: BÚSQUEDA AVANZADA .....	144
ILUSTRACIÓN 27: PLANTILLAS .....	145
ILUSTRACIÓN 28: PLANTILLA SIMPLE .....	145
ILUSTRACIÓN 29: PLANTILLA COMPLEJA .....	146
ILUSTRACIÓN 30: RECURSOS ASOCIADOS A PLANTILLAS .....	146
ILUSTRACIÓN 31: ÁRBOL DE RECURSOS .....	147
ILUSTRACIÓN 32: ÁRBOL DE RECURSOS .....	147
ILUSTRACIÓN 33: ÁRBOL DE RECURSOS .....	148
ILUSTRACIÓN 34: GESTIÓN DE USUARIOS .....	148
ILUSTRACIÓN 35: PLUGINS.....	149
ILUSTRACIÓN 36: IDIOMAS.....	149
ILUSTRACIÓN 37: PERFIL DE USUARIO .....	150