



**Universitat Autònoma
de Barcelona**

Kaos en la Red 2.0

Memòria del projecte

d'Enginyeria Tècnica en

Informàtica de Gestió

realitzat per

Jose Manuel Aguilar Nuñez

i dirigit per

Daniel Franco

Escola Universitària d'Informàtica

Sabadell, juliol de 2008

Tabla de contenido

1. Resumen de contenido	1
2. Introducción	2
3. Estudio de viabilidad	3
3.1. Situación actual de la página Web	3
3.2. Objetivos	3
3.3. Destinatarios de la aplicación	4
3.4. Fuentes de información	4
3.5. Descripción del sistema	5
3.6. Modelo de desarrollo	6
3.7. Recursos	8
3.7.1. Software	8
3.7.2. Hardware	9
3.7.3. Humanos	9
3.8. Costes	9
3.8.1. Coste de materiales	9
3.8.2. Coste de personal	10
3.9. Evaluación de riesgos	11
3.10. Viabilidad legal	11
3.11. Viabilidad técnica	12
3.12. Planificación del proyecto	12
3.12.1. <i>Estimación del tiempo</i>	12
3.12.2. <i>Diagrama de Gantt</i>	13
3.13. Conclusiones	14
4. Análisis de requerimientos	15
4.1. Introducción	15
4.2. Registro de usuarios	15
4.3. Perfil de usuarios	17
4.4. Personalización de la página Web	18
4.5. Blog	19
4.6. Administración de usuarios	19
5. Herramientas de desarrollo	20
5.1. Introducción	20
5.2. Herramientas para el diseño	20
5.3. Herramientas para la implementación	21
6. Diseño	29
6.1. Introducción	29
6.2. Descripción de los usuarios del sistema	29

6.3. Diagrama de casos de uso	30
6.4. Especificación de los casos de uso	31
6.4.1. <i>Registro</i>	31
6.4.2. <i>Login</i>	33
6.4.3. <i>Gestión de perfil</i>	34
6.4.4. <i>Gestión de noticias</i>	35
6.4.5. <i>Gestión de comentarios</i>	36
6.4.6. <i>Gestión de blog</i>	37
6.4.7. <i>Personalización de la página principal</i>	38
6.4.8. <i>Control de usuarios</i>	39
6.4.9. <i>Favoritos</i>	40
6.5. Diagrama de actividades	41
6.6. Diagrama de clases	50
6.7. Diagrama de secuencia	55
6.8. La base de datos	60
7. Implementación	65
7.1. Introducción	65
7.2. Estructura de archivos	65
7.3. Ejemplo de codificación	66
8. Pruebas	71
8.1. Introducción	71
8.2. Pruebas de unicidad	71
8.3. Pruebas de integración	71
8.4. Pruebas de validación	72
8.5. Pruebas de sistema	72
8.6. Pruebas de compatibilidad con navegadores Web	72
8.7. Pruebas de aceptación	73
9. Conclusiones	74
10. Resumen	75
Bibliografía y fuentes electrónicas	
Anexo 1 – Manual de usuario	

1. Resumen de contenido

En este apartado se va a hacer una ilustración del contenido de esta memoria. En principio, se ha hecho una breve introducción donde se revela para quien va dirigido este proyecto, la asociación que se beneficiará de ello, los principales objetivos y algunas motivaciones para realizarlo.

A continuación, se expone un estudio previo de viabilidad donde se explica la situación actual de la página Web, los objetivos más en detalle, los destinatarios de la información, fuentes de información, descripción del sistema, modelo de desarrollo, recursos utilizados, coste de materiales, evaluación de riesgos, viabilidad legal, viabilidad técnica y finalmente la planificación del proyecto. La conclusión de este estudio dicta que el proyecto es viable y, por lo tanto, se puede diseñar e implementar.

Seguidamente, se analiza en detalle los requerimientos del software impuestos por el cliente y se habla de las herramientas de desarrollo utilizadas tanto para el diseño de la aplicación como para la implementación.

Consecutivamente, se detalla el diseño donde se describe los usuarios del sistema, se muestra un diagrama de casos de uso, se describe una especificación de cada caso de uso, se visualiza y explica el diagrama de actividades de cada caso de uso, un diagrama de clases, un diagrama de secuencia de cada caso de uso y, finalmente, el diseño de la base de datos.

Lo siguiente que se ha hecho es el apartado de implementación donde se comenta como es la estructura de archivos del proyecto y se expone en detalle un ejemplo de codificación de uno de los módulos implementados.

Finalmente, cabe mencionar las pruebas que se han realizado, en este caso, se han hecho pruebas de unicidad, de integración, de validación, de sistema, de compatibilidad con diferentes navegadores y de aceptación.

2. Introducción

Kaosenlared es un colectivo de personas que conforman una Asociación Cultural, que tiene como objetivo principal luchar contra el capitalismo, para lo cual cuenta, entre otras herramientas, con un medio de expresión digital que solo se publica en Internet.

La página Web de *Kaosenlared* es un medio de comunicación libre y alternativo. Se quiere dar un paso adelante y profundizar en lo que se llama Web 2.0 ¹.

Se trata de un proyecto relacionado con la ingeniería del software o software de desarrollo en el cual se van a utilizar lenguajes de programación del lado del servidor de última generación, bases de datos, lenguaje para creación de páginas Web, entre otros.

La aplicación Web deberá permitir a usuarios registrarse, iniciar sesión, hacerse un perfil, consultar, editar y eliminar sus noticias y comentarios, hacerse su “Kaos” personalizando la página principal y tener un blog.

Como objetivos principales se puede destacar:

- Mejorar la situación y apariencia de la página Web www.kaosenlared.net.
- Permitir que los usuarios interactúen aún más con el sitio Web.
- Que los administradores puedan tener un control de las posibles acciones que puedan hacer los usuarios, es decir, altas, bajas, modificaciones...

Las motivaciones para realizar este proyecto son el hecho de que se va a trabajar con el término Web 2.0, muy actual hoy en día, con posibilidades para el futuro.

Es un proyecto que va dirigido principalmente a más de 25.000 usuarios de IP's diferentes que se conectan a la página Web para informarse de noticias de interés. Seguramente, al dinamizar un poco más la página, al ser más atractiva, más usuarios la visitarán.

¹ El término Web 2.0 fue introducido por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs, los wikis, etc., que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

3. Estudio de viabilidad

3.1. Situación actual de la página Web

El colectivo de *Kaosenlared* empezó teniendo una página Web estática, implementada en HTML simple. Se trataba de “Kaos” 1.0. Al cabo de un tiempo, se dinamizó un poco pudiendo que usuarios pudieran publicar y comentar noticias pero sin previo registro, se trataba de la versión 2.0 de “Kaos”, utilizando en este caso HTML, PHP y MySQL.

La situación actual de la página Web es que cualquier usuario puede entrar y leer cualquier noticia, comentarla, crear nuevas noticias, pero todo ello sin previo registro en una base de datos.

Los administradores solo pueden hacer una serie de funcionalidades, pero nada relacionado con la administración de usuarios.

Lo que se pretende ahora es realizar la versión 3.0 en la cual se va a dinamizar y dar más interactividad a los usuarios y control por parte de la administración de dichos usuarios.

3.2. Objetivos

El proyecto que se presenta es la continuación de una página Web con el objetivo de añadir nuevas funcionalidades:

- Permitir el registro de usuarios.
- Control, por parte de la administración, de los usuarios. Los administradores podrán ver la lista de usuarios registrados, ver los usuarios que están conectados en un momento determinado, modificar registros, eliminar usuarios, etc.
- Diseño más interactivo del sitio Web.

A parte de estos principales objetivos, también hay otros relacionados con la Web 2.0, como por ejemplo que los usuarios dispongan de su propio blog, tengan un perfil de usuario donde puedan consultar sus datos, sus noticias y sus comentarios, y que puedan cambiar el estilo o color y las categorías a mostrar de la página principal, es decir, personalización de la página principal.

3.3. Destinatarios de la aplicación

El destinatario de la aplicación Web es *Kaosenlared* y sus lectores (usuarios). El usuario del nuevo sistema de información puede ser cualquier persona con necesidad de informarse a menudo, o con ideología de Izquierda Plural Anticapitalista que tengan algo que aportar para defender a los seres más desprotegidos de los abusos del capitalismo depredador, y de los Gobiernos que lo sustentan, que existen en todo el mundo. Puede ser una persona con o sin conocimientos de informática pero para estar a gusto en el Web está claro que ha de ser una persona que comparta la misma ideología.

3.4. Fuentes de información

Como en toda aplicación, la principal fuente de información son los usuarios finales. En este caso, los lectores interesados en la materia, quienes podría ser que les interesase tener un espacio Web relacionado con su ideología.

Otra fuente de información muy importante es la asociación de la página Web, ya que sabe a qué tipo de gente va dirigido la información y sabe lo que se necesita.

Pero no puede faltar la fuente de información más utilizada por todos hoy en día, Internet, fijándose en la competencia y de esta manera poder solucionar los posibles problemas que puedan aparecer.

Actualmente existen varias páginas Web que también suministran noticias de todo tipo a usuarios, por ejemplo:

Soitu que es un sitio de emisión de información independiente, útil y abierto, indicado para personas con inquietudes que buscan contenidos en los que se conjuga el rigor del periodismo tradicional con la filosofía de Internet.

Manéame es un Web que te permite enviar una historia que será revisada por todos y será promovida, o no, a la página principal. Cuando un usuario envía una noticia ésta queda en la cola de pendientes hasta que reúne los votos suficientes para ser promovida a la página principal.

Libertad Digital es un Web que muestra noticias de última hora sobre España y el resto del mundo. Se trata de una empresa emisora de información situada en Madrid llamada Libertad Digital S.A.

3.5. Descripción del sistema

Una de las principales funcionalidades que se pretende realizar con este proyecto es la necesidad de que los usuarios puedan registrarse para así tener una serie de privilegios. El registro consistirá en mostrar al usuario un formulario solicitando nombre de usuario, contraseña y email, verificando los datos introducidos de forma dinámica. Una vez registrado, esos datos se guardarán en la base de datos de la organización y se enviará un mensaje de bienvenida a la dirección de correo electrónico facilitado por el usuario.

Sabiendo los datos de acceso, es decir, nombre de usuario y contraseña, el nuevo usuario ya podrá iniciar sesión. Con este inicio de sesión, se le presentará al usuario nuevas funcionalidades que de entrada no están disponibles.

Estas nuevas funcionalidades de las que dispondrá ahora el usuario son:

- Poder ver y modificar su perfil de usuario, es decir, mostrar o actualizar datos relevantes y personales del usuario.
- Ver sus noticias publicadas, eliminarlas o modificarlas, es decir, mostrar una lista de todas las noticias que el usuario ha aportado en el tiempo que lleva registrado teniendo la opción de eliminarlas o modificarlas cuando lo crea conveniente.
- Ver sus comentarios realizados, eliminarlos o modificarlos, es decir, mostrar una lista de todos los comentarios que el usuario ha ido realizando a lo largo de su registro en el sitio Web.

- Personalizar la página principal a través de una interfaz, es decir, al usuario se le mostrarán atributos de diseño que podrá aplicar y guardar para que su página principal esté a su gusto. A parte, también podrá seleccionar aquellas categorías de noticias que desee visualizar en la página principal.
- Los usuarios dispondrán de su propio blog, donde periódicamente podrán actualizar textos o artículos, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

En cuanto a funcionalidades que va a realizar el administrador, tenemos el poder consultar una lista de todos los usuarios registrados, ver los usuarios conectados en un momento determinado, modificar noticias, comentarios e incluso datos de usuario y eliminar usuarios conflictivos.

3.6. **Modelo de desarrollo**

Para desempeñar este proyecto se ha decidido utilizar un proceso de desarrollo evolutivo, donde el software se va a realizar por incrementos y cada incremento añadirá una nueva funcionalidad al sistema.

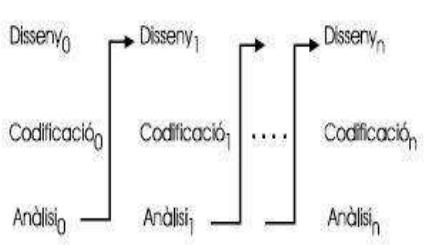


Figura 1. *Modelo de desarrollo evolutivo*

Se ha decidido utilizar esta forma de organización porque las funcionalidades que se pretenden implementar son, de alguna manera, independientes. Se trataría de dividir un problema grande (todos los requerimientos), en subproblemas (un requerimiento en concreto).

Cada incremento o subproblema se analizará, diseñará, implementará y se probará y esto supondrá una ventaja ya que el sistema recibirá pruebas más exhaustivas y precisas.

¿Qué entendemos por analizar? Primero, se va a hacer un estudio con el que se decidirá si el proyecto es viable o no, se describe el proyecto de forma global y se presenta un primer análisis y los requerimientos de este (**análisis de viabilidad**).

Seguidamente se va a realizar un análisis de la funcionalidad que se pretende alcanzar y las posibles restricciones del sistema (**análisis de requerimientos**). Esta parte sirve para hacer saber al usuario que se ha comprendido cada subproblema.

El **diseño** es la fase en la que se planificará la forma en la que se solucionarán los diferentes subproblemas que anteriormente se han analizado y poder decidir cuál es la manera más óptima de llegar a la solución.

Cuando ya se tiene diseñado un subproblema, podremos empezar la **codificación** del mismo con los lenguajes de programación que previamente se haya decidido utilizar. Cuando hayamos codificado cada subproblema obtendremos el resultado de la aplicación en si.

Una vez codificado un subproblema, se puede desarrollar la fase de **pruebas**, donde se propondrá las diferentes pruebas que se deban efectuar. Si la aplicación realiza las pruebas con éxito, estará listo para su utilización.

A medida que se realiza el proyecto, se irá redactando una **memoria** de forma ordenada y en un lenguaje claro todos los pasos descritos anteriormente.

Cuando ya hayamos analizado, diseñado, codificado y redactado todos los subproblemas, podremos pasar a la **instalación y mantenimiento** de la aplicación en el cliente.

3.7. Recursos

3.7.1. Software

El software necesario para desarrollar este proyecto son:



- Disponer de un ordenador con sistema operativo: Windows Xp Home Edition o superior.
- Disponer de una conexión a Internet, tener instalado Apache, PHP y MySQL. Una recomendación para trabajar en local es instalar el paquete de XAMPP (servidor Apache, lenguaje de programación PHP y la base de datos MySQL, entre otras aplicaciones para Windows).
- Tener instalados varios navegadores de Internet para hacer pruebas con cada uno de ellos, como mínimo; Internet Explorer y Mozilla Firefox.
- Otro requisito o recomendación es tener instalado Macromedia Dreamweaver 8 para el desarrollo de los scripts de PHP y JavaScript.
- Para diseñar la base de datos del sistema, se va a utilizar un programa muy útil y sencillo de utilizar llamado BDDesigner4 que de forma gráfica te crea las relaciones de las tablas que tu deseas y posteriormente te permite generar el código SQL que será el archivo que se importará en PHPMyAdmin, que es el programa administrador de la base de datos que ya incorpora el paquete XAMPP.
- Y para la generación de la documentación se va a utilizar Microsoft Office 2003, Microsoft Project para hacer la planificación del proyecto, Microsoft Visio para realizar los diferentes diagramas de diseño del sistema y Adobe Acrobat con PDFCreator para generar la memoria definitiva.

El usuario lo único que necesitará para poder navegar por esta Web será disponer de una conexión a Internet y tener instalado algún navegador de versión actual o no muy desfasada.

3.7.2 Hardware

El hardware necesario para desarrollar este proyecto es disponer de una memoria RAM de 512 MB como mínimo, un procesador Pentium III y un disco duro de 50 GB y una tarjeta de red.



El cliente los requisitos mínimos que debe cumplir son disponer de una memoria RAM de 256 MB, un procesador Pentium II, un disco duro de 50 GB y una tarjeta de red.

3.7.3. Humanos

Como en la mayoría de proyectos de final de carrera, el único recurso humano es solamente su autor con la colaboración y guía de los tutores, en este caso tanto de la asociación como de la Universidad.

3.8. Costes

3.8.1. Coste de materiales

Recursos	Coste Total	Coste mensual
Conexión Internet 20Mb	239,60 €	29,95 €
Windows XP	0,00 €	
Apache	0,00 €	
PHP 4.0	0,00 €	
MySQL 5.0	0,00 €	
PHPMyAdmin	0,00 €	
Macromedia Dreamweaver 8	0,00 €	
Amortización PC programador	350,00 €	
Amortización MSPROJECT analista	45,00 €	
Amortización MSOffice 2003	125,00 €	
Total	759,60 €	29,95 €

- Duración del proyecto = 8 meses
- Amortización del PC programador = $(700/8)*4 = 350$ €
- Amortización del MSPROJECT = $(360/8)*1 = 45$ €
- Amortización del MSOffice 2003 = $(250/8)*4 = 125$ €

3.8.2. Coste de personal

Teniendo en cuenta la tabla de recursos, hemos realizado la siguiente tabla en la que se detalla las tareas que se van a realizar, las horas que se le va a dedicar, su coste en función de la tabla de recursos y las horas de trabajo.

Tareas	Horas	Coste	Recurso
Comprensión del problema	2	50,00 €	Analista
Estudio de viabilidad	20	500,00 €	Analista
Instalación del software necesario	4	72,00 €	Programador
Registro de usuarios	29	625,00 €	
<i>Análisis de requerimientos</i>	1	25,00 €	Analista
<i>Diseño y configuración de la base de datos</i>	5	125,00 €	Analista
<i>Codificación</i>	10	180,00 €	Programador
<i>Pruebas locales</i>	2	30,00 €	Técnico de pruebas
<i>Pruebas de compatibilidad con navegadores</i>	1	15,00 €	Técnico de pruebas
<i>Elaboración de la documentación</i>	10	250,00 €	Analista
Perfil de usuarios	35	740,00 €	
<i>Análisis de requerimientos</i>	1	25,00 €	Analista
<i>Diseño y configuración de la base de datos</i>	6	150,00 €	Analista
<i>Codificación</i>	15	270,00 €	Programador
<i>Pruebas locales</i>	2	30,00 €	Técnico de pruebas
<i>Pruebas de compatibilidad con navegadores</i>	1	15,00 €	Técnico de pruebas
<i>Elaboración de la documentación</i>	10	250,00 €	Analista
Creación del blog	110	2.420,00 €	
<i>Análisis de requerimientos</i>	2	50,00 €	Analista
<i>Diseño y configuración de la base de datos</i>	28	700,00 €	Analista
<i>Codificación</i>	40	720,00 €	Programador
<i>Pruebas locales</i>	4	60,00 €	Técnico de pruebas
<i>Pruebas de compatibilidad con navegadores</i>	1	15,00 €	Técnico de pruebas
<i>Elaboración de la documentación</i>	35	875,00 €	Analista
Administración de usuarios	42	880,00 €	
<i>Análisis de requerimientos</i>	1	25,00 €	Analista
<i>Diseño y configuración de la base de datos</i>	8	200,00 €	Analista
<i>Codificación</i>	20	360,00 €	Programador
<i>Pruebas locales</i>	2	30,00 €	Técnico de pruebas
<i>Pruebas de compatibilidad con navegadores</i>	1	15,00 €	Técnico de pruebas
<i>Elaboración de la documentación</i>	10	250,00 €	Analista
Interfaz Web	20	360,00 €	Programador
Pruebas finales	10	150,00 €	Técnico de pruebas

Horas totales del proyecto	272
-----------------------------------	------------

Coste total del proyecto	5.797,00 €
Recurso	Coste/h
Analista	25
Programador	18
Técnico de pruebas	15

3.9. Evaluación de riesgos

Como se trata de un proyecto relacionado con la Web, el principal riesgo que hay que tener en cuenta es la incompatibilidad entre los navegadores de Internet que existen hoy en día y los problemas de conexiones y líneas telefónicas.

Debido a problemas técnicos del desarrollador del proyecto, es posible que no se cumplan las fechas previstas, el presupuesto acordado o los requerimientos funcionales impuestos por el cliente.

A medida que se va desarrollando el proyecto, pueden surgir otros riesgos difíciles de evitar, como por ejemplo: cambios en la legislación, cambios en los requerimientos por parte del cliente, disponibilidad, escasez de materiales, entre otros.

3.10. Viabilidad legal

Es importante tener en cuenta la seguridad de los datos, es decir, tener especial cuidado con la información que se maneja, sobretodo de los distintos usuarios para no tener problemas con la Ley Orgánica de Protección de Datos (LOPD).

En el caso de este proyecto, el hecho de enviar información por email sin finalidad comercial puede ser un problema en cuanto un usuario desea recuperar sus datos de acceso y solicita el servicio para que le envíen sus datos de acceso a su cuenta de correo electrónico, o al registrarse y suministre una dirección de correo electrónico que no es la suya.

En cuanto al registro de los datos personales de un usuario en la base de datos, se ha de tener muy en cuenta esta ley (LOPD). Según esta ley, los datos que se pretenden guardar con esta aplicación serán de nivel básico: algunos datos personales y foto. Pero estos datos no serán de carácter obligatorio, ya que será el usuario quien de forma voluntaria rellena su perfil con algunos datos personales. Los únicos datos obligatorios para pertenecer a la comunidad son el nombre de usuario, contraseña y email. Por lo tanto, no creo que se tengan problemas con esta ley.

3.11. Viabilidad técnica

En el momento de la asignación de este proyecto ya se dispone de las ideas y metodologías para elaborar documentación sobre ello. El único problema que puede surgir, es a la hora de codificar el diseño, por el motivo de que el alumno no es un experto en los lenguajes de programación que se pretende utilizar.

El hecho de que el alumno está muy interesado en aprender y adentrarse en los lenguajes de programación a utilizar, desea desarrollar aplicaciones dinámicas con denominación Web 2.0 y la gran cantidad de información que aporta Internet al respecto, podemos concluir que técnicamente el proyecto es viable.

3.12. Planificación del proyecto

3.12.1. Estimación del tiempo

La planificación de este proyecto se va a desarrollar teniendo en cuenta que solo se va a disponer de 4 días laborales a la semana: miércoles, jueves, viernes y sábados con jornada matinal de 4 horas.

Tareas	Horas
Comprendión del problema	2
Estudio de viabilidad	20
Instalación del software necesario	4
Registro de usuarios	29
Perfil de usuarios	35
Creación del blog	110
Administración de usuarios	42
Interfiecie Web	20
Pruebas finales	10
Horas totales del proyecto	272

Se ha realizado una estimación del tiempo y vemos que este proyecto va a tener una duración aproximada de 272 horas, que se considera que está dentro del rango de proyectos viables.

3.12.2. Diagrama de Gantt

En la siguiente tabla se muestra todas las tareas que se pretenden desarrollar, su duración, fecha de inicio, fecha de fin, la tarea predecesora y el nombre del recurso que desarrolla tal tarea:

	Nombre de tarea	Duración	Comienzo	Fin	Pred	Nombres de los recursos
1	Comprensión del problema	2 horas	mié 19/11/08	mié 19/11/08		Analista[50%]
2	Estudio de viabilidad	20 horas	mié 19/11/08	jue 27/11/08	1	Analista[50%]
3	Instalación del software necesario	4 horas	jue 27/11/08	vie 28/11/08	2	Programador[50%]
4	+ Registro de usuarios	3,63 días	vie 28/11/08	jue 11/12/08	3	
5	Análisis de requerimientos	1 hora	vie 28/11/08	vie 28/11/08	3	Analista
6	Diseño y configuración de la base de datos	5 horas	vie 28/11/08	sáb 29/11/08	5	Analista
7	Codificación	10 horas	mié 03/12/08	vie 05/12/08	6	Programador
8	Pruebas locales	2 horas	vie 05/12/08	vie 05/12/08	7	Técnico de pruebas
9	Pruebas de compatibilidad con navegadores	1 hora	sáb 06/12/08	sáb 06/12/08	8	Técnico de pruebas
10	Elaboración de la documentación	10 horas	sáb 06/12/08	jue 11/12/08	9	Analista
11	+ Perfil de usuarios	4,38 días	jue 11/12/08	vie 26/12/08	10	
18	+ Creación del blog	13,75 días	vie 26/12/08	jue 12/02/09	17	
25	+ Administración de usuarios	5,25 días	vie 13/02/09	mié 04/03/09	24	
32	Interfaz Web	20 horas	mié 04/03/09	jue 12/03/09	31	Programador
33	Pruebas finales	10 horas	jue 12/03/09	sáb 14/03/09	32	Técnico de pruebas

Figura 2. Tabla de tareas del proyecto

Como podemos observar, el proyecto se va a realizar por módulos, donde cada módulo se va a analizar, diseñar, codificar, hacer pruebas y elaborar la documentación correspondiente.

Cuando todos los módulos estén implementados, se procederá a realizar unas pruebas finales para comprobar que todo funciona correctamente.

En la siguiente figura tenemos el diagrama de Gantt desarrollado con Microsoft Project 2003 donde se detalla la planificación inicial y como se espera que el proyecto avance. Cada número del eje vertical corresponde al número de tarea de la figura 2.

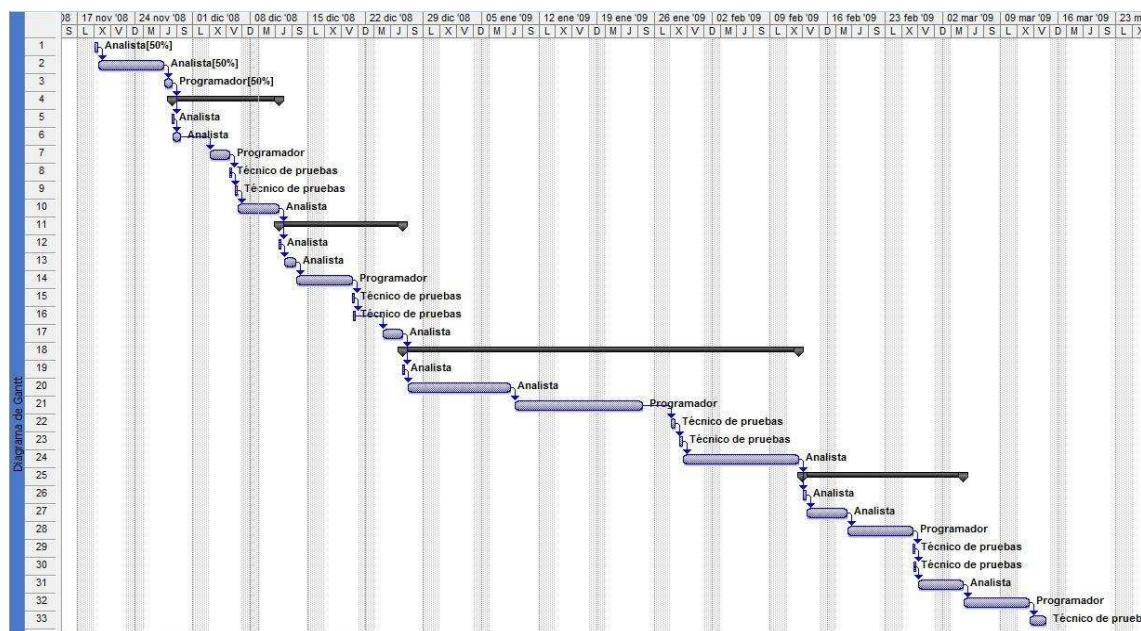


Figura 3. Diagrama de Gantt del proyecto

3.13. Conclusiones

Con todo lo mencionado anteriormente, a priori, parece que el proyecto es viable al 100%. Esto supondrá un beneficio no solo para la asociación implicada sino también a nivel personal para el autor de este proyecto, ya que de esta forma podrá valorar la asimilación de los conocimientos que se han ido adquiriendo a lo largo de la carrera y ampliarlos con el desarrollo de este proyecto y pensar en una proyección de futuro por esta rama de la programación.

Por último, estaría bien decir que la dinamización de páginas Web parece que tiene un gran futuro, ya sea porque empresas quieran llamar la atención de potenciales clientes con esta tecnología o para hacer más agradable la estancia de usuarios en una determinada Web.

4. Análisis de requerimientos

4.1. Introducción

Para obtener un buen resultado y poder predecir los posibles errores que puedan surgir en un futuro es necesario realizar una fase en la que se analicen los requerimientos del software. Esta es la fase de análisis de requerimientos, la fase que nos va a ayudar a comprender cual es el problema o necesidad que se pretende solucionar, es decir, que debe hacer el sistema.

Las diferentes funcionalidades que el sistema debe cumplir tienen que analizarse, es decir, una vez entendido el problema se debe especificar el requerimiento. Cuando todos estos requerimientos se contemplen en este documento, será el momento de validar que los requerimientos especificados reflejan las necesidades del cliente y los objetivos que se pretenden alcanzar.

En esta fase debemos especificar la funcionalidad, restricciones y limitaciones del software, pero lo que se pretende abarcar es la comprensión del problema y su especificación.

Cuando se completa todo este análisis, se tendrá como resultado un documento que nos ayudará a valorar la calidad del software que se pretende diseñar y si cumplirá con las expectativas.

4.2. Registro de usuarios

Los usuarios van a tener la opción de poder registrarse en el sitio Web para poder tener una serie de privilegios. En este caso, se deben poder registrar tres tipos de usuarios: administradores (núcleo), colaboradores y usuarios. El registro consistirá en mostrar al usuario un formulario en el cual se le pedirá el nombre de usuario, la contraseña y la dirección de correo electrónico para confirmar el registro.

Todos estos campos deberán verificarse dinámicamente, es decir, cuando el usuario comience a escribir en alguno de los campos se le mostrará un mensaje indicándole si lo que está escribiendo en dicho campo es correcto.

El nombre de usuario como mínimo deberá contener 3 caracteres, ya sean números o letras, y como máximo 12. No podrá existir 2 usuarios con el mismo nombre de usuario, por este motivo, al procesar el formulario, se deberá consultar la base de datos para verificar la existencia de tal nombre de usuario o no.

La contraseña podrá contener números o letras, pero como mínimo deberá tener 6 caracteres y como máximo 12. No importa si en la base de datos existe una contraseña igual, lo importante es que el usuario se acuerde de ella para poder entrar al sistema.

La dirección de correo electrónico también deberá ser única en la base de datos, por tanto, al procesar el formulario, se deberá consultar la base de datos para comprobar que no exista tal dirección. En la verificación se deberá comprobar que sea una cadena tipo email, como por ejemplo; ejemplo@gmail.com. Se deberá comprobar la existencia del "@" y el "." en la cadena.

Una vez se han verificado correctamente los campos anteriores, el usuario debería leer las condiciones de uso de la página Web y aceptarlas. Si no las acepta, no podrá procesar el registro.

Si hay datos erróneos o repetidos en el formulario, se le mostrará al usuario un mensaje de error indicándole donde está el fallo y se le dará la opción de volver a intentarlo.

Si todo ha resultado correcto, el formulario se procesará, se registrarán los datos del usuario en la base de datos y se le enviará un correo electrónico al email que se proporcione.

Este correo electrónico deberá contener un título de registrado con éxito en la Web y en el cuerpo del mensaje se le dará la bienvenida y los datos de acceso al sistema.

Una vez registrado, con estos datos de acceso el usuario podrá iniciar sesión para poder interactuar un poco más con el sitio.

El inicio de sesión también será un formulario donde se le pedirá al usuario el nombre de usuario y la contraseña de acceso. Estos campos también serán verificados dinámicamente de la misma forma que en el formulario de registro.

Si el usuario introduce un nombre de usuario incorrecto o una contraseña incorrecta, el sistema le informará de tal situación y no le dejará acceder hasta que introduzca un nombre de usuario correcto con su correspondiente contraseña.

En el caso extremo de que el usuario no se acuerde del nombre de usuario o contraseña, tendrá el servicio de recuperar sus datos de acceso.

La recuperación de datos de acceso también se corresponderá a un formulario donde solo se le pedirá al usuario la dirección de correo electrónico para así facilitarle sus datos de acceso mediante este medio.

Si el email introducido para intentar recuperar los datos es incorrecto o no está en la base de datos, el sistema informará de tal situación.

En caso contrario, se le mostrará un mensaje en pantalla informando que en breve recibirá un correo electrónico donde se le facilitará el nombre de usuario y contraseña para poder acceder al sistema, y se procederá a enviar esos datos a la dirección facilitada.

4.3. Perfil de usuarios

Para acceder al perfil de usuario, primero será necesario que un usuario esté registrado e inicie sesión. Si se consigue esto, el usuario tendrá la opción de acceder a su perfil donde se muestra sus datos de usuario y personales. Los datos de usuario al ser obligatorios en el registro se mostrarán, pero los datos personales, como son opcionales, el usuario los podrá llenar si lo desea una vez esté dentro de su perfil.

Los datos de usuario que se mostrarán serán el nombre de usuario, el password, el email, la fecha de registro, el número de noticias publicadas y el número de comentarios realizados en lo que lleva de registro.

Los datos personales que se mostrarán, si el usuario lo desea, serán su nombre, apellido, fecha de nacimiento, profesión, una descripción personal y una foto.

Tanto datos de usuario como personales podrán ser modificados en cualquier momento por el usuario. El único dato que no será modificable será la fecha de registro y el nombre de usuario.

Cuando el usuario acceda a modificar su perfil, se le mostrará en una nueva página un formulario en el que podrá cambiar todos los datos anteriores.

También desde su perfil, el usuario podrá consultar todas las noticias que haya publicado desde su fecha de ingreso, podrá leerlas, modificarlas e incluso eliminarlas. Podrá consultar todos sus comentarios realizados en lo que lleva de registro para poder modificarlos o eliminarlos. Podrá consultar o eliminar sus noticias seleccionadas (favoritos).

Otra funcionalidad importante que tendrá el usuario desde su perfil será poder personalizar tanto el estilo de la página principal como elegir que temas de noticias se visualizarán en dicha página.

Y por último, cada usuario deberá disponer de un blog para que pueda recopilar cronológicamente textos o artículos, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente, es decir, un lugar donde el usuario tenga total libertad para publicar lo que desee.

4.4 Personalización de la página principal

Un usuario registrado tendrá opción de cambiar la página principal en cuanto a estilos se refiere y temas de noticias a mostrar. Se tratará de una interfaz que se le mostrará al usuario donde podrá elegir el color de la página y las categorías de noticias que le interese ver.

Por otro lado, en la página principal, las noticias deben ser dinámicas, es decir, el usuario las debe poder mover a su antojo en el espacio de noticias.

4.5 Blog

Un usuario registrado también tendrá la opción de actualizar su blog. De entrada, deberá ser una página Web que presente al usuario la opción de introducir un tema, un mensaje y poder añadir alguna foto relacionada con el mensaje. Podrá incluir tantos temas como quiera, que se guardarán cronológicamente, pero que al acceder al blog se mostrará el último.

Los demás usuarios registrados podrán acceder a los blogs de los demás usuarios y podrán comentarle sus temas.

En cualquier momento el usuario podrá acceder a sus antiguos temas para recordarlos o también podrá comentar sus temas o editarlos e incluso eliminarlos.

4.6 Administración de usuarios

Se tratará de una página privada a la cual solo podrán acceder aquellos usuarios que son administradores y que podrán gestionar o controlar todo lo referente a los usuarios.

El control consistirá en tener una lista de usuarios donde se muestre todo lo que van realizando a medida que pasa el tiempo, sancionar aquellos usuarios que no respetan las condiciones de uso y borrar noticias o comentarios de usuarios sancionados, en fin, poder eliminar un usuario sancionado o castigarlo restringiendo el uso del sistema.

5. Herramientas de desarrollo

5.1. Introducción

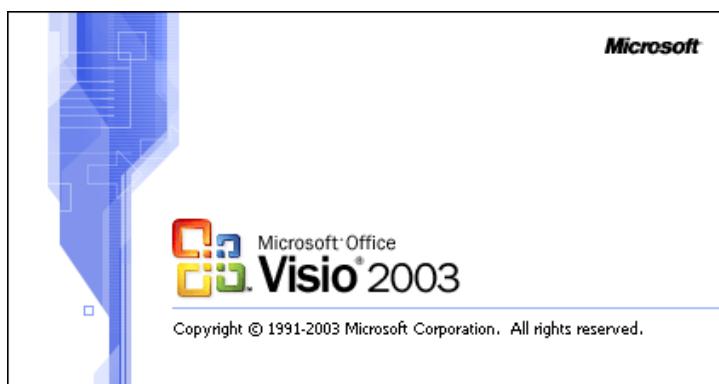
Las principales tecnologías para diseñar este proyecto son utilizar herramientas UML, en especial, el programa Microsoft Visio para la realización de los diferentes diagramas de diseño del sistema como son: los diagramas de casos de uso, diagramas de actividad, diagramas de secuencia y diagrama de clases.

Para el diseño de la base de datos se va a utilizar el programa BDDesigner4 con el cual se generará un código SQL que se importará a PHPMyAdmin (Programa administrador de la base de datos).

En cuanto a la implementación se va a utilizar principalmente el lenguaje de programación orientado al servidor PHP, la base de datos MySQL y JavaScript.

5.2. Herramientas para el diseño

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación. Existen varios programas de este estilo, pero se elige este por su facilidad de uso y por estar respaldado por Microsoft.



Buscando por Internet un programa con el que se pudiera diseñar una base de datos de forma gráfica y que seguidamente te generara el código SQL para su posterior uso, se encontró BDDesigner4, un programa muy útil, sencillo y accesible con el cual se puede diseñar una base de datos de forma gráfica que se guarda en formato XML, pero que te permite convertir a SQL, conversión necesaria para luego importarlo al servidor.

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas webs, utilizando Internet.



Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL y está incorporado en el paquete XAMPP, que más adelante se hablará de ello.

En PhpMyAdmin es donde vamos a importar el archivo .sql que generará el programa BDDesigner4. En PhpMyAdmin se puede diseñar bases de datos, pero hay problemas para hacer las relaciones entre tablas. Es decir, para tablas sencillas con PhpMyAdmin ya tendríamos suficiente, pero como en este proyecto se necesita una base de datos relacionada, se tuvo que buscar otra alternativa.

5.3. Herramientas para la implementación

Se van a elegir estas tecnologías (PHP y MySQL) para la implementación porque este proyecto trata de ampliar uno ya existente el cual utiliza estas tecnologías y porque son idiomas que presentan ciertas dificultades que la gente no sabe apreciar y con ello se quiere demostrar que se puede aprender de una manera sencilla sin ningún tipo de base. Para realizar el trabajo se va a empezar casi desde cero, teniendo conocimientos muy básicos.

Actualmente hay muchas webs desarrolladas con PHP, tales como YouTube ©, El Rellano © u otras más sencillas como Fotolog ©, son páginas que consiguen millones de visitas al día. Conociendo este lenguaje de

programación todos podríamos tener una Web parecida a esas y así poder formar nuestra propia empresa por Internet.

Antes de explicar PHP hace falta explicar que para hacer webs PHP a diferencia de HTML, se necesita un servidor para ir mirando cómo queda la Web, no se puede mirar directamente en el ordenador a no ser que tengamos instalado un servidor.

PHP es un lenguaje de programación interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web dinámico y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida.



A diferencia de otros lenguajes de programación como el HTML que muestra la Web tal y como es para todo el mundo, el PHP muestra una Web exclusiva para cada usuario, PHP es capaz de reconocer el usuario y hacer una acción u otra según de quien se trate. Otra diferencia notable es que el HTML se ejecuta en el navegador de cada persona, mientras que el PHP al tener que realizar acciones se ejecuta directamente en el servidor para luego poder reproducirlo más rápidamente en nuestro navegador.

PHP es el lenguaje del lado del servidor ampliamente disponible en la mayoría de las páginas Web. En una estadística del año 2005 se comprobó que más de 22 millones de dominios lo usaban.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Es un lenguaje que permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Es un lenguaje multiplataforma, tiene una capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones). Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Permite las técnicas de Programación Orientada a Objetos en versiones superiores a la 4.0.

PHP es Open Source que podría traducirse como código fuente abierto. Esto significa que cualquier persona puede emplear el software libremente (incluso para su uso comercial).

Como principales beneficios de que PHP sea código Open Source, destacan la existencia de una gran cantidad de información fiable en la red, la rápida corrección de los errores que puedan aparecer (cualquiera que detecte un error puede hacerlo saber al resto de la comunidad y reparar el código) y la inmediata incorporación de nuevas funcionalidades.

Se puede decir que la mayor parte del proyecto está escrito en PHP e incluso se utiliza alguna librería externa como por ejemplo; para reducir el número de registros por páginas en las noticias publicadas, noticias seleccionadas, comentarios realizados y entradas publicadas de cada usuario se ha utilizado Paginator que es un script de paginación desarrollado en PHP para dividir resultados de consultas extensas a una base de datos MySQL, en grupos de "n" registros por página. Como ejemplo de paginación podemos ver la página de Google, donde se dice que hay 13,235 resultados, pero aparecen divididos en varias páginas de 20 resultados cada una.

Paginator genera, además, una barra de navegación que contiene los enlaces a las diferentes páginas (<<anterior 1 2 3 4 siguiente>>).

La característica principal de este script es su fácil utilización, ya que la forma de paginar es muy conocida y utilizada. Lo que se ha pretendido con este script es ordenarlo de tal forma que no sea necesario entenderlo ni editarlo, sino simplemente incluirlo después de definir al menos una variable.

Otro motivo importante de elección de estas tecnologías es la capacidad de cambio de plataforma, puede desarrollar en su ordenador personal y desplegar el mismo código exactamente en el servidor de producción, incluso si se está ejecutando en sistemas operativos distintos.

MySQL también tiene la fama de ser la base de datos de código abierto más popular con más de 6 millones de instalaciones activas. Además es un sistema de gestión de base de datos relacional, multihilo y multiusuario que interacciona muy bien con PHP.



Como PHP es un lenguaje del lado del servidor, necesita funcionar en conexión con un servidor Web. A menudo se encuentra en combinación con Apache, el software que se ejecuta en dos de cada tres servidores en el mundo.



Apache es un software (libre) servidor HTTP de código abierto que implementa el protocolo HTTP y la noción de sitio virtual.

Apache, PHP y MySQL se ejecutan en los sistemas operativos Windows, Mac OS X y Linux. Esta flexibilidad es una de las mayores ventajas de desarrollar con PHP y MySQL.

Configurar estas tres tecnologías por separado es perder el tiempo y la cabeza, y más aún disponiendo de herramientas como XAMPP, que ya incorpora las tres tecnologías debidamente configuradas.



XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

En cuanto a coste, todas son tecnologías gratuitas y esto no quiere decir que sea tecnología desprestigiada. Se pueden conseguir grandes aplicaciones con su uso.

En cuanto a JavaScript, cabe decir que es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C, que en este caso es código que se ejecuta en el cliente.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Por ejemplo, cada campo obligatorio de los formularios será verificado dinámicamente mediante JavaScript. Para hacerlo se ha incluido una librería muy útil llamada “livevalidation²” obtenida de <http://www.livevalidation.com/>, pero adaptada a nuestras necesidades. Lo que hacemos es incrustar código JavaScript en cada campo obligatorio donde creamos una nueva instancia de LiveValidation pasándole como parámetro el atributo “id” del tag <input> correspondiente.

Entonces con esta instancia se comprueba, por ejemplo, la presencia de datos en el campo y la longitud en el caso del nombre de usuario y contraseña.

La principal ventaja de la utilización de esta librería es que para que se procese el formulario tienen que estar todos los campos correctamente rellenados y, con ello, posteriormente no tenemos que controlar si tienen valor las variables o si no tienen la longitud adecuada y, por tanto, la codificación de la clase tipo control que le sigue se hace más sencilla y entendible.

² **LiveValidation** es una utilidad en Javascript que permite validar los campos o entradas de textos de los formularios HTML (textarea, input text, input check) en tiempo real.

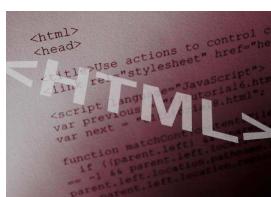
Si un usuario intentase procesar el formulario con algún campo que no cumpla con los requerimientos impuestos por el cliente, el formulario no se procesa e informa sobre el error.

Para dar formato a los textos de las noticias, comentarios o entradas al blog se ha utilizado Tinymce, que es una aplicación prefabricada muy útil para usar en blogs, foros, páginas Web etc. Es una toolbar que se coloca en los textareas con botones a través de los cuales podemos dar formato al texto, mediante un sencillo script.

Para que las noticias puedan moverse por la pantalla al gusto del usuario se ha utilizado una librería llamada jQuery. Se trata de una biblioteca o Framework 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 tecnología AJAX a páginas Web.

JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Tradicionalmente, se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.



HTML corresponde a las siglas HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. 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.

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.



La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.



Para procesar todos los archivos es necesario disponer de un buen editor de textos. Se ha elegido Dreamweaver 8 por el hecho de ser más que un editor y el poder procesar archivos de todo tipo: HTML, ColdFusion, PHP, ASP, JavaScript, XML, CSS, entre otros.

Dreamweaver permite editar las páginas mientras se visualizan tal y como se verán en el navegador. También permite editar el código HTML directamente, mediante un editor propio, o con el editor de texto que se desee. Además, y para aquellos elementos de la página que no son visibles (como los scripts JavaScript, etiquetas meta, etc.), incorpora iconos en la pantalla que los describen y que permiten acceder a ellos de forma rápida y editar sus propiedades fácilmente.



WordPress es un sistema de gestión de contenido enfocado a la creación de blogs (sitios Web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL. WordPress fue creado a partir del desaparecido b2/cafelog y se ha convertido junto a Movable Type en el CMS más popular de la blogosfera. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

Otro motivo a considerar sobre su éxito y extensión, es la enorme comunidad de desarrolladores y diseñadores, que se encargan de desarrollarlo en general o crear plugins y temas para la comunidad (que ascendían a 2.524 y 1.320 respectivamente en julio de 2008).

FileZilla es un cliente FTP, gratuito, libre (GNU) y de código abierto. Sustenta FTP, SFTP y FTP sobre SSL. Inicialmente sólo diseñado para funcionar bajo Windows, desde la versión 3.0.0, gracias al uso de wxWidgets, es multiplataforma, estando disponible además para otros sistemas operativos, entre ellos Linux, FreeBSD y MacOS X.

Las principales características son el Site Manager (Administrador de sitios), Message Log (Registro de mensajes), y Transfer Queue (Cola de transferencia).

6. Diseño

6.1. Introducción

En esta fase se van a aplicar diferentes técnicas y principios con el propósito de definir un sistema con suficiente detalle como para poder realizarlo físicamente. Esta fase es el núcleo técnico del proceso de ingeniería del software y se aplica independientemente del modelo de desarrollo a utilizar.

El objetivo principal es producir un modelo o representación del sistema que pueda ser utilizado, en una fase posterior, con el fin de implementarlo.

Cada uno de los subproblemas o elementos que se analizan en la fase anterior (análisis de requerimientos) proporciona la información necesaria para crear un modelo de diseño.

Para modelar el sistema, desde el concepto hasta los elementos ejecutables, se va a utilizar UML, que es una metodología para visualizar, especificar, construir y documentar los elementos de un sistema que involucra una gran cantidad de software desde una perspectiva orientada a objeto.

El UML permite modelar diferentes vistas o visiones de un problema y su solución, dependiendo de lo que interese representar en cada ocasión. A cada vista le corresponde uno o más diagramas.

6.2. Descripción de los usuarios del sistema

Antes de empezar con el diseño se va a describir quienes van a ser los actores o usuarios de la aplicación, que intervendrán en los diferentes casos de uso. Se citan por orden de jerarquía de mayor a menor nivel dentro del sistema.

Nombre	Descripción
Administrador	Es la persona o personas que van a tener el control de la página Web. Podrán administrar todo el contenido.
Colaborador	Son las personas que, a parte de ser usuarios, colaboran económicamente con la asociación.

Usuario	Son las personas que se van a registrar para interactuar aún más con el sitio Web (amigos de Kaos).
---------	---

6.3. Diagrama de casos de uso

Es el diagrama que corresponde a la visión del usuario, muestra el comportamiento del sistema des de el punto de vista del usuario, es decir, muestra qué podrán hacer los usuarios en el sistema. Se va a especificar el comportamiento deseado del sistema, se va a representar los requerimientos funcionales del sistema.

A continuación se muestra el diagrama de casos de uso de la aplicación.

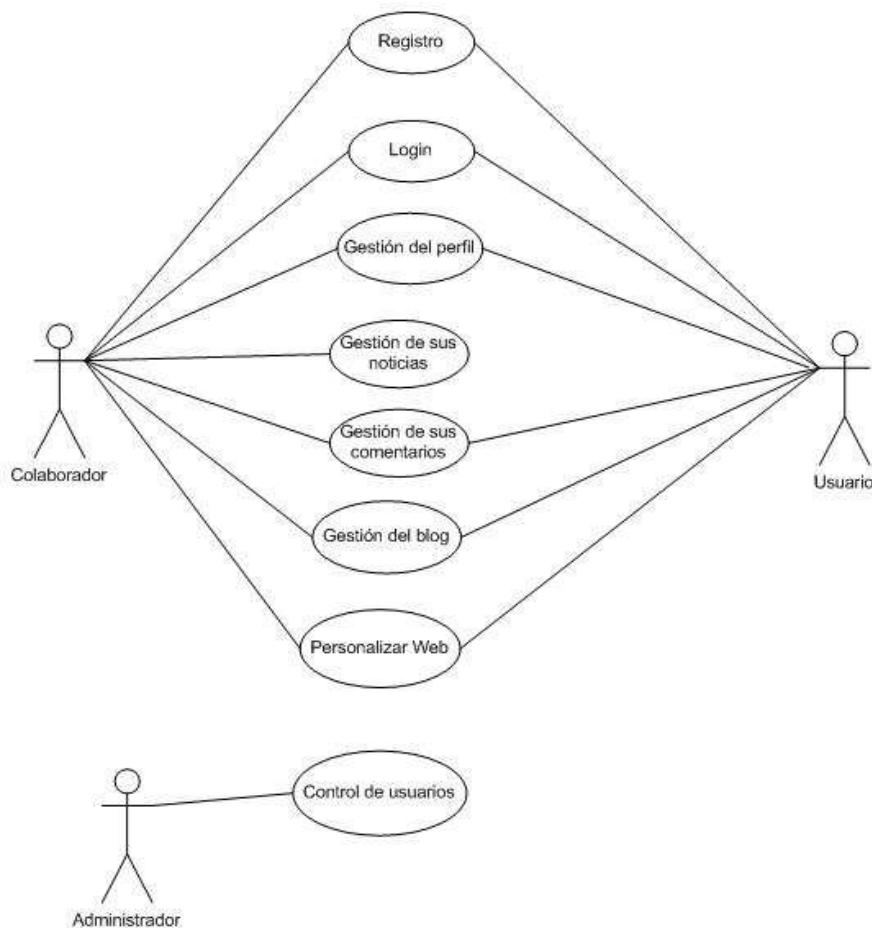


Figura 4. Diagrama de casos de uso del sistema.

El administrador no solo tiene el control de usuarios, sino que también puede realizar todos los demás casos de uso.

6.4. Especificación de los casos de uso

6.4.1. Registro

Descripción: El registro de usuarios consistirá principalmente en mostrar al usuario un formulario donde rellene un nombre de usuario, password y email con el objetivo de poder pertenecer a la comunidad de usuarios.

Actores: Todos (administrador, colaborador y usuario).

Precondición: Estar en la página principal de la Web.

Flujo principal: El caso de uso comenzará cuando un usuario quiera registrarse. Pulsa tal hipervínculo y se le mostrará una nueva página donde aparece el formulario de registro. El sistema espera a que el usuario introduzca nombre de usuario, contraseña y email.

- Recogida de datos de entrada
 1. Nombre de usuario
 2. Password
 3. Email
- Verificación de datos introducidos
 1. Verificar que el nombre de usuario sea mayor a tres caracteres y menor a doce.
 2. Verificar que el password tenga más de seis caracteres y menos de doce.
 3. Verificar que el email tenga formato email, es decir, sea una cadena que contenga @ y un ‘.’
 4. Verificar la aceptación de las condiciones de uso.

- Validación de datos de usuario

Comprobar la existencia del nombre de usuario y email introducido por el usuario en la base de datos de la comunidad una vez procesado el formulario.

- Insertar datos en la base de datos (registro)

Si la verificación y la validación se han llevado a cabo con éxito se procederá a insertar los datos introducidos en la base de datos.

- Confirmación del registro.

Enviar un email al usuario registrado para darle la bienvenida y tenga sus datos de entrada al sistema disponibles.

Flujo alternativo: No se procesará el formulario en los siguientes casos:

1. El nombre de usuario es incorrecto.
2. El password es incorrecto.
3. El email es incorrecto.
4. No se ha aceptado las condiciones de uso.

Si se procesa el formulario se debe comprobar en la base de datos:

1. La duplicación del nombre de usuario.
2. La duplicación del email.

Si el nombre de usuario y el email no están duplicados en la base de datos de la organización, se procederá a la inserción, registro del usuario y confirmación del registro. En caso contrario, se le mostrará un mensaje de error indicando al usuario que concepto es el que está duplicado.

Postcondición: Login

Prioridad: Alta

6.4.2. Login

Descripción: Consistirá principalmente en mostrar un formulario para entrar al sistema.

Actores: Todos (administrador, colaborador y usuario).

Precondición: Un usuario se debe registrar.

Flujo principal: El caso de uso comenzará cuando un usuario quiera iniciar sesión. Pulsa tal hipervínculo y se le mostrará una nueva página donde aparecerá el formulario de entrada. El sistema espera a que el usuario introduzca nombre de usuario y contraseña.

- Recogida de datos de entrada

1. Nombre de usuario.
2. Password.

- Verificación de datos introducidos

1. Verificar que el nombre de usuario sea mayor a tres caracteres y menor a doce.
2. Verificar que el password tenga más de seis caracteres y menos de doce.

- Validación datos de usuario

1. Comprobar la existencia del nombre de usuario introducido en la base de datos de la organización.
2. Comprobar la existencia del password asociado a este nombre de usuario.

Flujos alternativos: No se procesará el formulario en los siguientes casos:

1. El nombre de usuario es incorrecto.
2. El password es incorrecto.

Si se procesa el formulario se procederá a la validación de datos del usuario. Si algún dato es incorrecto se le mostrará por pantalla.

Postcondición: Entrada al sistema.

Prioridad: Alta

6.4.3. Gestión del perfil

Descripción: Consistirá principalmente en la visualización de un perfil de usuario en concreto.

Actores: Usuario y administrador.

Precondición: El usuario debe hacer login.

Flujo principal: El caso de uso comenzará cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo y se le mostrará una nueva página donde aparecerá su perfil de usuario.

- Visualización del perfil.
 1. Datos de usuario (obligatorios). Corresponden a nombre de usuario, contraseña, email, fecha de registro y fecha de última modificación, además del número de noticias y comentarios realizados. El número de noticias y comentarios realizados se irá rellenando dinámicamente en función de lo que vaya haciendo el usuario con la cuenta creada.
 2. Datos personales (opcionales). Corresponden a nombre, apellido, fecha de nacimiento, profesión, descripción personal y foto.
- Modificación del perfil.

Tanto datos de usuario como personales podrán ser modificados y actualizados en cualquier momento accediendo a un formulario donde cada campo será verificado y validado para proceder a la modificación.

Flujo alternativo: En cuanto a la visualización, de entrada tanto datos personales como noticias y comentarios realizados van a estar vacíos.

En la modificación del perfil se debe tener especial cuidado en los cambios de nombre de usuario y email, ya que estos no pueden estar duplicados en la base de datos. Por tanto, si el usuario cambia estos datos, el sistema deberá comprobar otra vez si existen en la base de datos.

Prioridad: Alta

6.4.4. Gestión de noticias

Descripción: Consistirá en tener un apartado en el cual un usuario en concreto podrá leer, modificar y eliminar todas las noticias que ha publicado.

Actores: Administrador y colaborador.

Precondición: El usuario debe hacer login.

Flujo principal: El caso de uso comenzará cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo donde habrá un menú en el cual podrá acceder a la visualización de sus noticias.

- Visualización de todas las noticias publicadas

De entrada, se mostrará una lista con todas las noticias publicadas donde se mostrará el título, la fecha de publicación, la fecha de modificación y el número de comentarios que se hayan realizado sobre la noticia. En cualquier momento el usuario tendrá la opción de obtener más información sobre las noticias que deseé, editarlas o eliminarlas.

- Modificación de noticias

Se tratará de un formulario en el cual aparecerá la noticia que se desea editar para después enviarla de nuevo modificada.

- Eliminación de noticias

El usuario podrá eliminar aquellas noticias que ya no considere importante tenerlas presente. Se tratará de un botón que tenga como función eliminar la noticia seleccionada.

Flujos alternativos: Si el usuario todavía no ha publicado ninguna noticia y desea acceder a sus noticias, se le mostrará un mensaje informándole de que todavía no ha publicado ninguna noticia y se le dará la opción de poder publicar una nueva noticia. Además, lógicamente no podrá editar ni borrar una noticia desde la gestión de noticias.

Prioridad: Alta

6.4.5. Gestión de comentarios

Descripción: Consistirá en tener un apartado en el cual un usuario en concreto podrá leer, modificar y eliminar todos sus comentarios realizados.

Actores: Todos (administrador, colaborador y usuario).

Precondición: El usuario debe hacer login.

Flujo principal: El caso de uso comienza cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo donde habrá un menú en el cual podrá acceder a la visualización de sus comentarios.

- Visualización de todos los comentarios realizados

Se le mostrará el título, el mensaje y la fecha de realización o última modificación de cada comentario asociado a una noticia.

- Modificación de comentarios

Se tratará de un formulario en el cual aparecerá el comentario que se desea editar para después enviarlo de nuevo modificado.

- Eliminación de comentarios

El usuario podrá eliminar aquellos comentarios que ya no considere importante tenerlos presente. Se tratará de un botón que tenga como función eliminar el comentario.

Flujos alternativos: Si el usuario todavía no ha realizado ningún comentario y desea acceder a sus comentarios, se le mostrará un mensaje informándole de

que todavía no ha realizado ningún comentario. Además, lógicamente no podrá editar ni borrar un comentario desde la gestión de comentarios.

Prioridad: Alta

6.4.6. Gestión del blog

Descripción: Consistirá en poder publicar entradas que posteriormente van a poder ser leídas y comentadas por otros usuarios.

Actores: Todos (administrador, colaborador y usuario).

Precondición: El usuario debe hacer login.

Flujo principal: El caso de uso comienza cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo donde habrá un menú en el cual podrá acceder a la visualización de su blog.

- Visualización del blog

El usuario, inicialmente podrá ver en la página principal del blog su última entrada publicada pero también podrá ver todas sus entradas y los comentarios que otros usuarios hayan realizado.

- Inserción de entradas

El usuario podrá introducir entradas sobre cualquier tema, que posteriormente van a poder ser comentadas por otros usuarios y el autor de dicha entrada va a poder responder a estos comentarios. Las entradas se mostrarán en orden cronológico y en portada del blog aparecerá la última entrada. Las entradas anteriores podrán ser consultadas en cualquier momento para poder modificarlas o eliminarlas.

- Modificación de entradas

Se tratará de mostrar un formulario con la entrada que se desea editar para después enviarla de nuevo modificada.

- Eliminación de entradas

El usuario podrá eliminar aquellas entradas que ya no considere importante tenerlas presente. Se tratará de un botón que tenga como función eliminar la entrada con sus respectivos comentarios.

Flujos alternativos: Si el usuario todavía no ha realizado ninguna entrada y desea acceder al blog, se le mostrará el formulario para llenar la primera entrada. Además, lógicamente no podrá editar ni borrar una entrada desde el blog si no hay entradas.

Prioridad: Alta

6.4.7. Personalización de la página principal

Descripción: Consistirá en tener un apartado en el cual un usuario en concreto podrá cambiar el estilo y elegir las noticias que desea visualizar en la página principal de la Web.

Actores: Todos (administrador, colaborador y usuario).

Precondición: El usuario debe hacer login.

Flujo principal: El caso de uso comenzará cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo donde habrá un menú en el cual podrá acceder a personalizar la página principal.

- Cambiar estilo

Se tratará de mostrar al usuario una interfaz en la cual podrá elegir distintos atributos de estilo para aplicar a la página principal.

- Selección de categorías

Se tratará de un formulario en el cual aparecerán checkbox's con todas las categorías de noticias, donde el usuario podrá seleccionar aquellas categorías que quiera ver en la página principal.

Flujos alternativos: Si el usuario decide no seleccionar ninguna categoría, se mostrará una página principal vacía.

Prioridad: Baja

6.4.8. *El control de usuarios*

Descripción: Consistirá en tener una lista para controlar las acciones que hacen los usuarios y ver estadísticas sobre usuarios.

Actores: Administrador

Precondición: El usuario (administrador) debe hacer login.

Flujo principal: El caso de uso comenzará cuando un usuario tipo administrador inicia sesión y accede al control de usuarios pulsando tal hipervínculo.

- *Visualización de todos los usuarios registrados*

Consistirá en una tabla en la cual se podrá ver información sobre los usuarios registrados y acceder a sus contenidos.

- *Eliminar usuarios*

Cuando un usuario interactúa con el sitio Web de una forma inadecuada, es importante poder restringirlo del sitio para que no siga perturbando con su comportamiento a los demás usuarios.

- *Eliminación o modificación de noticias*

Podrá eliminar o modificar aquellas noticias de un usuario que no se consideren como tal.

- *Eliminación o modificación de comentarios*

Podrá eliminar o modificar aquellos comentarios de un usuario que no se consideren oportunos o no impliquen respeto.

Flujos alternativos: Si no hay usuarios registrados, se mostrará un mensaje por pantalla informando que no hay usuarios registrados, pero esto prácticamente nunca sucederá porque en la lista de usuarios al menos aparecerá el usuario administrador que está consultando dicha lista.

Prioridad: Alta

6.4.9. Favoritos

Descripción: Consistirá en tener un apartado en el cual un usuario en concreto podrá guardar aquellas noticias publicadas en Kaos de su interés.

Actores: Todos (administrador, colaborador y usuario).

Precondición: El usuario debe hacer login y entrar en su perfil.

Flujo principal: El caso de uso comenzará cuando un usuario inicia sesión y accede a su perfil pulsando tal hipervínculo donde habrá un menú en el cual podrá acceder a la visualización de sus noticias seleccionadas (favoritas).

- Visualización de todas las noticias seleccionadas

De entrada, se mostrará una lista con todas las noticias seleccionadas donde se mostrará el título, resumen, la fecha de publicación, la fecha de modificación y el número de comentarios que se hayan realizado sobre la noticia. En cualquier momento el usuario tendrá la opción de obtener más información sobre las noticias que desee, o eliminarlas cuando ya no las considere como favoritas.

- Eliminación de noticias

El usuario podrá eliminar aquellas noticias seleccionadas que ya no considere importante tenerlas guardadas. Se tratará de un botón que tenga como función eliminar la noticia seleccionada.

Flujos alternativos: Si el usuario todavía no ha seleccionado ninguna noticia y desea acceder a sus favoritos, se le mostrará un mensaje informándole de que todavía no ha agregado a favoritos ninguna noticia.

Prioridad: Alta

6.5. Diagrama de actividades

6.5.1. Introducción

Un diagrama de actividades es esencialmente un diagrama de flujo de acciones que muestra la actividad de un caso de uso a lo largo del tiempo. Es un caso especial de diagrama de estados en el que los estados son estados de actividad y las transiciones se disparan por el fin de las actividades en el estado de origen.

El diagrama de actividades permite seleccionar el orden en que se trabajará. Esto es, simplemente dicta las reglas esenciales de secuenciación que se tiene que seguir. Generalmente modelan los pasos de un algoritmo.

A continuación para cada caso de uso vamos a mostrar su diagrama de actividades correspondiente.

6.5.2. Registro de usuarios

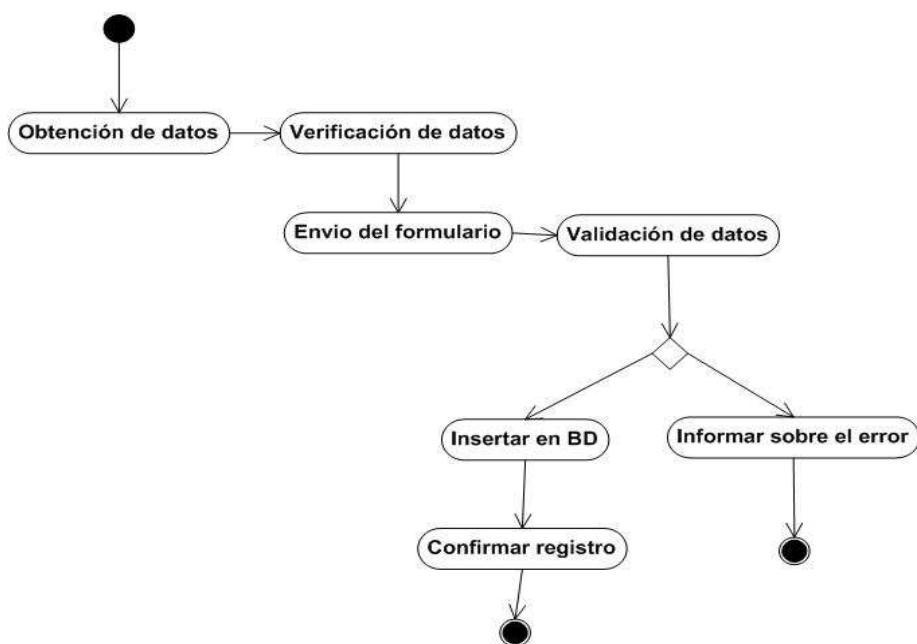


Figura 5. Diagrama de actividades del caso de uso Registro.

Tal como muestra la figura 5, el sistema va a obtener mediante un formulario de registro los datos que introduzca el usuario dinámicamente, verificándolos y si una vez lo que haya introducido en (nombre de usuario, contraseña y email)

es correcto, se podrá enviar el formulario y seguidamente se validarán los datos comprobando la existencia del nombre de usuario y email en la base de datos para no permitir la duplicación de estos. Si el nombre de usuario o email ya existen en la base de datos, el sistema informará que ha habido un error. En caso contrario, se inserta el nuevo usuario en la base de datos y se confirma el registro.

6.5.3. Login

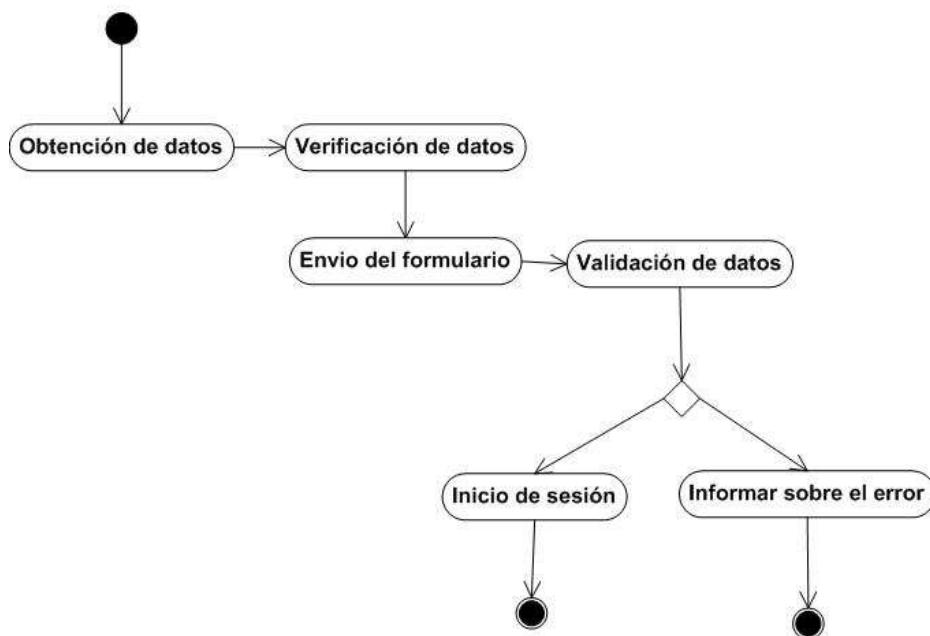


Figura 6. Diagrama de actividades del caso de uso Login.

Tal como muestra la figura 6, el sistema va a obtener mediante un formulario de inicio de sesión los datos que introduzca el usuario dinámicamente, verificándolos y si una vez lo que haya introducido en (nombre de usuario y contraseña) es correcto, se podrá enviar el formulario y se validarán los datos comprobando la existencia de nombre de usuario y password en la base de datos para permitir la entrada o no. Si el nombre de usuario o contraseña son incorrectos, ya sea porque no existen en la base de datos o porque la contraseña no corresponde al nombre de usuario o al revés, el sistema informará que ha habido un error, en caso contrario, se entrará al sistema iniciando sesión.

6.5.4. Gestión del perfil

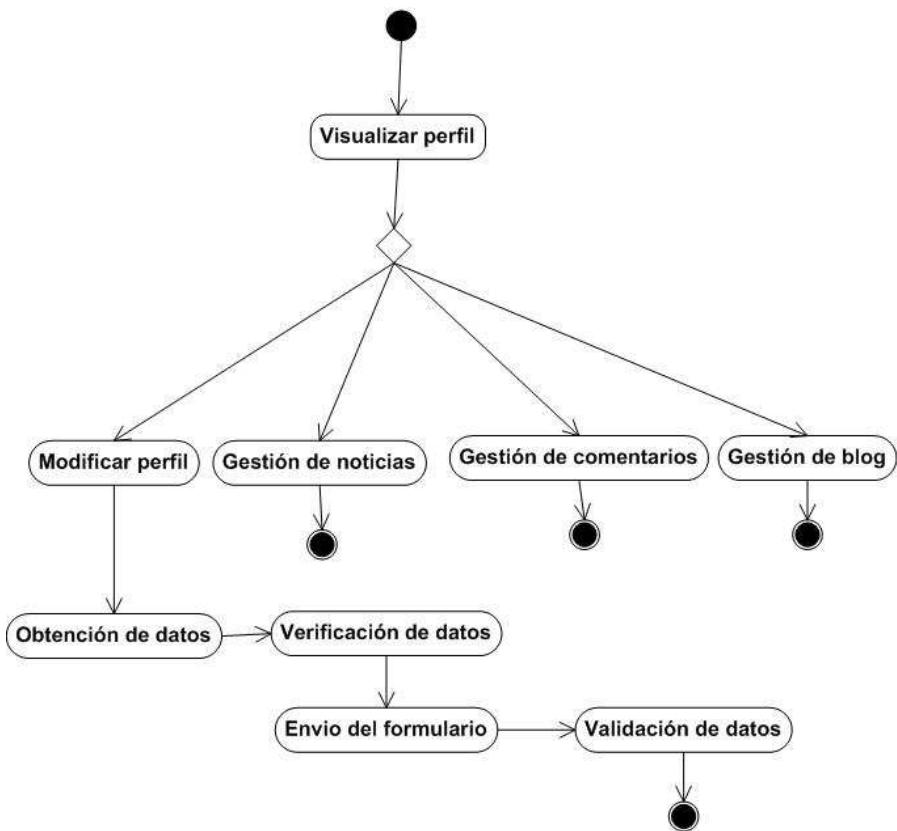


Figura 7. Diagrama de actividades del caso de uso Gestión de perfil.

Cuando un usuario consiga hacer login, si observa la figura 7, ahora tendrá la opción de visitar su perfil, visualizándolo. En esa misma página podrá modificar dicho perfil accediendo a un formulario de modificación donde el sistema obtendrá los nuevos datos que vaya introduciendo el usuario dinámicamente, verificándolos y si los datos introducidos son correctos, se procederá al envío del formulario y su validación, que corresponderá principalmente a la comprobación de nombre de usuario y email para no permitir su duplicación.

Desde su perfil, un usuario también tendrá la opción de acceder a la gestión de noticias, la gestión de favoritos, la gestión de comentarios, personalización del sitio Web y la gestión de blog.

6.5.5. Gestión de noticias

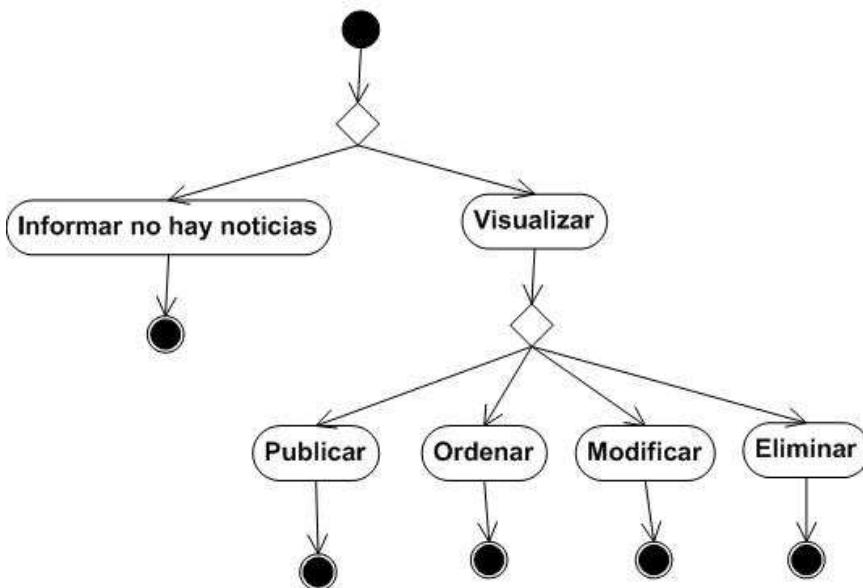


Figura 8. Diagrama de actividades del caso de uso Gestión de noticias.

La gestión de noticias (figura 8) corresponderá principalmente a la visualización de una lista de noticias que haya publicado el usuario en cuestión, donde cada una de las noticias podrá ser modificada mediante un formulario de modificación o eliminada tanto de la base de datos como de la lista de noticias. También aquí tendrá la opción de añadir o publicar una nueva noticia.

El formulario de modificación corresponderá a mostrar el mismo formulario de publicación de una noticia pero rellenado con la noticia que se desee editar. Si se procesa el formulario se procederá a la actualización de la noticia en la base de datos y en la lista de noticias.

Una acción útil, cuando un usuario ha publicado muchas noticias y desea encontrar una en concreto, es el poder ordenar las noticias según las fechas de modificación o creación y por título en orden ascendente o descendente.

Si un usuario accede a la gestión de noticias y todavía no ha publicado ninguna noticia, el sistema le informará sobre ello y le dará la opción de poder publicar una nueva noticia.

6.5.6. Gestión de favoritos

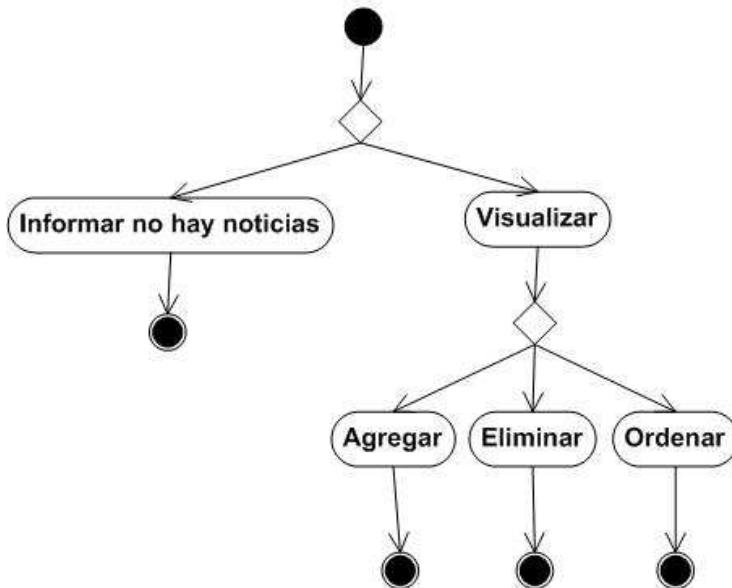


Figura 9. Diagrama de actividades del caso de uso Gestión de favoritos.

La gestión de favoritos o noticias seleccionadas por un usuario (figura 9) corresponderá principalmente a la visualización de una lista de noticias que cada usuario, en concreto, haya seleccionado o agregado por interés a través de la página principal.

En cualquier momento, el usuario puede eliminar de sus favoritos aquellas noticias que ya no considere oportuno tenerlas presente.

De la misma forma que en gestión de noticias, si el usuario tiene muchas noticias agregadas a sus favoritos tendrá la opción de poder ordenarlas por fecha de modificación o creación y por título en orden ascendente o descendente.

Si un usuario accede a la gestión de favoritos y todavía no ha agregado ninguna noticia, el sistema le informará sobre ello.

6.5.7. Gestión de comentarios

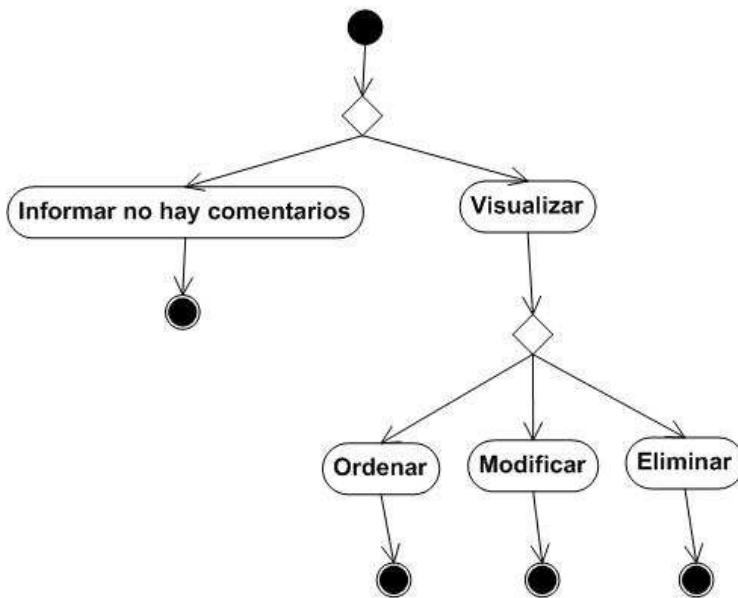


Figura 10. Diagrama de actividades del caso de uso Gestión de comentarios.

La gestión de comentarios (figura 10) corresponderá principalmente a la visualización de una lista de comentarios sobre noticias que haya realizado el usuario en cuestión, donde cada uno de los comentarios podrá ser modificado mediante un formulario de modificación o eliminado tanto de la base de datos como de la lista de comentarios.

El formulario de modificación corresponderá a mostrar el mismo formulario de realización de un comentario pero rellenado con el comentario que se deseé editar. Si se procesa el formulario se procederá a la actualización del comentario en la base de datos y en la lista de comentarios.

De la misma forma que en gestión de noticias, si el usuario tiene muchos comentarios realizados tendrá la opción de poder ordenarlos por fecha de modificación o creación y por título en orden ascendente o descendente.

Si un usuario accede a la gestión de comentarios y todavía no ha realizado ninguno, el sistema le informará sobre ello.

6.5.8. Gestión de blog

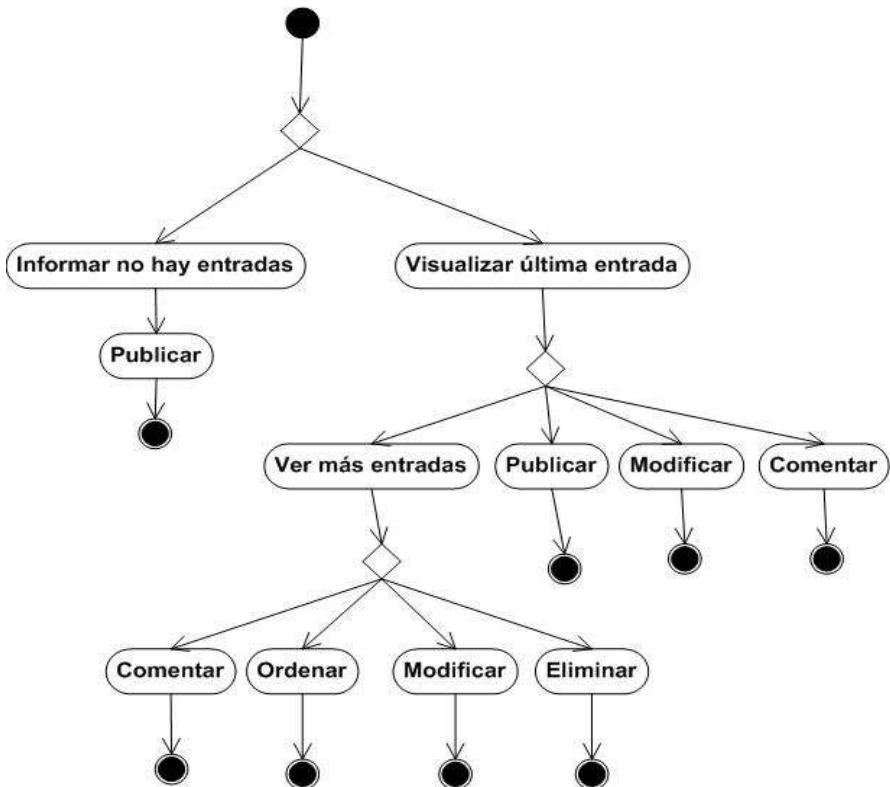


Figura 11. Diagrama de actividades del caso de uso Gestión del blog.

Desde el perfil de usuario, se podrá acceder a la gestión de blog (figura 11), donde se visualizará la última entrada con sus respectivos comentarios, en el caso de que el usuario tenga entradas. Si no tiene entradas, se le informará sobre ello y tendrá la opción de publicar una nueva entrada.

Cuando se visualiza la última entrada, el usuario podrá modificarla, comentarla, publicar otra nueva entrada o consultar todas las entradas que haya publicado anteriormente.

En la página donde se visualizan todas las entradas publicadas, el usuario podrá acceder a cualquier entrada para ver si hay comentarios de otros usuarios y poder responderles. Para eliminar una entrada, solo debe seleccionarla y pulsar el botón eliminar.

En cualquier momento podrá ordenar las entradas por fecha de modificación o creación y por título en orden ascendente o descendente.

6.5.9. Personalización de la página principal

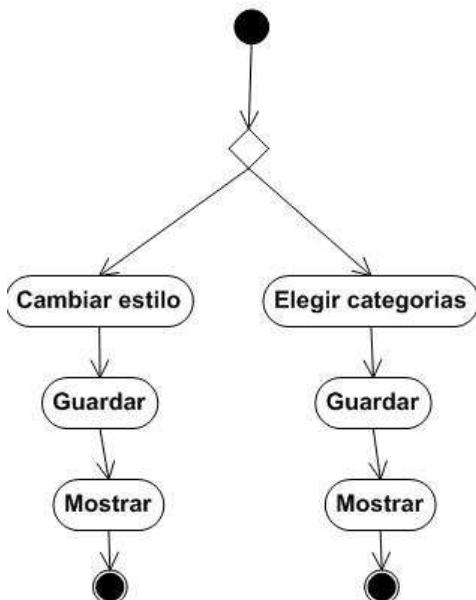


Figura 12. Diagrama de actividades del caso de uso Personalización del sitio Web.

En cuanto a la personalización de la página Web (figura 12), habrá dos posibilidades:

En primer lugar, el usuario podrá cambiar el color de fondo de la página Web accediendo a una interfaz para elegir el color deseado, guardarla y redireccionar a la página principal para mostrar el resultado. Entonces, cada vez que inicie sesión, la página Web se mostrará con el color de fondo guardado previamente por el usuario.

La otra opción es la elección de categorías de noticias a mostrar en la página principal. Se trataría de mostrar un formulario tipo checkbox's donde el usuario pueda seleccionar aquellas categorías que le interese ver cada vez que acceda como usuario a la página principal. Cuando tenga elegidas las categorías de interés, podrá guardarlas para que en la próxima entrada al sitio Web solo vea aquellas noticias de las que está interesado.

6.5.10. El control de usuarios

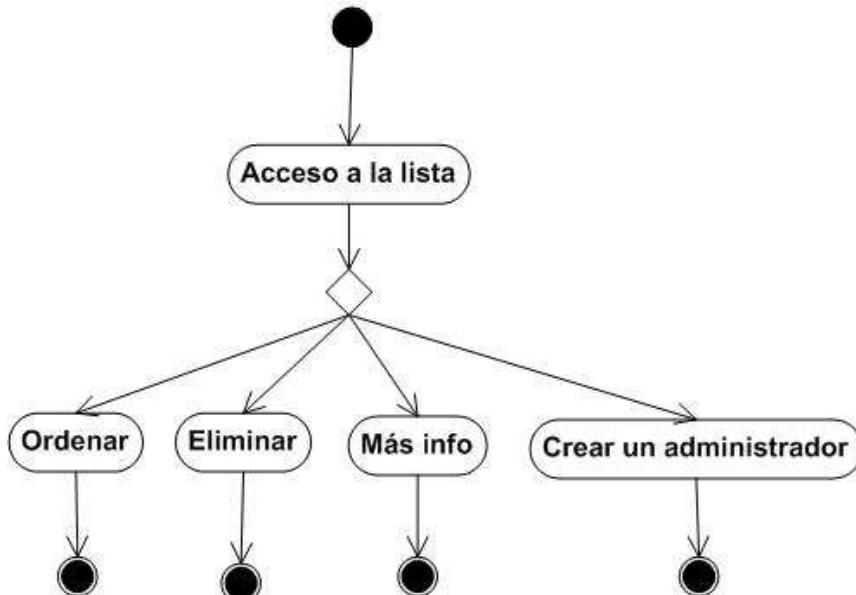


Figura 13. Diagrama de actividades del caso de uso Gestión de usuarios.

Solamente los administradores podrán acceder a esta lista de control o gestión de los usuarios registrados. Será una lista que como mínimo habrá un usuario, el que la consulte, en este caso, el administrador principal.

Esta lista se podrá ordenar por id, nombre de usuario, tipo, estado, fecha de registro, fecha de última modificación o por puntos obtenidos.

Los administradores podrán seleccionar un usuario y eliminarlo, podrán obtener más información sobre un usuario consultando todos los datos que haya llenado de su perfil más sus noticias publicadas, sus favoritos, sus comentarios realizados y visitando su blog. Además de consultar su posición en el ranking de usuarios.

En esta página también se puede acceder al formulario de registro para crear un nuevo administrador.

6.6. Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

Para representar el diagrama de clases del sistema se va a utilizar las clases del modelo de análisis, que son clases estereotipadas que representan un modelo conceptual inicial para los elementos del sistema que tienen responsabilidades y comportamiento. Hay tres tipos de clases de análisis:

- **Comunicación:** Gestionan la interacción entre el sistema y su entorno. Se utilizan principalmente para las interfaces del sistema.
- **Control:** Coordinan los eventos necesarios para realizar el comportamiento especificado en cada caso de uso.
- **Entidad:** Modelan información y su comportamiento. Representan entidades del mundo real o entidades internas necesarias para ejecutar las tareas del sistema. Son independientes de cómo se comunica el entorno con el sistema y a menudo también independientes de la aplicación (transportables a otras aplicaciones).

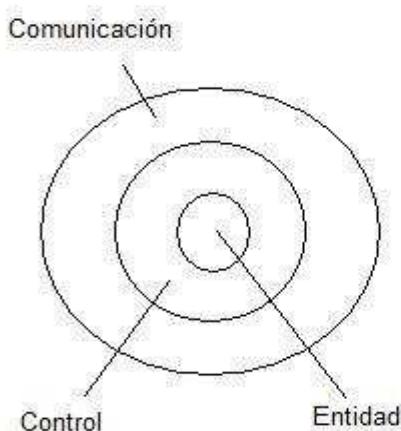


Figura 14. Modelo de 3 capas o de análisis.

En la siguiente figura se muestra el diagrama de clases del sistema:

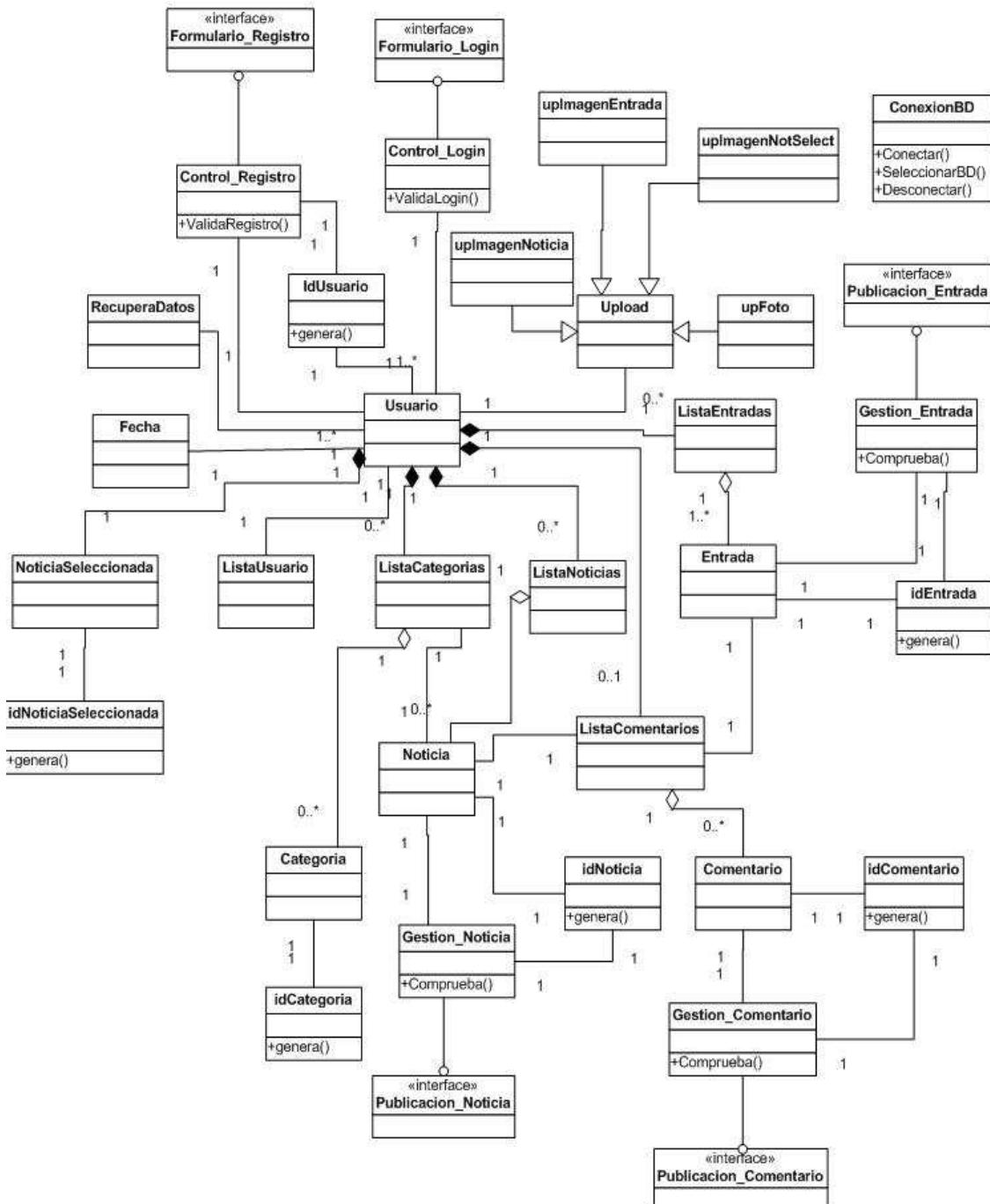


Figura 15. Diagrama de clases.

En la figura 15 se muestran las principales clases de la aplicación. Se trata de un diagrama inicial, así que a medida que se vaya desarrollando el proyecto podría suceder que no se implemente exactamente este diagrama debido a restricciones del software o del lenguaje de desarrollo a utilizar.

La clase central o principal de esta aplicación está claro que es la entidad “Usuario”, ya que todas las demás están muy relacionadas con ella. Contendrá los atributos necesarios para representar un usuario del sistema y las operaciones necesarias para poder mostrar la información de un usuario en concreto, para poder modificar dicha información y eliminarla en el caso de que el usuario decida darse de baja o cuando los administradores decidan que un usuario, por motivos de disciplina, ya no deba pertenecer a la comunidad.

Mediante la entidad “RegistroUsuario” se podrá crear un nuevo usuario. Esta clase se encargará de validar la información que un usuario introduzca en el formulario de registro. En cuanto la información sea validada correctamente, se podrá guardar en la base de datos la información correspondiente al usuario.

La entidad “Login” se encarga de validar la entrada al sistema siempre comprobando que el usuario que quiera entrar esté en la base de datos de la organización.

Teniendo en cuenta el contexto de lo que representa la página Web, otras dos entidades muy importantes son la entidad “Noticia” y la entidad “Comentario”. Son entidades muy parecidas pero, cada una representará lo que su mismo nombre indica. Para representar las noticias y comentarios se van a tener los atributos necesarios y las operaciones básicas: visualizar, insertar, modificar y eliminar.

Cada vez que un usuario desee publicar una noticia o un comentario, se instancia la clase “PublicacionNoticia” o “PublicacionComentario”, según convenga. El objetivo principal de estas clases es comprobar que el nuevo registro a insertar no esté duplicado, generar una id óptimamente e insertar donde corresponda.

Las entidades “ListaNoticias” y “ListaComentarios” son entidades que se considera oportuno aportar para poder mostrar en el perfil de cada usuario las noticias y comentarios realizados por él en el tiempo que lleva registrado.

Por otro lado, también está la entidad “NoticiaSeleccionada” que representa una noticia agregada por el usuario para que aparezca en sus favoritos.

Cada usuario tendrá su propio blog, el cual tendrá entradas que se podrán visualizar, modificar, eliminar y comentar. La entidad “Entrada” será parecida a las entidades “Noticia” y “Comentario”, pero con sus respectivos atributos y métodos. Por esta parte, también se tiene una clase que representa una lista de entradas de un usuario “ListaEntradas” con el mismo objetivo que las entidades “ListaNoticias” y “ListaComentarios”.

Para cuando ocurra el caso de que un usuario haya olvidado algún dato imprescindible para entrar al sistema, tendremos la entidad “RecuperacionDatos”, que se encargará de enviar por correo la información necesaria para entrar al sistema siempre y cuando a través del formulario de recuperación de contraseña se haya introducido un email validado y que se encuentre en la base de datos.

Aunque no se muestre en la figura 15, se ha creado una clase padre que representa las identidades de todos los objetos a registrar. Para tener un control o un valor distinto de cada registro de la base de datos se ha creado una clase “Id” de la cual heredarán “IdUsuario”, “IdNoticia”, “IdNoticiaSeleccionada”, “IdComentario”, “IdEntrada”, “IdComentarioBlog” y “IdCategoria”. Cada vez que se vaya a registrar un usuario o publicar algún documento (noticia, comentario, entrada,...) se procede a generar un id de forma óptima para el objeto que sea. Para representar esto, se ha utilizado un patrón de diseño, Strategy. Este patrón permite mantener un conjunto de algoritmos de los que el objeto cliente puede elegir aquel que le conviene e intercambiarlo según sus necesidades. En este caso, dependiendo del registro que se vaya a insertar se elige un algoritmo u otro del método Genera(), que se encarga de buscar huecos o de recoger la id del último registro en las distintas tablas para asignar una nueva identidad.

Cada usuario va a tener asociada una lista de categorías representado mediante la entidad “ListaCategorias”. Esta lista contendrá la información necesaria de cada categoría que le interese al usuario. Será útil para mostrar en página principal aquellas noticias de interés para un usuario.

El usuario tendrá varias opciones para subir imágenes al servidor. Podrá colgar una foto a su perfil, podrá publicar noticias con imagen y podrá publicar entradas con imagen. Para representar esta estructura también se ha utilizado el patrón de diseño Strategy de la misma forma que con las identidades. En este caso, tendremos una clase padre “Upload” que contendrá los métodos a implementar de distinta manera en sus clases derivadas (Upfoto, UpImagenNotica, UpImagenEntrada y UpImagenNoticiasSeleccionadas).

6.7. Diagrama de secuencia

6.7.1 Introducción

Los diagramas de secuencia son un tipo de diagrama de interacción que modelan los aspectos dinámicos del sistema donde se destaca la relación temporal de los mensajes.

El diagrama de secuencia es uno de los diagramas más efectivos para modelar la interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

A continuación se va a utilizar los diagramas de secuencia para representar los posibles escenarios de cada caso de uso, es decir, casos de uso en ejecución.

6.7.2. Registro de usuarios

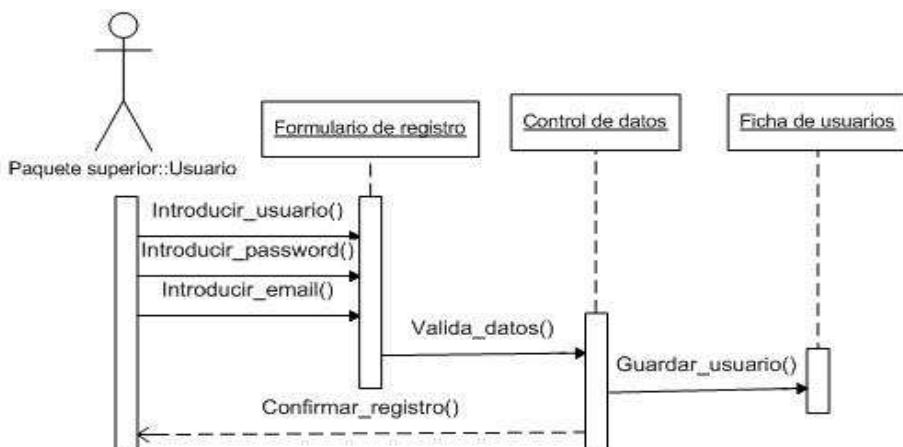


Figura 16. Diagrama de secuencia del caso de uso Registro.

Tal como muestra la figura 16, un usuario introducirá el nombre de usuario, password y email en un formulario de registro. Estos datos pasarán por un control de datos y si son correctos se guardarán en la base de datos y se confirmará el registro del nuevo usuario.

6.7.3. Login

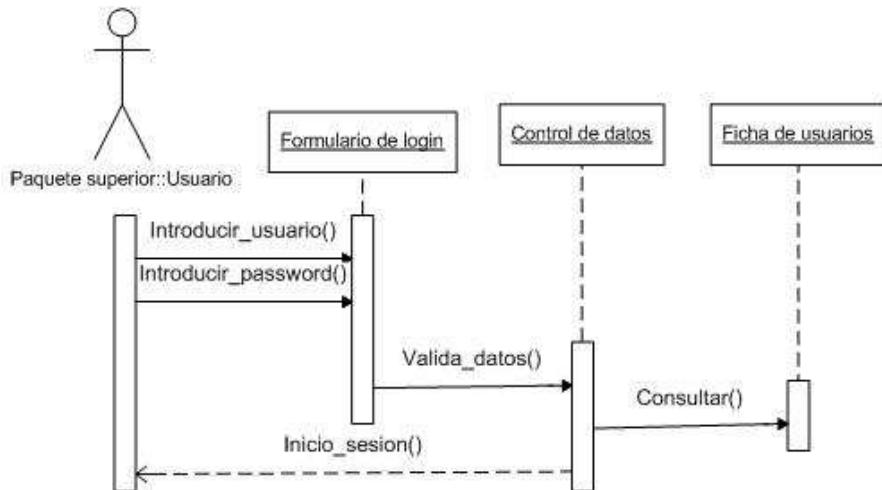


Figura 17. Diagrama de secuencia del caso de uso Login.

Tal como muestra la figura 17, un usuario, una vez se haya registrado, introduce un nombre de usuario y su contraseña en un formulario de login. Estos datos se validarán pasando un control y consultando la existencia de los datos introducidos en la base de datos. Si el usuario existe, se inicia sesión.

6.7.4. Gestión del perfil

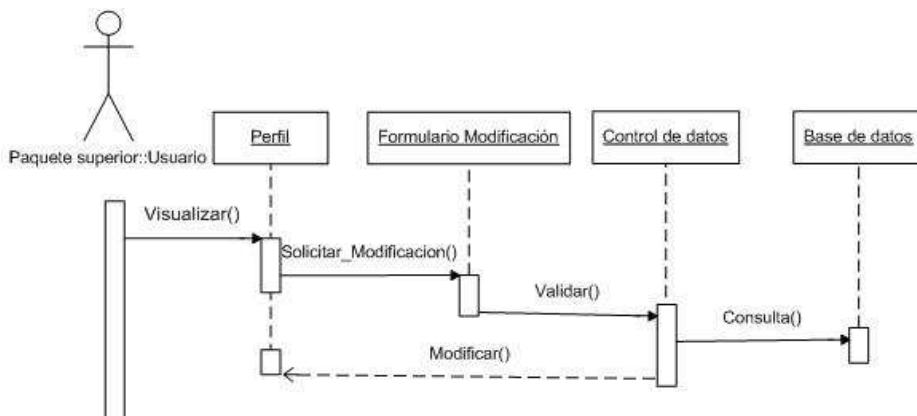


Figura 18. Diagrama de secuencia del caso de uso Gestión de perfil.

Tal como se muestra en la figura 18, un usuario, una vez haya iniciado sesión, podrá acceder a su perfil y lo primero que hará será visualizarlo. Tendrá la opción de modificar su perfil en cualquier momento mediante un formulario de modificación. Una vez haya introducido los nuevos datos en el formulario de modificación se procederá a la validación (sobre todo del email) consultando la base de datos para no permitir la duplicación.

6.7.5. Gestión de noticias

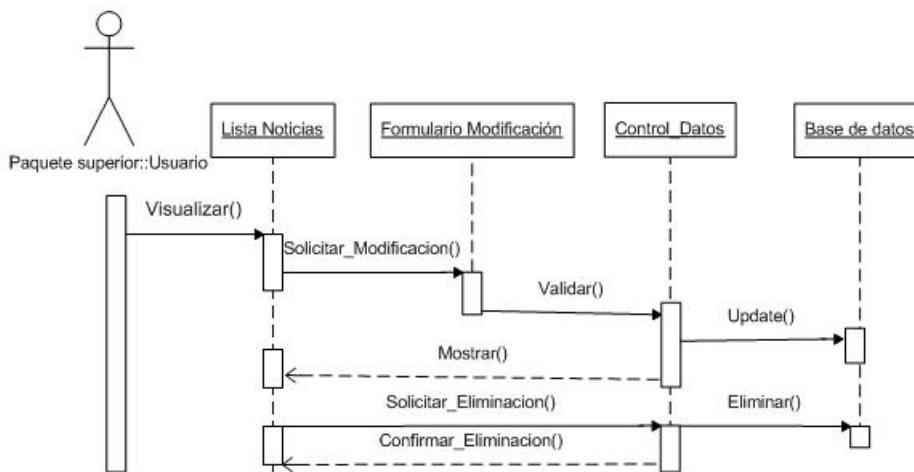


Figura 19. Diagrama de secuencia de la Gestión de noticias.

Tal como se muestra en la figura 18, un usuario podrá visualizar sus noticias publicadas, modificarlas mediante un formulario de modificación, donde se controlarán esos nuevos datos introducidos y se actualizará el registro en la base de datos y seguidamente se mostrará por pantalla. Cada noticia también podrá ser eliminada, tanto de la lista de noticias como de la base de datos.

6.7.6. Gestión de favoritos

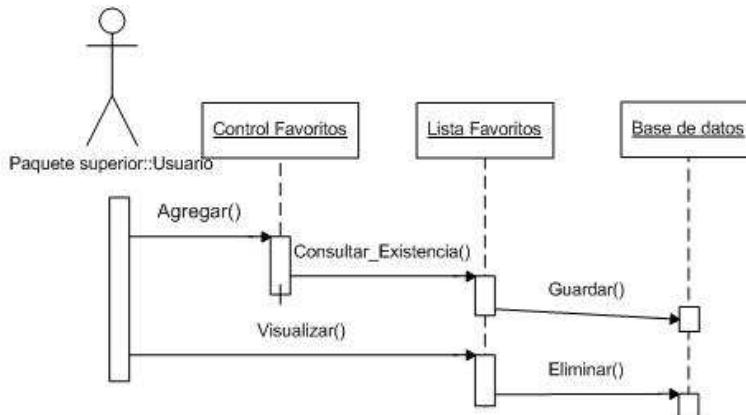


Figura 20. Diagrama de secuencia de la Gestión de favoritos.

Tal como se muestra en la figura 20, un usuario a través de la página principal podrá agregar a sus favoritos aquellas noticias que le interesen. Lógicamente no podrá seleccionar una noticia que ya esté en sus favoritos. A través de su perfil, podrá gestionar sus favoritos, donde se visualizará la lista de noticias

seleccionadas que en cualquier momento podrán ser consultados, ordenadas o eliminadas.

6.7.7. Gestión de comentarios

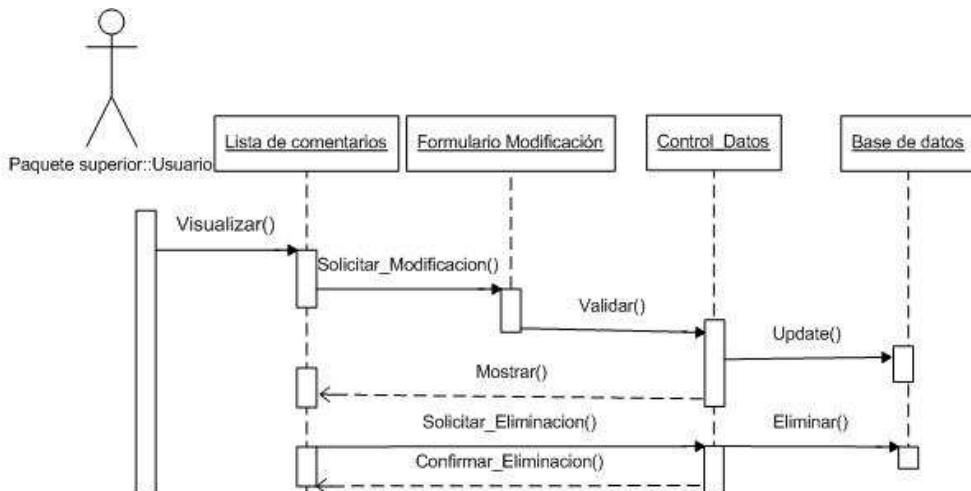


Figura 21. Diagrama de secuencia de la Gestión de comentarios.

Tal como se muestra en la figura 19, un usuario podrá visualizar sus comentarios realizados, modificarlos mediante un formulario de modificación, donde se controlarán esos nuevos datos introducidos y se actualizará el registro en la base de datos y seguidamente se mostrará por pantalla. Cada comentario también podrá ser eliminado, tanto de la lista de comentarios como de la base de datos.

6.7.8. Personalización del sitio Web

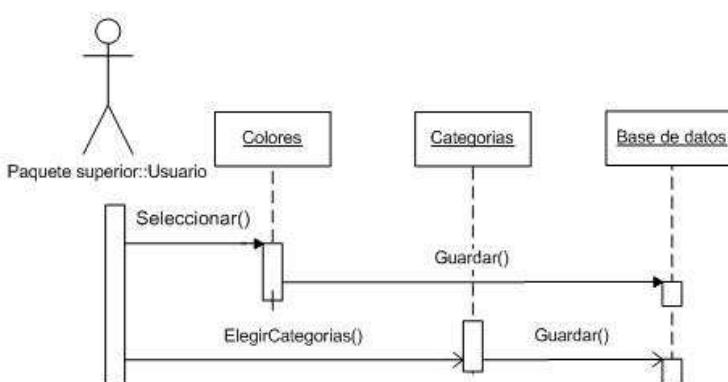


Figura 22. Diagrama de secuencia de la Personalización del sitio Web.

En este caso, el usuario podrá seleccionar un color de fondo para el sitio Web o elegir aquellas categorías sobre noticias a mostrar en página principal.

6.7.9. Gestión de blog

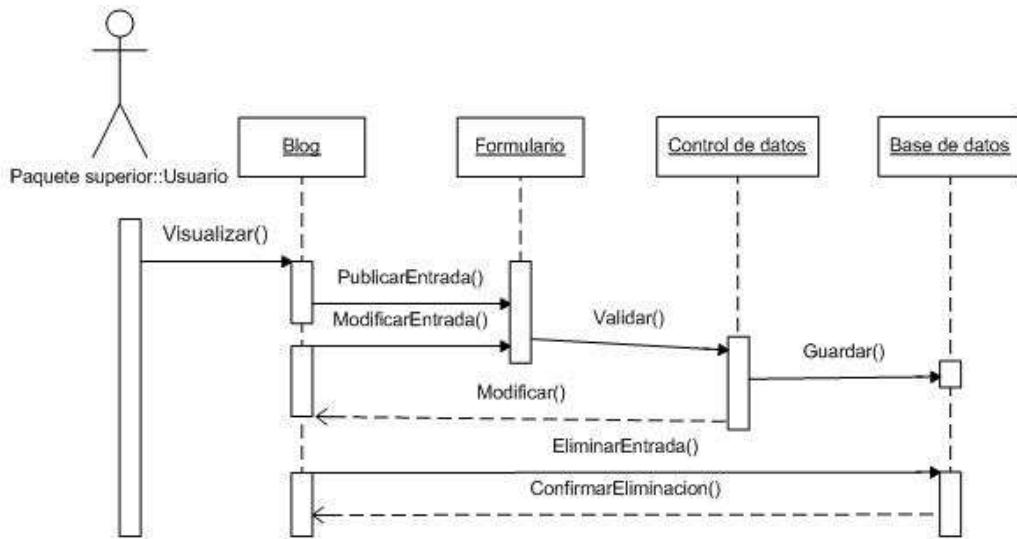


Figura 23. Diagrama de secuencia de la Gestión de blog.

Tal como se muestra en la figura 20, un usuario podrá visualizar su blog mostrando por pantalla la última entrada realizada. Tendrá las opciones de publicar una nueva entrada, modificar la entrada que esté en pantalla, eliminarla y consultar otras entradas. Tanto cuando publica una entrada como cuando la modifica, el sistema validará esa entrada y si es correcta se procederá a la inserción en la base de datos y visualización de la modificación o nueva entrada. Cuando un usuario decida eliminar una entrada, se borrará directamente de la base de datos y se confirmará por pantalla la eliminación.

6.8. La base de datos del sistema

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen unos programas denominados Sistemas Gestores de Bases de Datos, (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas, como es en el caso de este proyecto. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD).

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación. Según la variabilidad de los datos almacenados la base de datos que se va a definir es dinámica, ya que la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

El contenido de la base de datos serán datos relevantes al usuario; su información de usuario, sus datos personales si lo desea, sus noticias publicadas y comentarios realizados, sus categorías favoritas y sus entradas al blog.

Al tratarse de una base de datos relacional, las relaciones o tablas deben cumplir una serie de propiedades:

- No hay registros duplicados.
- Los registros y los atributos no están ordenados.
- Todos los valores de dominio son atómicos, es decir, si se divide el valor de los atributos se pierde su significado original.

En la siguiente figura se muestra la base de datos del sistema:

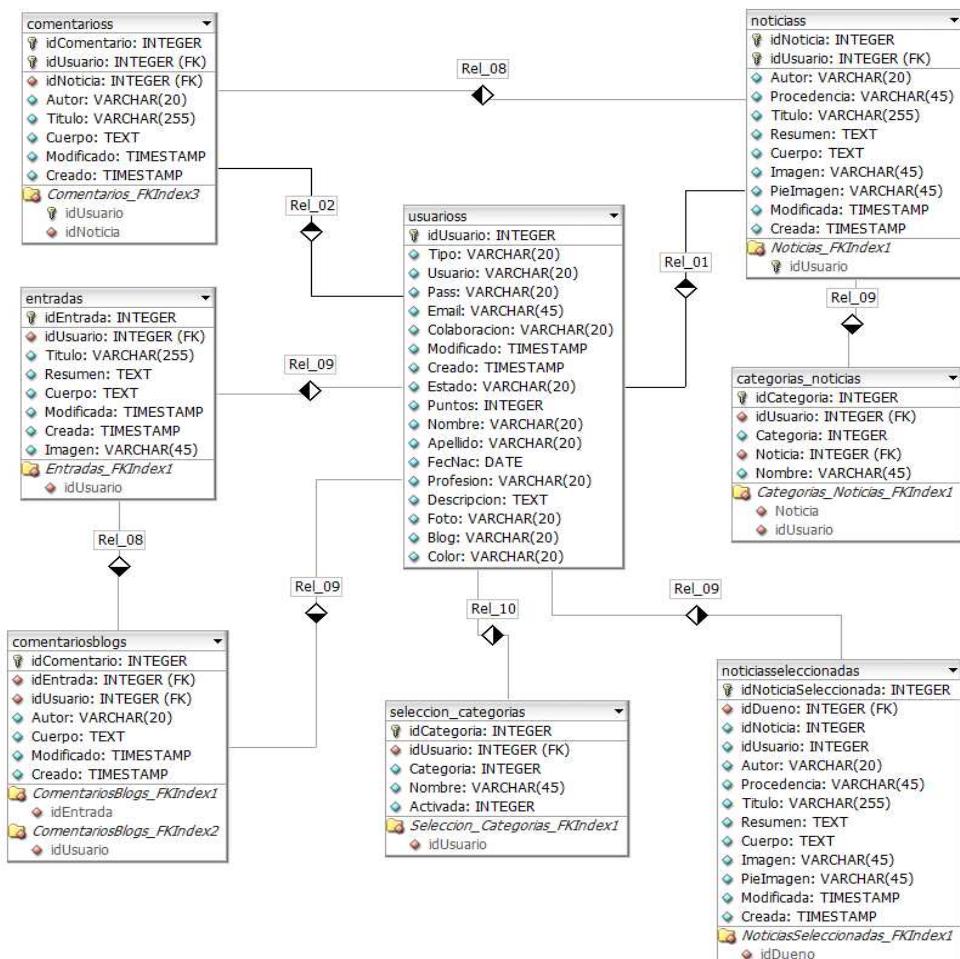


Figura 24. La base de datos del sistema.

La tabla más importante de esta base de datos está claro que es la tabla “usuarios” ya que casi todas las demás tablas están relacionadas con ella. Será una tabla en la que se guardarán todos los usuarios registrados (tanto administradores, colaboradores como amigos de Kaos).

El atributo *idUsuario* corresponde a la clave primaria y externa de la tabla “usuarios”. Se trata del identificador único de cada usuario y con él podremos acceder a la tabla “noticias”, “comentarios”, “categorias” y “entradas”. Se genera un nuevo identificador cada vez que un usuario se registra para evitar la duplicación y es tipo numérico auto-incremental.

El atributo *Tipo* será una cadena que puede contener 3 valores distintos, que un usuario sea administrador querrá decir que el atributo *Tipo* = “admin” (el usuario podrá gestionar o controlar a todos los usuarios), que un usuario sea colaborador querrá decir que el atributo *Tipo* = “colaborador” y que un usuario sea un amigo de “Kaos”, querrá decir que el atributo *Tipo* = “user” (el usuario solo podrá gestionar sus datos).

En el atributo *Usuario* se guarda el *nickname* o nombre de usuario que utiliza un usuario para entrar al sistema. Se guardará una cadena de 20 caracteres como máximo.

En el atributo *Pass* se guarda la contraseña que emplea un usuario para iniciar sesión. Se guardará una cadena de 20 caracteres como máximo.

En el atributo *Email* se guarda el correo electrónico asociado a un usuario. Se guardará una cadena de 45 caracteres como máximo.

En los atributos *Creado* y *Modificado* se guardarán respectivamente la fecha de registro del usuario y la fecha de última modificación de su perfil.

El atributo *Nombre* y *Apellido* contendrán una cadena con el nombre y apellido que tienen los usuarios en la vida real. Pueden estar vacíos ya que son opcionales.

En *FecNac* se guardará la fecha de nacimiento de cada usuario. Será tipo DATE porque solo desearemos guardar el día, mes y año de nacimiento. Podrá contener un valor vacío ya que también es opcional.

El atributo *Profesion* también contendrá una cadena con información sobre a que se dedica cada usuario en la vida real. Podrá contener un valor vacío ya que también es opcional.

El atributo *Descripcion* será tipo TEXT para que cada usuario pueda redactar una parrafada sobre como es. Podrá contener un valor vacío ya que también es opcional.

El atributo *Foto* contendrá una cadena con el nombre del archivo que se desea colgar para mostrar en el perfil. Podrá contener un valor vacío ya que también es opcional.

La tabla “noticias” va a tener 3 claves distintas ya que será una tabla relacionada con la tabla “Usuarios” (cada usuario podrá tener de 0 a muchas noticias), relacionada con la tabla “Comentarios” (una noticia podrá tener 0 o muchos comentarios), y relacionada con la tabla “categorías_noticia” (cada noticia debe pertenecer al menos a una categoría, en este caso, de cada tipo: territorios, secciones o especiales).

IdNoticia es la clave primaria que identifica cada noticia. Se genera una nueva cada vez que un usuario publica una noticia y es tipo numérico auto-incremental.

IdUsuario indicará a que usuario pertenece una noticia. Es una clave foránea que apunta a la tabla “Usuarios”.

El atributo *Autor* guardará el nombre de usuario autor de la noticia en cuestión. Será una cadena de 20 caracteres como máximo. Es de carácter obligatorio, no puede contener un valor vacío.

El atributo *Titulo* también guardará una cadena pero de longitud máxima 255 caracteres. Como bien indica el nombre, guardará el título de una noticia. Es de carácter obligatorio, no puede contener un valor vacío.

En el atributo *Cuerpo* se guardará el texto de la noticia. Será tipo TEXT para poder guardar un conjunto de parrafadas. Es de carácter obligatorio, no puede contener un valor vacío.

El atributo *Imagen* contendrá una cadena con el nombre del archivo que se desea colgar para acompañar a la noticia. Es opcional, podrá tener un valor vacío.

En los atributos *Creada* y *Modificada* se guardarán respectivamente la fecha de publicación de la noticia y la fecha de última modificación.

La tabla “comentarios” tiene relación con 2 tablas: la tabla “usuarios” y la tabla “noticias”. Es una tabla muy parecida a la tabla “noticias” pero con algunas pequeñas diferencias, como por ejemplo, que un comentario no contiene una imagen como si una noticia.

La tabla “entradas” también es semejante a “comentarios” y “noticias”, prácticamente con los mismos atributos pero con información a guardar de distinto índole.

La tabla “comentariosblog” es idéntica a la tabla de “comentarios”, pero en este caso los comentarios van referidos a las entradas de los blogs.

La tabla “categorias_noticias” va a ser una tabla para guardar la lista de categorías a la que pertenece una noticia en concreto.

Y, por último, la tabla selección_categorias es una tabla que se utiliza para guardar aquellas categorías de interés para cada usuario en concreto con el objetivo principal de mostrar aquellas noticias con categoría de interés para el usuario.

Un atributo interesante a comentar es el atributo *Activada*. Se trata de un boolean que si está a *true* indica que la categoría está seleccionada por el usuario para posteriormente mostrar aquellas noticias en página principal que pertenezcan a esta categoría. Si está a *false* indicará todo lo contrario.

7. Implementación

7.1. Introducción

Al finalizar la fase de diseño de cada requerimiento ya se puede decir que la mayoría de decisiones ya han sido tomadas y que se puede pasar a la fase de codificación. Es posible que algunas de las decisiones tomadas en la anterior fase se vean acotadas por el lenguaje de programación que se vaya a utilizar. En esta fase lo que se pretende es transportar el diseño definido a un código escrito en el lenguaje de programación escogido.

El código que se genera a partir del diseño debe resultar lo más claro posible, es por ello que donde se ha creído pertinente se ha añadido comentarios para explicar que hace ese código en concreto.

7.2. Estructura de archivos

La estructura del proyecto se desglosa en carpetas y archivos. Cada clase implementada está en un único archivo en la carpeta de clases. Las interfaces de usuario, la plantilla del blog y el blog de cada usuario están situados en el directorio raíz.

Las hojas de estilo y algunas funciones JavaScript que se utilizan en algún lugar del sitio Web también van a tener su carpeta específica.

Las hojas de estilo que se van a utilizar principalmente son dos hojas distintas, una para toda la Web y otra solo para la plantilla del blog. La hoja de estilo para la Web en general tendrá copias con algunas modificaciones para que el usuario pueda escoger una de esas hojas según un color escogido. El estilo del blog es como es y no se podrá personalizar.

Además, se va a tener una carpeta dinámica que será donde se guardará todas las imágenes de cada usuario. Es decir, en el servidor cada usuario tendrá su carpeta específica donde se guardará cada imagen que cuelgue, ya sea para su perfil, para alguna noticia o para alguna entrada de su blog.

7.3. Ejemplo de codificación

En este apartado vamos a detallar como se ha implementado el módulo del perfil de cada usuario, teniendo en cuenta, en este caso, que hay tres tipos de usuarios distintos para esta comunidad.

Se ha elegido este apartado porque se considera uno de los apartados que más se quería conseguir y se trata del centro de la aplicación. A partir de un inicio de sesión, todo lo que vea el usuario es información propia y privada que solo él, en principio, podrá modificar.

En la siguiente imagen se muestra el perfil de un usuario recién registrado. A medida que vaya publicando noticias, realizando comentarios, agregando noticias a sus favoritos o modificando su perfil, la información de su perfil irá cambiando. Lo único que se ha hecho es modificar el perfil para añadir una foto y así ver el perfil más atractivo, pero al perfil se pueden añadir más datos o menos según quiera el usuario.



Figura 25. Interfaz Perfil de usuario.

Como ven, desde el perfil de un usuario se puede volver a la página principal pulsando en el logo o en el menú donde dice “Página principal”. Se puede visualizar la información de un usuario; identificador, usuario, password, email, tipo, estado, fecha de registro, fecha de última modificación, noticias publicadas, noticias seleccionadas, comentarios realizados y los puntos que lleva hasta el momento, entre otros datos opcionales.

El sistema de puntos se explica pulsando en el hipervínculo “Puntos” y pulsando en el número de puntos se accede al ranking de usuarios.

En cualquier momento se puede modificar el perfil pulsando en tal hipervínculo donde se accede a un formulario de modificación de datos. Puedes visitar tu blog o quitar la foto asociada al perfil si es que está puesta.

Desde el perfil, cada usuario también puede acceder a sus noticias publicadas, noticias seleccionadas o favoritos, comentarios realizados y tener opción a personalizar el sitio Web.

Nos vamos a centrar solamente en la implementación de la página perfil. Como en todas las páginas privadas o que necesiten inicio de sesión para poder ser vistas, lo primero que incluimos son las siguientes dos líneas de código:

```
<?php
session_start(); //Inicio la sesión
header("Cache-control: private"); //Arregla IE 6
?>
```

La función `session_start()` de PHP se utiliza para inicializar una variable de sesión o para mantener una existente. Es lo primero que debe hacerse en cada página privada.

Con el código que se muestra a continuación, incluimos en la página la clase “usuario”, asignamos la variable de sesión a una variable y creamos una instancia de usuario pasándole como parámetro dicha variable de sesión, que en este caso se trata de la id del usuario en sesión.

```
<?php
include_once("clases/usuario.php");
$idsesion = $_SESSION['id'];
$usuarioSesion = new Usuario($idsesion);
?>
```

Con lo siguiente, lo que se hace es asignar el estilo que toque dependiendo del usuario que esté accediendo. Es un trozo de código que se utiliza en todas las páginas en las que para visualizarlas se necesite iniciar sesión.

```
<?php
//Si todavía no ha cambiado el color de la Web, utilizar el color por defecto
if ($usuarioSesion->getColor() == NULL || $usuarioSesion->getColor() == "estilo")
    echo '<link rel="stylesheet" href="css/estilo.css" type="text/css"/></link>';
else
    echo '<link rel="stylesheet" href="css/'.$usuarioSesion->getColor().'css" type="text/css"/></link>'; 67
?>
```

El método getColor() de la clase “usuario” retorna el color de fondo de página que eligió el usuario al personalizar el sitio Web. Es un método que accede a la base de datos y retorna el atributo *color* de un usuario en concreto, en este caso, el que esté en sesión.

El menú horizontal que se muestra en la figura 25 está hecho en HTML y CSS. Dependiendo del tipo de usuario que acceda, se mostrarán unos hipervínculos u otros. Por ejemplo, como solo administradores y colaboradores pueden publicar noticias, el resto de usuarios no tendrán el apartado de noticias publicadas.

A continuación viene el cuerpo de la página, donde se incluyen otras clases necesarias y se procesan los datos que vienen del formulario de modificación o se procesa la eliminación de la foto de perfil.

Para la eliminación de la foto del perfil, simplemente hacemos lo siguiente:

```
if ($_GET['id'] == 'quitar') {
    $usuarioSesion->EliminarFoto();
    $usuarioSesion->setFoto("NULL");
}
```

Si se pulsa el hipervínculo “Quitar foto”, se procesa otra vez la página con id=’quitar’. Se cumple la condición de este if, se elimina la foto físicamente del servidor con el siguiente método de la clase “usuario”:

```
public function EliminarFoto() {
    $directorio = 'usuarios/'.$this->getId().'/foto';
    //Si se ha colgado una foto anteriormente y existe el directorio en el servidor
    if ($this->getFoto() != "NULL" && is_dir($directorio)) {
        $fichero = $directorio.'/'.$this->getFoto();
        unlink($fichero);
    }
}
```

Donde lo que se hace es acceder al directorio donde está situada la foto y eliminarla si existe con la función unlink(archivo) de PHP. El método getId() que se utiliza para acceder al directorio donde está la foto, es un método que retorna la id de un usuario. Como el método EliminarFoto() pertenece a la clase “usuario”, utilizamos \$this para acceder getId() ya que están en la misma clase.

Y también se pone un valor NULL a la imagen asociada a un usuario en la base de datos con el siguiente método de la clase “usuario”:

```
//Método que modifica la foto de usuario de un usuario en concreto
public function setFoto($nuevaFoto) {
    $update = "Update usuarios Set Foto = '$nuevaFoto' WHERE idUsuario = '$this->id'";
    mysql_query($update, $this->con);
}
```

La parte de código que viene a continuación solo se ejecutará cuando un usuario decida acceder al formulario de modificación y actualizar algún dato de usuario, es decir, email o contraseña, ya que nombre de usuario no es posible cambiarlo.

```
$correcto = true;
if (isset($_POST['upusuario'])) {
    $correcto = false;
    $registro = new RegistroUsuario($_POST);
    if (!$registro->ExisteEmail() || $_POST['email'] == $usuarioSession->getMail()) {
        $correcto = true;
    }
    else {
        echo '<div align="center">';
        echo '<p><div class="ui-state-error" style="width:730px" align="left"><br/>';
        echo '';
        echo '&nbsp;&nbsp;Error. No puede utilizar el email '.$_POST['email'].', ya existe.<br/><br/></div></p>';
        echo '</div><br/>';
        $correcto = false;
    }
    if ($correcto) {
        $usuarioSession->Modificar($_POST);
        echo '<div align="center">';
        echo '<p><div class="ui-state-highlight" style="width:730px" align="left"><br/>';
        echo '';
        echo '&nbsp;&nbsp;Datos de usuario actualizados correctamente.<br/><br/></div></p>';
        echo '</div><br/>';
    }
}
```

En este trozo de código lo primero que se hace es mirar si los nuevos datos (en este caso solo email) no están duplicados en la base de datos o es el mismo que tenía antes. Si los nuevos datos no son correctos, se entra en el else y se imprime tal error. Para procesar la modificación de los datos personales el código es parecido.

Si los datos son correctos se procede a la modificación mediante el método Modificar(datos) de la clase “usuario” y se informa que la modificación se ha realizado con éxito.

El método Modificar(datos) es el siguiente:

```

//Método que modifica los atributos de un usuario en concreto
public function Modificar($datos) {
    if (isset($datos['upusuario'])) {
        $this->setPassword($datos['pass']);
        $this->setMail($datos['email']);
    }
    else {
        $this->setNombre($datos['nombre']);
        $this->setApellido($datos['apellido']);
        $fecha = $datos['ano'].'-'.$datos['mes'].'-'.$datos['dia'];
        $this->setFecNac($fecha);
        $this->setProfesion($datos['profesion']);
        $this->setDescripcion($datos['descripcion']);
    }
    $this->setFechaModificacion(date("Y-m-d H:i:s"));
}

```

Como ven, dependiendo de los datos que se vayan a modificar (datos de usuario o datos personales) se ejecutará una parte u otra de la condicional. Los métodos `setX(nuevaX)` son idénticos al método explicado anteriormente `setFoto(nuevafoto)`, pero modificando el atributo que corresponda.

Al final de esta página lo que se hace es mostrar el usuario actualizado en cuestión mediante el método `Mostrar(flag)` de la clase “usuario”. Y lo que le sigue ya son los hipervínculos para modificar el perfil, visitar su blog o quitar la foto de perfil si tiene.

```

$usuarioSesion->Mostrar(0);
echo '<div align="right"><ul><li><a href="modificaperfil.php" title="Modifica tu perfil de usuario">
    <span>Modificar Perfil</span></a></li>
<li><a href="'.$usuarioSesion->getUsuario().'.'.php"><span>Visita tu blog</span></a></li>';
if ($usuarioSesion->getFoto() != "NULL")
    echo '<li><a href="tuperfil.php?id=quitar" title="Eliminar foto">
        <span>Quitar foto</span></a></li></ul>';
echo '</div>';

```

El método `Mostrar(flag)` recibe un parámetro para indicar quien es el que está accediendo a visualizar el perfil, ya que dicha página puede ser visualizada por el propio dueño u otros usuarios registrados y logeados que deseen visitar el perfil. El parámetro es importante ya que si se trata de otro usuario el que desee ver el perfil de otro, habrá datos como por ejemplo la contraseña que no debe ver.

El resto de páginas del proyecto son más o menos por el estilo, siempre utilizando programación orientada a objetos con PHP, HTML y CSS. Para más detalles consulte el manual de usuario.

8. Pruebas

8.1. Introducción

Con la fase de pruebas se intenta realizar una evolución por módulos del sistema para verificar que se satisfacen los requerimientos especificados o para identificar las diferencias entre los resultados esperados y los obtenidos.

Mediante las pruebas podemos evaluar la calidad del software. Su función principal se basa en la detección de errores que pueden haberse dejado escapar en cualquiera de las etapas anteriores.

A continuación se comenta los diferentes tipos de pruebas que se han realizado para esta aplicación Web.

8.2. Pruebas de unicidad

Se trata de verificar uno a uno los módulos (clases), funciones o partes del programa de manera individual.

A medida que se iba codificando las diferentes funcionalidades se procedía a depurar el código con el fin de evitar errores. Cada componente que se iba añadiendo se le realizaba pruebas para asegurar que su funcionamiento sea el correcto.

8.3. Pruebas de integración

Se trata de verificar los diferentes módulos, pero esta vez como un conjunto y no por separado. Sirven para detectar interfaces inconsistentes o problemas de compartición de variables. Estas pruebas se realizan una vez se ha codificado todos los módulos o una parte importante de ellos.

A medida que se iba avanzando el trabajo de los diferentes módulos se ha ido comprobando que la interacción entre ellos se realiza de forma correcta.

Por ejemplo, para realizar las pruebas de interfaz se han ido creando varios usuarios, con los que se ha ido interactuando con el sitio para verificar que se cumplen los requerimientos impuestos por el cliente.

8.4. Pruebas de validación

Las pruebas de validación de este proyecto se han hecho de acuerdo con la técnica de diseño de los casos de prueba a partir de los casos de uso.

Se han desarrollado dos tipos de técnicas de pruebas de validación:

- **Black Box Testing:** El testador ve el programa (o la unidad a probar) como una caja negra el contenido de la cual desconoce. Los casos de prueba (test cases) se obtienen únicamente atendiendo a la especificación (los requerimientos si es toda una aplicación), y no se tiene que tener conocimientos sobre su implementación.
- **White Box Testing:** Los casos de prueba se obtienen (se diseñan) a partir de la estructura interna de la unidad a probar y de la especificación. Esta es necesaria para poder comparar el resultado obtenido con el esperado. Esta estrategia asume que los errores se esconden en expresiones y sentencias de código individuales y en las condiciones. Por esto, los casos de prueba se diseñan para cubrir todas las sentencias, ramas de ejecución y condiciones lógicas.

8.5. Pruebas de sistema

Se comprueba que se haya cumplido los requerimientos establecidos en la fase de análisis y que, por tanto, el programa se comporta de la forma deseada y da respuesta de manera adecuada a los requerimientos formulados previamente.

Solo al terminar el proyecto se han planificado pruebas determinadas para comprobar la usabilidad, la seguridad, el rendimiento y otros factores presentes en una Web de estas características.

8.6. Pruebas de compatibilidad en varios navegadores

Este tipo de pruebas han consistido principalmente en probar la aplicación Web en distintos navegadores. En este caso, la aplicación ha sido probada en los navegadores: Internet Explorer 7, Mozilla Firefox 3.0 y Opera 9.6.

El único problema que se ha detectado es que no se muestra de la misma forma alguna parte del contenido en Internet Explorer que en Mozilla Firefox, pero no se ha considerado un problema grande porque se trata de conceptos de estilos, como por ejemplo que una imagen se muestre a la derecha en Mozilla Firefox y en Internet Explorer se muestre centrada.

8.7. Pruebas de aceptación

Tiene el fin de obtener la aceptación de la aplicación por parte del cliente. Cada una de estas pruebas debe ser superada con el fin de que el cliente adquiera la aplicación cumpliendo las perspectivas del análisis, es decir, que realice todas las funciones diseñadas y que no cometa errores en el cálculo de las mismas.

En este caso, el proyecto ha sido probado por el cliente y aceptado.

9. Conclusiones

Como conclusión de este proyecto se puede decir que el trabajo desarrollado ha aportado para el desarrollador una gran experiencia en cuanto a lenguajes de programación orientados a servidor, en la realización de hojas de estilo interesantes para el diseño de una Web y en la utilización de la programación orientada a objetos. Además, de haber tenido la oportunidad de poner en práctica los conocimientos que se han ido adquiriendo durante los estudios de la carrera universitaria.

El desarrollador ha podido realizar toda la fase de creación de software desde su planificación, análisis, diseño, implementación y su puesta en marcha. Hecho el cual ha supuesto una muy buena experiencia ya que estas son las tareas a desarrollar en cualquier proyecto de software en el mundo profesional.

Haciendo un repaso resumido a los objetivos que se marcaron al inicio del proyecto, puede observarse que se han podido cumplir todos en más o menos medida.

Aspectos originales y más creativos a destacar sobre este trabajo es sin duda alguna la experiencia que se ha desarrollado con la implementación del blog para cada usuario registrado o la dinamización de las noticias en la página principal.

Cabe decir que se han producido desviaciones en la planificación, parece que resulta inevitable que surjan nuevos factores o problemas no previstos que alteran la duración de las tareas y que de esta manera afecte a la duración total del proyecto. En este caso, la desviación sufrida no ha sido muy importante y al menos se ha podido realizar en el tiempo que dura un curso académico.

Aunque el programa cumple la mayoría de objetivos previstos, se pueden efectuar algunas ampliaciones, como por ejemplo: que los usuarios se puedan enviar mensajes entre ellos, con algún sistema de correo o mensajería instantánea, que los usuarios puedan elegir más opciones a la hora de cambiar el estilo de la página Web, que el blog también sea personalizable, etc.,

10. Resumen

En la actualidad, el término Web 2.0 está siendo muy adherido a todo tipo de páginas Web o han aparecido una multitud de páginas Web que fomentan su uso.

Este término engloba todos aquellos sitios Web donde el usuario o cliente pasa de un papel más pasivo (lector) a un papel más activo (generador de contenido). Este tipo de sitio Web sirve como vehículo para facilitar que sean los propios usuarios o clientes los que creen el contenido o enriquezcan el mismo. Los ejemplos quizás más famosos son www.wikipedia.org o www.youtube.com.

En general las técnicas, conceptos o tecnologías Web 2.0 aplicadas a los sitios Web corporativos hacen que estos sean altamente interactivos y permiten experiencias de marca más bidireccionales y participativas.

Sin duda alguna este proyecto conlleva toda la tecnología Web 2.0 o una gran parte lo que ha aportado al desarrollador una gran experiencia en esta terminología que podría dar salida profesional.

Bibliografía y fuentes electrónicas

F. Javier Gil Rubio, Santiago Alonso Villaverde, Jorge A. Tejedor Cerbel y Agustín Yagüe Panadero, "Creación de sitios web con PHP5", McGraw-Hill, 2006.

David Powers, "Desarrollo web dinámico con Dreamweaver 8 y PHP", ANAYA, 2006.

Wikipedia (2008, Noviembre). PHP [Online]. Disponible en:
<http://es.wikipedia.org/wiki/.php> (Enero 2009).

Wikipedia (2008, Octubre). MySQL [Online]. Disponible en:
<http://es.wikipedia.org/wiki/MySQL> (Enero 2009).

Wikipedia (2008, Diciembre). JavaScript [Online]. Disponible en:
<http://es.wikipedia.org/wiki/JavaScript> (Enero 2009).

Desarrolloweb.com. (). PHP a fondo. Manuales gratis PHP [Online]. Disponible en: <http://www.desarrolloweb.com/php/> (2008, Noviembre).

Juaquín García. (2006, Agosto). Manual de PHP [Online]. Disponible en: <http://www.webestilo.com/php/> (2008, Noviembre).

Wikipedia (2008, Septiembre). Web 2.0 [Online]. Disponible en:
http://es.wikipedia.org/wiki/Web_2.0 . (Noviembre 2008).

Wikipedia (2008, Septiembre). Blog [Online]. Disponible en:
<http://es.wikipedia.org/wiki/Blog>. (Diciembre 2008).

Maestros del Web. (2005, Octubre). ¿Qué es la Web 2.0? [Online]. Disponible en: <http://www.maestrosdelweb.com/editorial/web2/>. (Diciembre 2008).

liveValidation (2007-2008). LiveValidation. Validation as you type [Online]. Disponible en: <http://www.livevalidation.com/> (Febrero 2009).

Wordpress. (2008). Blog tools and publish platform [Online]. Disponible en: <http://wordpress.org/> . (Marzo 2009).

eZ. (2007, Agosto). PHP. PHP / Standards / Development [Online]. Disponible en: <http://ez.no/ezpublish/documentation/development/standards/php> . (Abril 2009).

CartoWeb. Code Convention. CartoWeb Documentation [Online]. Disponible en: http://www.cartoweb.org/doc/cw3.2/xhtml/dev.code_convention.html . (Abril 2009).

Moxiecode Systems AB. (2009). TinyMCE - Javascript WYSIWYG Editor. <http://tinymce.moxiecode.com/> (Abril 2009)

Mike Hall. (1999-2002). Arrastre Genérico. Kursor xdhtml – BrainJar.com Generic Drag. Disponible en: <http://www.kusor.net/traducciones/brainjar.es/drag1.es.html> (Mayo 2009)

Jorge Pinedo. (2005). Paginator version 1.6. Paginación con PHP y mySQL [Online]. Disponible en: <http://jpinedo.webcindario.com/scripts/paginator/> . (Mayo 2009).

stu nicholls. (2005). List of Menus. CSS only menus [Online]. Disponible en: <http://www.cssplay.co.uk/menus/index.html>. (Mayo 2009).

Paul Bakaus and the jQuery UI Team. (2009). JQuery ui. JQuery user interface [Online]. Disponible en: <http://jqueryui.com/> (Mayo 2009).

Wikipedia. (2009, Junio). FileZilla [Online]. Disponible en: <http://es.wikipedia.org/wiki/FileZilla> . (Junio 2009).