



**Universitat Autònoma
de Barcelona**

PFC

Gestor de contenidos web ASP.NET

**Memoria del proyecto
de Ingeniería Técnica en
Informática de Sistemas**

realizado por

Ricard Blasco Turón

Y dirigido por

Marc Talló Sendra

Escola d'Enginyeria

Sabadell, Juny de 2010

El abajo firmante, *Marc Talló Sendra*,
profesor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el trabajo al que corresponde esta
memoria ha sido realizado bajo su dirección
por *Ricard Blasco Turón*

Y para que conste firma la presente.

Sabadell, *Junio* de *2010*

Firmado: *Marc Talló Sendra*

Resumen

Mediante el presente documento se pretende describir el proceso de creación de una herramienta de tipo CMS (Gestor de contenidos). Concretamente, se desarrollará una herramienta que permita al usuario crear su propio sitio web sin necesidad de tener conocimientos de programación. Debido a la gran variedad de sitios web tanto en su forma como en su contenido y ante la imposibilidad de abarcar todo el espectro desde la primera versión del producto, se centrarán los esfuerzos en obtener una aplicación sencilla pero funcional y, sobretodo, escalable, para que mediante futuras actualizaciones se puedan ir incrementando sus posibilidades y capacidad.

Además de para desarrollar una herramienta útil, este proyecto ha servido para poner en práctica los conocimientos adquiridos durante la carrera (especialmente en lo referente a estructura de datos), para adquirir experiencia en la programación de aplicaciones web y para conocer de primera mano algunos de los errores comunes en la planificación de proyectos real.

INDICE

1. INTRODUCCIÓN	8
1.1 Introducción	8
1.2 Objetivos	8
1.3 Estado del arte	8
1.4 Estructura de la memoria	16
2. ESTUDIO DE VIABILIDAD	17
2.1 Introducción	17
2.2 Objetivos	17
2.3 Especificaciones	18
2.3.1 Especificaciones funcionales	18
2.3.2 Especificaciones no funcionales	21
2.3.3 Especificaciones técnicas	22
2.4 Planificación	22
2.5 Análisis de costes	24
2.5.1 Software	24
2.5.2 Hardware	24
2.5.3 Horas de trabajo	25
2.5.4 Coste total	26
2.6 Análisis de riesgos	26
2.7 Conclusiones	26
3. FUNDAMENTOS TEÓRICOS	27
3.1 Introducción	27
3.2 Arquitectura de la aplicación	27
3.3 Alternativas y tecnología utilizada	28
3.3.1 Servidor web	28
3.3.2 Sistema gestor de bases de datos	30
4. ANÁLISIS	35
4.1 Introducción	35
4.2 Tipos de usuario y ámbitos de la aplicación	35
4.3 Especificaciones funcionales	36
4.3.1 Pantalla de entrada	36
4.3.2 Administración	40
4.3.3 CMS	43
4.4 Diseño de la base de datos	53
5. DISEÑO	58
6. IMPLEMENTACIÓN	60
6.1 Descripción técnica	60
6.2 Estructura de la aplicación	60
6.2.1 Capa de presentación	60

6.2.2 Capa de negocio	65
6.2.3 Capa de datos	70
6.3 Herramienta de soporte para la implementación	73
6.4 Componentes externos utilizados	74
6.5 Instalación y configuración del servidor	77
6.5.1 Base de datos	77
6.5.2 Servidor web	79
6.5.3 Otras configuraciones	81
7. PRUEBAS	83
8. CONCLUSIONES	86
9. BIBLIOGRAFÍA	89

CAPÍTULO 1: INTRODUCCIÓN

1.1 Introducción

Cuando hace unos años Internet se popularizó y llegó hasta los hogares, la mayoría de sitios Web que la conformaban pertenecían a empresas e instituciones públicas. Estos primeros sitios Web apenas permitían la interacción del usuario y se actualizaban con muy poca frecuencia.

Con el paso del tiempo el contenido de Internet ha ido adquiriendo dinamismo y se ha hecho más accesible para todos. Hoy en día prácticamente todo el mundo tiene su sitio Web personal, sitio que se actualiza casi a diario y que permite la participación activa de los visitantes y usuarios.

Ante esta proliferación de sitios Web personales, con su necesidad de constante actualización, el concepto de desarrollo Web también ha tenido que cambiar. Se ha tenido que eliminar la dependencia del desarrollador Web. Se han tenido que crear herramientas para que el usuario no informático pueda crear y mantener su propio sitio Web. Estas herramientas se denominan Sistemas gestores de contenido (o CMS).

La complejidad de muchos sitios Web hace que siga siendo necesaria la figura del desarrollador Web, pero para la gran mayoría de páginas de carácter personal e incluso portales de pequeñas y medianas empresas, la funcionalidad que aportan estos gestores de contenido es más que suficiente, permitiendo que personas en todo el mundo, sin tener apenas nociones informáticas, puedan crear y mantener su propia página Web.

1.2 Objetivos

Se pretende desarrollar una herramienta que permita generar y gestionar sitios Web, dando especial importancia a las siguientes características:

- 1) Polivalencia: la herramienta debe ofrecer múltiples posibilidades para generar sitios Web variados tanto en su forma como en su contenido.
- 2) Accesibilidad: es muy importante que la herramienta pueda ser utilizada por el máximo número de personas, con independencia de su nivel de conocimientos informáticos.
- 3) Escalabilidad: el sistema tiene que ser lo más abierto posible, Internet está en constante evolución y el producto tiene que permitir en la medida de lo posible la incorporación de nuevas funcionalidades y mejoras.
- 4) Económicamente accesible: el uso de la herramienta desarrollada tiene que tener un coste cuanto más bajo mejor, a ser posible gratuito.

1.3 Estado del arte

Actualmente existen muchos sistemas gestores de contenido en el mercado. Se busca una herramienta económica y fácil de utilizar así que no todos ellos son alternativas válidas al software que se pretende desarrollar.

Las opciones comerciales (o “de pago”) quedan prácticamente descartadas: por lo general se trata de productos destinados al entorno empresarial y que cuestan miles de euros.

En cuanto a las alternativas de libre distribución, no todas ellas son lo sencillas que cabría esperar, algunas de las más populares requieren de una instalación adicional en el PC local, hecho que choca por completo con el segundo de nuestros objetivos: la accesibilidad. Otras son fáciles de manejar pero limitadas en tanto que todos los sitios Web que generan son muy similares entre sí. A continuación se analizan las alternativas más serias:

Drupal

Drupal es junto a Joomla! uno de los CMS que mejor relación sencillez-funcionalidad aportan. Buscando en Internet es fácil encontrar servicios de hosting gratuitos que ofrezcan la posibilidad de administrar sitios Web mediante Drupal (por ejemplo Freedrupal5hosting.com) sin necesidad de realizar ninguna instalación previa.



Drupal: Panel de administración

El menú de administración ofrece las opciones más comunes: configuración visual de la Web, gestión de contenidos, definición de usuarios y roles, etc.



Drupal: edición de contenidos

Dos grandes problemas sin duda son el idioma de la interficie y la inclusión sistemática de propaganda en la Web. El segundo se soluciona contratando un *hosting* de pago; el primero tiene más difícil solución: para utilizar Drupal en español lo mejor es instalarlo en el propio ordenador, opción que se descartó. Además, el sistema de administración en general es demasiado técnico. Sin ir más lejos, el editor de contenido que se ofrece por defecto espera que el usuario introduzca directamente código HTML o que se limite a crear páginas de texto plano. Drupal es un muy buen CMS pero no está orientado hacia el tipo de usuario que nosotros pretendemos satisfacer.

Sitio oficial: <http://drupal.org>

Joomla!

En 2005 la comunidad de desarrolladores y usuarios del CMS Mambo se fracturó porque los propietarios de la herramienta decidieron limitar las condiciones de la licencia de uso de su software. Desde entonces, unos se dedicaron a mantener Mambo mientras que otros crearon el que a día de hoy es uno de los CMS de libre distribución más utilizados y con más futuro: Joomla!

Como ocurre con Drupal, no es difícil encontrar servicios de hosting gratuitos que ofrezcan Joomla! ya instalado (por ejemplo www.freejoomlas.com). Tras un proceso de registro muy sencillo el usuario puede empezar a configurar su Web y a publicar contenido. Como ya se comentó, este tipo de servicios presentan una serie de problemas: interficie de administración en inglés, propaganda por toda la Web y limitaciones en muchos aspectos (número de visitas, espacio en disco, tamaño de los archivos...).



Más allá de estos inconvenientes, el panel de control es muy claro y amigable gracias al uso de numerosos iconos y a una distribución eficaz de las diferentes áreas y opciones. Además, en la página principal se presentan una serie de accesos directos para que el usuario ni tan siquiera necesite “bucear” por los menús para realizar las tareas más comunes.

Dentro de los parámetros de configuración, una opción bastante interesante que ofrece Joomla! es la de poder deshabilitar el sitio Web temporalmente, muy útil por ejemplo para hacer pruebas o actualizaciones cuando no se dispone de entorno de pruebas.

En cuanto a la gestión de usuarios, este CMS no permite al administrador crear sus propios perfiles: el sistema ofrece una serie de roles preestablecidos muy orientados al control y publicación de contenido (autor, editor, publicador, administrador...).



Este comportamiento impide por ejemplo publicar un listado con precios y definir con exactitud qué usuarios podrán acceder a él, tarea muy necesaria en sitios Web de carácter más empresarial.

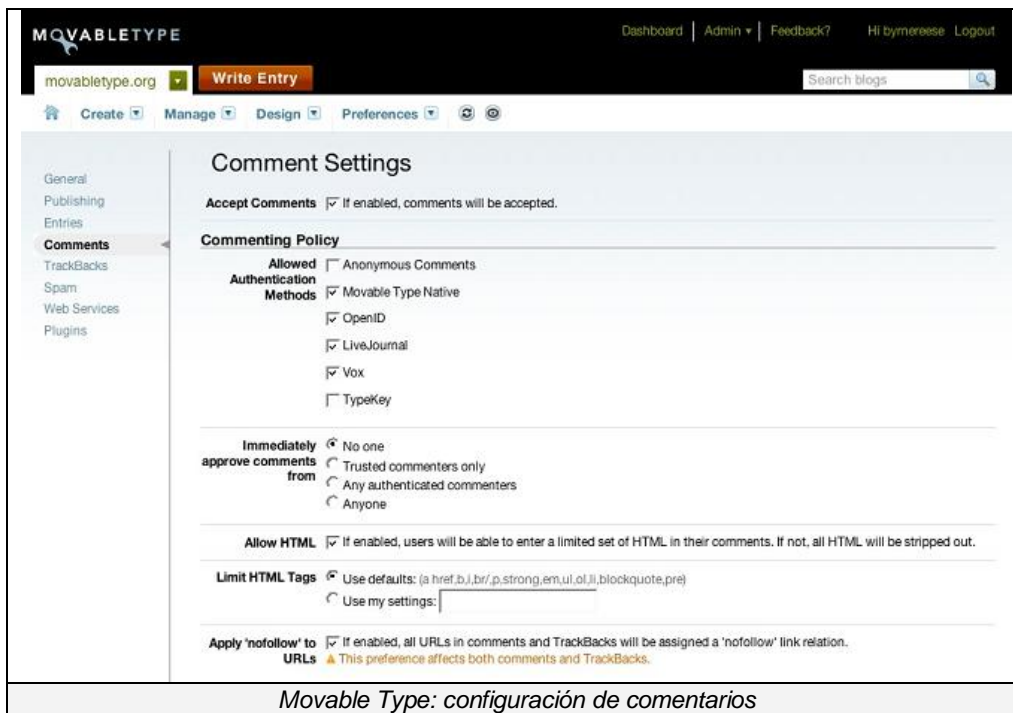
La edición de contenidos es muy completa (apta tanto para los usuarios noveles como para los más avezados) y la gestión del diseño ofrece muchas posibilidades (existen decenas de plantillas para cambiar el aspecto visual del sitio Web, con la interesante característica de poder establecer para cada elemento de la página el estilo de una plantilla diferente).

En general las opciones y el manejo resultan bastante satisfactorios por lo que Joomla! constituye una alternativa realmente interesante para quienes deseen administrar un sitio Web tipo *Blog* (*) o de publicación de noticias. Si se precisa de más funcionalidad habrá que barajar otras posibilidades.

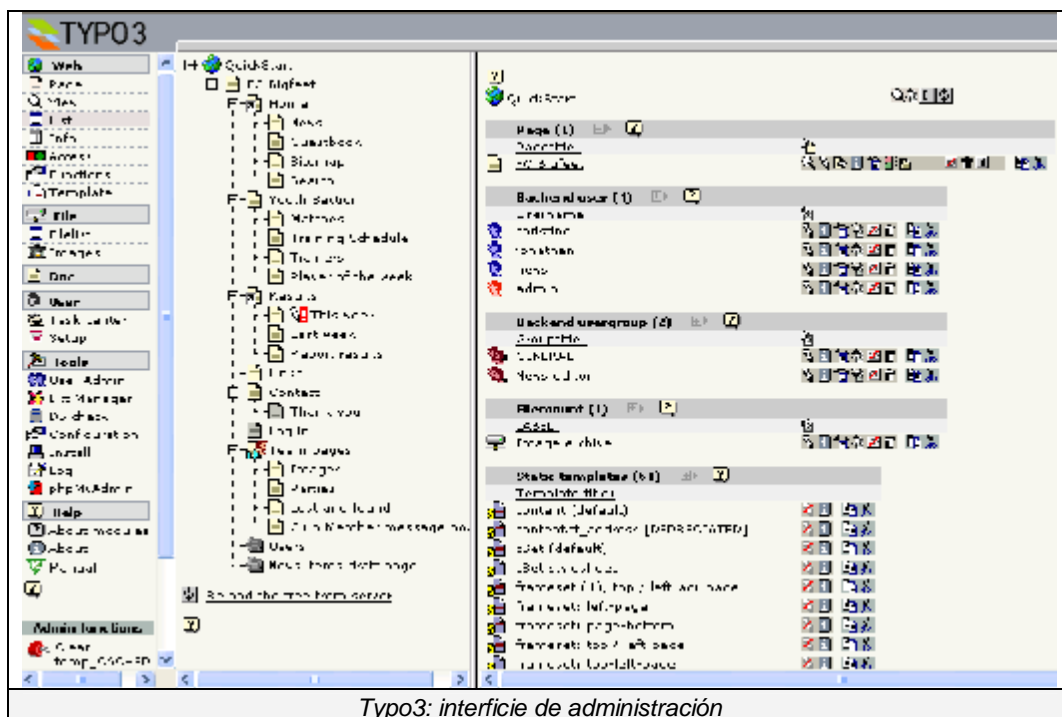
Sitio oficial: <http://www.joomla.org>

Typo3 y Movable Type

Herramientas muy potentes pero sobretodo orientadas al mundo corporativo. Tanto las interficies de administración como la gran mayoría de información que se encuentra en Internet están en inglés. Además, Movable Type no es estrictamente libre, hay que pagar a partir de la versión *Business* lo que significa que utilizando versiones inferiores nuestro horizonte está limitado.



Por si esto fuera poco y, debido a los requerimientos que presentan ambos sistemas y al tipo de usuarios a los que van destinados, encontrar servicios de hosting gratuitos con las herramientas instaladas y preparadas para trabajar es prácticamente imposible. Para Typo3 hay algunas posibilidades, pero el periodo gratuito está limitado en el tiempo; para Movable Type hay que pagar.



Precio, idioma y complejidad: estos dos CMS no son una alternativa a considerar en nuestro caso.

Sitios oficiales: <http://typo3.com> y <http://www.movabletype.org>

Zikula

Hasta hace poco una de los más célebres CMS fue sin duda PHP-Nuke. A partir de éste y con el propósito de solucionar problemas de eficiencia y sobretodo de seguridad se creó PostNuke. A mediados de Junio de 2008 PostNuke adoptó el nombre de Zikula. PHP-Nuke sigue existiendo aunque muchos servicios de *hosting* se niegan ya a alojar sitios Web gestionados con el mismo debido a los problemas antes mencionados.

Encontrar servicios de *hosting* gratuitos que suministren PostNuke/Zikula preinstalado es muy difícil, esto implica que el usuario debe instalar el CMS en su propio ordenador y subir posteriormente el sitio Web a un hosting con soporte PHP-MySQL. En nuestro caso ésta es razón suficiente para desestimar el uso de Zikula.

Sitio oficial: <http://zikula.org>

WordPress

Lo primero que destaca en este gestor de contenidos es su accesibilidad, no hay más que ver el formulario de registro de la página oficial:

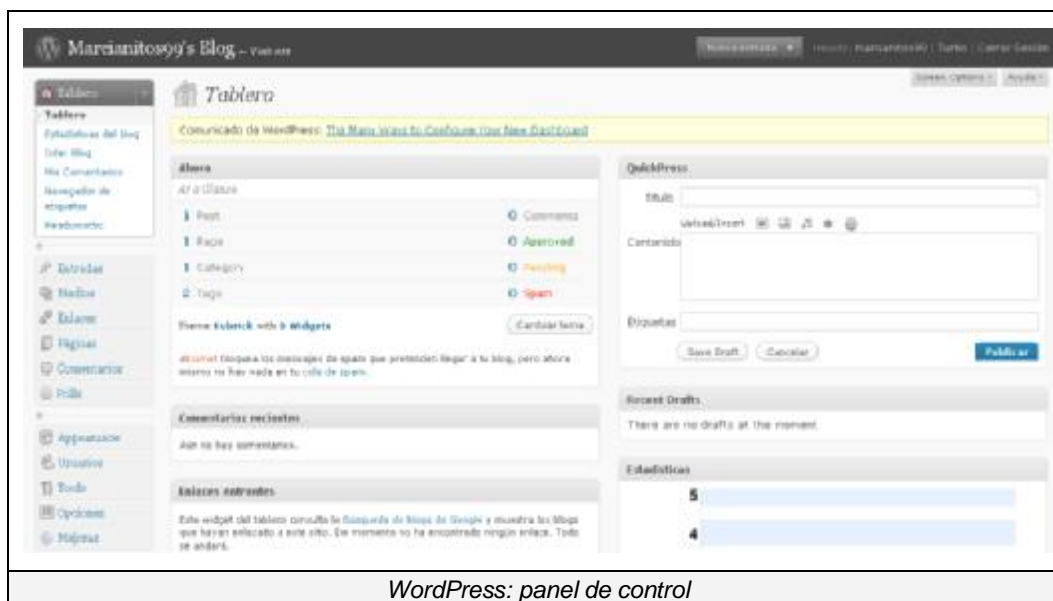
Consiga su propia cuenta WordPress en segundos

¡Llene este formulario de un paso y estará blogueando segundos después!

Nombre de usuario	<input type="text" value="Marcianitos99"/>
	<small>¡No olvide usar un nombre de usuario de al menos 4 caracteres!</small>
Contraseña	<input type="password" value="*****"/>
Confirmar	<input type="password" value="*****"/>
	<small>¡Compruebe las coincidencias, mínimo 6 caracteres y al menos una mayúscula y una minúscula! Password Strength</small>
	<input type="button" value="BARD"/>
Dirección de Correo Electrónico	<input type="text" value="ricard.blasco@gmail.com"/>
	<small>¡No olvide verificar la instalación de correo electrónico si no dispone de correo electrónico!</small>
Lo legal	<input type="checkbox"/> <small>Estoy de acuerdo con los términos de servicio</small>
	<input type="checkbox"/> <small>¡Venga, cámbale el blog a otro idioma o modifique el comportamiento!</small>
	<input type="checkbox"/> <small>¡No olvide leer el manual de usuario, por favor!</small>

WordPress: formulario de registro

Durante el registro se pueden configurar de manera muy fácil el idioma de la interficie de administración y el comportamiento del sitio Web respecto a los principales buscadores.



En general las opciones que ofrece WordPress son sencillas pero suficientes: estadísticas básicas (palabras buscadas, enlaces seleccionados, entradas más visitadas...), configuración del diseño por temas, uso de *Widgets* (*) (incrementando con ello la flexibilidad funcional del CMS) y una gestión y edición de contenidos muy simple y directa.

Claramente WordPress es una herramienta orientada a todos los públicos: desde la página oficial, de forma gratuita y con un par de clicks, el usuario puede estar ante el panel de control en español de su sitio Web.

Además de la no necesidad de buscar *hosting* gratuito, este CMS ofrece otras características interesantes: compatibilidad con la innovadora tecnología Gears (*), módulo anti-spam Akismet (*) incorporado y la posibilidad de Importar contenido desde otros CMS (como Movable Type o Blogger por ejemplo).

El principal inconveniente de WordPress está muy relacionado con su principal virtud: al ser tan sencillo sus posibilidades son bastante limitadas. La opción de incorporar Widgets atenúa esta carencia pero en este caso también hay problemas: WordPress no permite el uso de *scripts de usuario* (*) en los sitios Web alojados en sus servidores.

La gestión de usuarios y perfiles es prácticamente testimonial: por defecto sólo se ofrecen 4 roles, que pueden ser asignados a otros usuarios de WordPress para permitir que colaboren en la edición de contenidos de nuestro sitio Web.

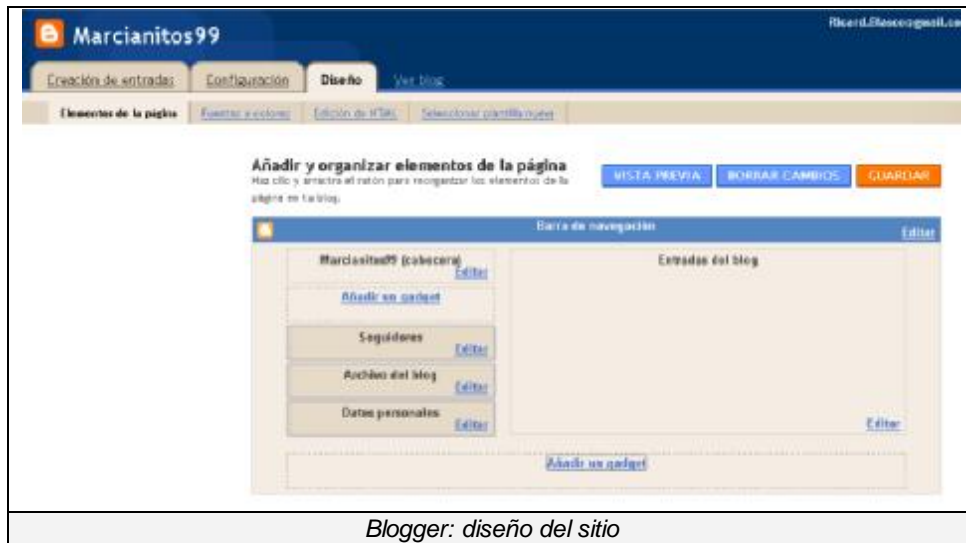
Otra pega importante es la lentitud del servicio en general y de la administración en particular: gestionar los contenidos del sitio Web se puede convertir en una tarea bastante desesperante. Se soluciona alojando la Web en un servidor de pago, pero esta no es una opción en nuestro caso.

WordPress es un CMS muy asequible, tanto por su precio como por su sencillez, dirigido a la generación de sitios Web de tipo personal basados en la publicación de contenido a base de texto e imágenes. Demasiado simple para nuestro propósito.

Sitio oficial: <http://es.wordpress.com>

Blogger

Blogger es la herramienta que ofrece Google para la creación de páginas Web personales. Siguiendo la misma línea que WordPress, el servicio es totalmente gratuito y si cabe aún más sencillo en todos sus aspectos.



La administración y publicación de contenidos son muy sencillas, tanto que todos los sitios Web alojados en Blogspot parecen prácticamente el mismo. Es la alternativa más sencilla pero también la más básica y limitada.

Sitio oficial: <http://www.blogger.com>

DotNetNuke

Destaca como principal característica de este CMS el hecho de que esté implementado mediante tecnología Microsoft y que por lo tanto permita trabajar bajo entorno ASP.NET (*) y utilizar SQLServer (*) como sistema gestor de base de datos. Como los presentados hasta ahora, este producto es gratuito, de código abierto y libre distribución.

Sin embargo DotNetNuke presenta dos grandes problemas:

- 1) el usuario puede utilizar el software libremente pero encontrar un servicio de *hosting* gratuito para alojar una Web que utiliza IIS (*), SQLServer y ASP.NET es prácticamente una quimera.
- 2) la instalación del entorno necesario en un ordenador local es demasiado compleja incluso para usuarios con conocimientos informáticos.

En nuestro caso éstas son dos razones de peso para descartar DotNetNuke.

Conclusiones

Tras este análisis se pueden obtener varias conclusiones:

- 1) Los usuarios sin los conocimientos informáticos necesarios para realizar complejas instalaciones en su equipo y con pocas nociones de inglés se tienen que limitar a construir sitios Web más bien sencillos y parecidos entre ellos.
- 2) La gran mayoría de los gestores de contenido disponibles han sido desarrollados y funcionan sobre entorno LAMP (Linux, Apache, MySQL, PHP *), las alternativas en este sentido son muy escasas.

Por estas razones y para eliminar la dependencia que puede generar el uso de herramientas desarrolladas y mantenidas por terceros, se considera una buena opción la implementación de nuestro propio CMS.

1.4 Estructura de la memoria

Este documento estructura sus contenidos según el siguiente esquema:

- **Introducción:** se citan los objetivos generales del proyecto y se detalla el estudio de productos similares, realizado con el fin de buscar carencias funcionales y obtener ideas.
- **Estudio de viabilidad:** detalle de los objetivos a cumplir, estimación de costes para llevar a cabo el proyecto, planificación temporal y conclusiones en referencia a la viabilidad del proyecto.
- **Fundamentos teóricos:** bases teóricas sobre las que se asenta la aplicación. Descripción de la arquitectura general de la misma. Discusión de las alternativas tecnológicas disponibles para la implementación. Selección del marco de trabajo definitivo.
- **Análisis:** definición de los ámbitos de la aplicación. Definición de los tipos de usuario de la aplicación. Explicación detallada de cada uno de los requisitos funcionales de la aplicación, incluyendo diagramas de casos de uso y de flujo. Presentación del diseño de la base de datos y explicación detallada del mismo.
- **Diseño:** criterios de diseño de la aplicación. Tipos de pantalla. Formato de los campos. Objetivos del diseño.
- **Implementación:** descripción detallada de las tres capas que forman la aplicación, de la relación entre ellas y de sus componentes. Descripción de la herramienta desarrollada para dar soporte durante la fase de implementación del proyecto. Enumeración de los componentes externos utilizados para cubrir ciertas funcionalidades. Detalle del proces de instalación de la aplicación en un servidor de producción.
- **Pruebas:** resumen de los procesos de pruebas a los que ha sido sometida la aplicación. Problemas que se han encontrado y sus posibles soluciones.
- **Conclusiones:** análisis de las desviaciones sufridas por el proyecto respecto a la planificación inicial. Futuras mejoras. Valoración final.

CAPÍTULO 2: ESTUDIO DE VIABILIDAD

2.1 Introducción

En Internet existe gran variedad de sitios: páginas corporativas, tiendas virtuales, foros públicos, casas de apuestas... La creación y mantenimiento de estas Web suele ser tarea de equipos de profesionales, desarrolladores Web experimentados y con grandes conocimientos técnicos.

Sin embargo existen otro tipo de páginas Web: las páginas Web personales o *Blogs*. Éstos son sitios mucho más sencillos, a menudo gestionados por el propio usuario mediante herramientas conocidas como sistemas gestores de contenido (o CMS). Estas herramientas permiten a cualquier usuario, independientemente de sus conocimientos a nivel de programación Web, crear y actualizar su página Web de forma fácil y económica.

Mediante la consecución del presente proyecto se pretende desarrollar un sistema gestor de contenidos accesible a cualquier tipo de usuario y que sea capaz de generar sitios Web de apariencias, contenidos y estructuras muy variadas, incrementando con ello las posibilidades que a día de hoy ofrecen los CMS actuales para el desarrollo de páginas Web de carácter personal.

2.2 Objetivos

El objetivo principal de este proyecto consiste en crear un sistema que permita generar y mantener sitios Web de carácter personal. Los usuarios que deseen utilizarlo deberán registrarse. Tras ello, podrán acceder a la interficie de administración de su propio sitio Web. Esta interficie debe permitir al propietario del sitio Web realizar las siguientes tareas:

- Configurar la estructura del sitio (disposición de los elementos en la pantalla). En este aspecto el sistema tiene que ser muy flexible (de lo contrario sucederá lo que sucede con la mayoría de CMS orientados a la creación de *Blogs*: todas sus Web se parecen entre sí).
- Permitir incluir contenidos de tipo variado: páginas con entradas tipo noticias, páginas de contenido general, imágenes, encuestas, menús...
- Gestionar el acceso al sitio Web mediante usuarios y perfiles. El propietario del sitio Web debe poder definir tantos perfiles de acceso como quiera y para cada uno de ellos podrá determinar el aspecto de la Web, los contenidos y las acciones que se le permite realizar. A su vez, también deberá poder gestionar los usuarios y determinar con qué perfil accede cada uno al sitio Web. El sistema también permitirá generar sitios Web en los que el acceso anónimo no esté permitido.
- Consultar indicadores y estadísticas que le ayuden a conocer aspectos como las páginas de su Web más visitadas o los usuarios más activos.
- Realizar copias de seguridad de su sitio Web y poder restaurarlas en caso de ser necesario.
- Acceder a un sistema de ayuda que le permita conocer de forma rápida y clara cómo se llevan a cabo las tareas principales.

El funcionamiento del sistema tiene que ser claro y lo suficientemente intuitivo como para que incluso los usuarios sin nociones de desarrollo Web puedan utilizarlo.

2.3 Especificaciones

2.3.1 Especificaciones funcionales

Administración y usuarios
La página principal de la aplicación permitirá identificarse como usuario, registrarse como nuevo usuario o solicitar una nueva contraseña en caso de pérdida.
La aplicación presentará dos tipos de usuario: los administradores de la aplicación y los usuarios del CMS. Tras identificarse en la página principal, se accederá a una u otra interficie de según el tipo de usuario.
Los administradores de la aplicación gestionarán a los usuarios del CMS, podrán consultar las estadísticas que afecten a todo el sistema y atender a las peticiones que los usuarios del CMS les hayan mandado mediante la opción "contactar con los administradores".
La gestión de los usuarios CMS permitirá altas, ediciones y bajas. También se podrá bloquear y desbloquear usuarios y activar usuarios que hayan emitido una solicitud de registro desde la página principal.
Las estadísticas que los administradores podrán consultar se referirán al número de Sitios Web virtuales creados, espacio de disco ocupado por cada sitio y los tipos de página y tipos de módulo utilizados. Estas estadísticas deberán permitir al administrador detectar qué características son las más y las menos utilizadas por los usuarios CMS, para poder así concentrar los esfuerzos en los aspectos que más interés tengan.
Cuando se atiendan las peticiones de los usuarios CMS, se marcarán estas peticiones como "atendidas" y opcionalmente se podrá responder al usuario desde la misma pantalla mediante un sencillo formulario.
Tras identificarse en la página principal, los usuarios del CMS accederán a una interficie que les permitirá gestionar su sitio Web virtual. Las tareas que se podrán llevar a cabo desde esta interficie se detallan a continuación.

Gestión CMS: Distribuciones
El usuario podrá definir tantas distribuciones de pantalla para su sitio Web como desee (<i>Layouts</i>). Para cada distribución se definirá el nombre y una descripción.
Siempre habrá una distribución por defecto (solo una), que será la que verán los visitantes para cuyo perfil de acceso no se haya definido ninguna distribución concreta. El usuario podrá especificar cuál es la distribución por defecto de entre todas las que tenga creadas. El sistema no permitirá eliminar la distribución por defecto.
Cuando se cree un nuevo sitio Web virtual, se creará automáticamente una distribución por defecto.
Cuando se defina una distribución, se podrá escoger entre utilizar marcos (o <i>frames</i>) y no utilizarlos.
Si no se utilizan marcos, se tendrá que especificar una página principal.
En caso de utilizar marcos, se podrá dividir la pantalla hasta en 7 zonas. El marco central siempre será visible, los 4 laterales, el superior y el inferior podrán mostrarse u ocultarse. Para cada marco visible el usuario deberá definir la página Web a mostrar.
El usuario podrá configurar el tamaño de los marcos de forma fácil e intuitiva, sin tener que especificar medidas en píxeles ni nada por el estilo.
En las distribuciones con marcos, el usuario podrá definir el estilo de los bordes.

Gestión CMS: Contenidos
El contenido de un sitio Web virtual se presentará en forma de páginas y módulos. Las páginas serán el equivalente a páginas Web, los módulos serán bloques de código con funcionalidades específicas y que se mostrarán en el sitio Web incrustados en páginas contenedoras de módulos.
El usuario podrá gestionar todas las páginas de su Web de forma intuitiva y clasificarlas en carpetas tipo explorador de Windows. Se le permitirá crear, editar y eliminar carpetas, y mover elementos (páginas y carpetas) de una carpeta a otra.
Se podrán crear diferentes tipos de página según su funcionalidad: <ul style="list-style-type: none"> - páginas de contenido general - páginas de noticias - páginas contenedoras de módulos <p>El sistema será lo suficientemente abierto como para poder ir incorporando nuevos tipos de página. Cada tipo de página presentará una serie de acciones posibles a realizar sobre ella, los permisos sobre estas acciones se definirán a nivel de perfil de acceso.</p>
Cuando se cree una página, se especificará obligatoriamente el nombre, tipo y la ubicación (carpeta).
Tras crear una página, se podrá definir su contenido, que variará en función del tipo de página.
Tras crear una página también se podrán definir los permisos de acceso por perfil (permitido o denegado según perfil).
Para las páginas de contenido general, el usuario dispondrá de un editor tipo WYSWYG (arrastrar y soltar). Este editor permitirá escribir texto con formato (tipo de letra, colores, tamaño...), crear tablas, incrustar imágenes y enlaces. Durante la edición de páginas de carácter general se dispondrá de la opción de previsualizar la página, para que el usuario no tenga que guardar e ir a su sitio Web cada vez que desee comprobar cómo queda la página.
Para las páginas de tipo contenedor de módulos, el programa presentará al usuario la pantalla dividida en cuadrículas y éste sólo tendrá que arrastrar los módulos hasta la posición en la que desee que aparezcan. Un mismo módulo solo podrá aparecer una vez por página.
Para las páginas de tipo noticias, el usuario especificará el número de noticias por página y el formato de las mismas, pudiendo ocultar o mostrar campos según desee (título, fecha, autor y etiquetas). También se permitirá al usuario decidir si las noticias admiten comentarios por parte de los visitantes o no. Para cada noticia se podrán asociar hasta 3 etiquetas, que servirán para poder realizar búsquedas y filtros de noticias desde la Web.
Las páginas tipo noticias permitirán definir los siguientes permisos a nivel de perfil de usuario: <ul style="list-style-type: none"> - <u>Inserción de noticias</u>: los usuarios con este perfil podrán insertar nuevas entradas en las páginas de noticias y editar las entradas que ellos mismos hayan insertado con anterioridad (el sistema guardará las fechas de creación y de última edición). - <u>Edición de noticias</u>: los usuarios con este perfil podrán modificar las entradas de las páginas de noticias aún no siendo ellos los autores de las noticias. - <u>Inserción de comentarios</u>: los usuarios con este perfil podrán insertar comentarios en las noticias (siempre que el administrador del CMS haya configurado la página de noticias para permitir los comentarios). Estos usuarios también podrán editar los comentarios que hayan sido escritos por ellos mismos con anterioridad (el sistema guardará las fechas de creación y de última edición). - <u>Edición de comentarios</u>: los usuarios con este perfil podrán modificar comentarios escritos por otros usuarios.
Inicialmente el usuario podrá crear cinco tipos de módulo: <ul style="list-style-type: none"> - módulos tipo Menú

- módulos tipo Login
- módulos tipo Imagen
- módulos tipo Encuesta
- módulos tipo Búsqueda

El sistema será lo suficientemente abierto como para poder ir incorporando nuevos tipos de módulo. Cada tipo de módulo presentará una serie de acciones posibles a realizar sobre él, los permisos sobre estas acciones se definirán a nivel de perfil de acceso.

Los módulos de tipo Menú servirán para mostrar menús en el sitio Web. Para cada menú, el usuario decidirá si es horizontal o vertical y podrá añadir tantas entradas como desee. Para cada entrada menú se especificará el nodo del que colgará (se podrán crear menús de la profundidad que se desee), el orden en el que se mostrará y el enlace al que conducirá. Este enlace podrá ser una URL, una página del propio sitio Web o ninguno (para entradas de menú con subentradas).

Los módulos de tipo Login aportarán la funcionalidad de identificación de usuarios en el sitio Web. Si el administrador del sitio Web virtual desea utilizar la gestión de usuarios necesitará incluir un módulo de tipo Login para que éstos se puedan identificar. Un sitio Web virtual sin módulo de Login sólo tendrá visitantes de tipo Anónimo. El módulo de Login permitirá identificarse, cerrar sesión, solicitar nueva contraseña y registrarse como nuevo usuario. Esta última característica podrá deshabilitarse cuando se configure el módulo de tipo Login.

Los módulos de tipo Imagen permitirán mostrar imágenes en las páginas contenedoras de módulos. La imagen podrá estar alojada en el propio sitio Web virtual o en Internet (URL). El usuario podrá modificar el tamaño con el que el módulo de Imagen muestra la imagen. La principal finalidad de este tipo de módulos será la inclusión de banners en el sitio Web.

Los módulos tipo encuesta permitirán realizar consultas. Estos módulos constarán de una pregunta y hasta 5 posibles respuestas. Tanto la acción de responder como la de consultar resultados se podrán regular a nivel de permisos por perfil. Se deberá controlar que un mismo usuario no conteste más de una vez a la misma encuesta.

Los módulos de tipo búsqueda permitirán buscar noticias y contenidos dentro del sitio Web virtual. La búsqueda simple sólo buscará por el título de la noticia o página, la búsqueda avanzada permitirá buscar por título, contenido o etiquetas. El administrador podrá deshabilitar la búsqueda avanzada durante la configuración del módulo de Búsqueda.

Además de páginas y módulos, el usuario del CMS podrá definir el aspecto de páginas especiales de su sitio Web virtual como son la página de acceso denegado o la de registro de nuevo usuario.

El sistema permitirá al usuario del CMS subir imágenes a su sitio Web virtual. Estas imágenes serán las que luego utilizará en sus páginas Web y módulos. La gestión de estas imágenes será fácil y clara. El sistema controlará que no se exceda un máximo de tamaño fijado por un administrador.

Gestión CMS: Control de acceso

Cuando se cree un sitio Web virtual se creará automáticamente un perfil denominado Anónimo. Este perfil no se podrá eliminar y será el que se le asignará a todos los visitantes del sitio Web hasta que no utilicen un módulo de tipo Login para identificarse.

El usuario del CMS podrá crear tantos perfiles de acceso como desee. Para cada perfil definirá un nombre, una descripción y opcionalmente la distribución (o *Layout*) asociada al perfil. Si no se define una distribución, los usuarios del perfil accederán a la Web mediante la distribución por defecto.

Para cada perfil también se definirá a qué páginas pueden y no pueden acceder los usuarios que lo utilicen, así como las acciones que pueden realizar sobre cada tipo de página (por ejemplo, se podrá definir que los usuarios de cierto perfil, en las páginas de noticias, puedan insertar noticias).

El usuario CMS podrá crear tantos usuarios de su sitio Web como desee. Estos usuarios deberán utilizar un módulo de tipo Login para identificarse. Para cada usuario se almacenará el nombre, email, login, password, fecha de alta, estado y perfil. Los estados serán: activo,

bloqueado, pendiente de activación y pendiente de cambio de contraseña. Los pendientes de activación serán usuarios nuevos que han solicitado registrarse en la Web desde el módulo de Login. Los pendientes de cambio de contraseña serán los que han solicitado una nueva contraseña desde el módulo de Login.

Gestión CMS: Logs y estadísticas

El usuario del CMS podrá consultar una serie de estadísticas que le permitan ver cuáles son las páginas más visitadas de su Web, los usuarios más activos, la actividad por fecha y el espacio utilizado y disponible. Estas estadísticas se mostrarán en forma de tabla y en formato gráfico.

Gestión CMS: Copias de seguridad

El usuario del CMS realizar en cualquier momento una copia de seguridad de su sitio Web para poder restaurarla posteriormente. La copia implicará copiar todas las páginas y módulos, usuarios, perfiles y permisos, y las distribuciones que haya definido. Cada usuario podrá almacenar un máximo de 5 copias de seguridad de su sitio Web virtual.

Gestión CMS: Ayuda online

El usuario del CMS podrá acceder desde el menú a una sencilla documentación que le explique los conceptos básicos del funcionamiento de la herramienta y le muestre con ejemplos cómo realizar las tareas comunes: crear una distribución de pantalla, crear una página de carácter general, una de noticias y un contenedor de módulos; crear módulos de tipo menú y login; configurar las pantallas especiales tipo "acceso denegado" o "registro nuevo usuario"; crear usuarios y perfiles y configurar permisos; realizar copias de seguridad y restauraciones; y consultar las estadísticas de su sitio Web.

Gestión CMS: Contacto con administradores

Por último, la interficie de uso del CMS también ofrecerá la opción de contactar con los administradores. Se podrán enviar peticiones a los administradores del sistema mediante un formulario simple tipo motivo y descripción, y se podrán leer sus respuestas en caso de haberlas.

2.3.2 Especificaciones no funcionales

La aplicación deberá ser 100% compatible con los navegadores mayoritarios: Internet Explorer (versión 6 o superior) y Mozilla Firefox (versión 3 o superior).

La interficie de administración del CMS tiene que ser clara y sencilla: el menú se situará en la parte izquierda de la pantalla y no tendrá más que un nivel de profundidad; en la parte superior de la pantalla se mostrará el usuario conectado y se situarán dos botones, uno para cerrar la sesión y otro para abrir el sitio Web que se está administrando en una ventana externa.

La resolución de pantalla mínima para utilizar el CMS será de 1024 x 768. La resolución de pantalla aconsejable para visualizar los sitios Web generados dependerá del contenido de los mismos.

La interficie de administración del CMS podrá visualizarse en varios idiomas. El idioma de los sitios Web generados dependerá del contenido que muestren y por lo tanto dependerá del propietario del sitio.

Al usuario no se le requerirán en ningún momento conocimientos de programación Web. Todas las pantallas serán lo más sencillas, visuales y claras posible.

2.3.3 Especificaciones técnicas

La aplicación a desarrollar tiene que ser de tipo Web.
Se utilizará el entorno de programación Microsoft .NET Framework, concretamente el lenguaje C#, por lo tanto para su funcionamiento se requerirá un servidor con sistema operativo Windows y servidor de aplicaciones IIS.
La aplicación utilizará una base de datos relacional, concretamente Microsoft SQLServer, que es la que se corresponde con el entorno de programación seleccionado.
La aplicación se deberá estructurar en una arquitectura de capas: presentación, negocio y datos.
El control de transacciones se realizará a nivel de la capa de negocio.
El control de errores incluirá un registro o <i>log</i> , donde se anotarán todos los errores en tiempo de ejecución que pueda llegar a producir la aplicación.

2.4 Planificación

El proyecto se llevará a cabo bajo el modelo de desarrollo tradicional (o *en cascada*): cada etapa se empezará cuando la predecesora haya terminado. El mayor inconveniente de esta metodología es el hecho de que introducir cambios posteriormente al desarrollo resulta muy costoso (hay que volver a pasar por todos los eslabones de la cadena de etapas). Se ha optado por este tipo de desarrollo por dos motivos principales:

- 1) No se dispone de personal suficiente como para avanzar en varios procesos paralelamente.
- 2) Se parte de un listado de control cerrado, en el que se detallan los requisitos a cumplir por el programa. Durante el desarrollo del proyecto no se admitirán nuevas características no contempladas en dicho listado de control.

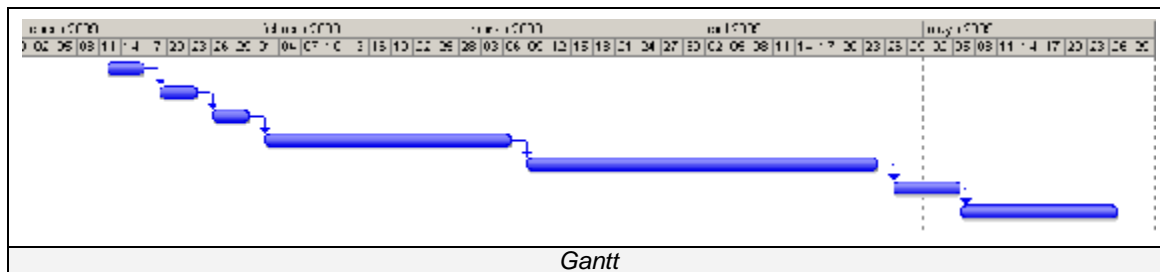
Concretamente el desarrollo del proyecto se dividirá en las siguientes 7 etapas:

- 1) Análisis de productos similares: ante el reto de desarrollar un gestor de contenidos propio, el primer paso será analizar algunos de los CMS más populares para obtener ideas tanto funcionales como de diseño.
- 2) Estudio de viabilidad: se planificarán los trabajos y valorará el coste del proyecto, con el fin de decidir si vale la pena el desarrollo del mismo.
- 3) Elaboración del listado de control (o análisis de requerimientos): en base a las necesidades que se quieren cubrir, se redactarán una serie de objetivos y características que el programa deberá cumplir. Este listado servirá de "hoja de ruta" durante el desarrollo del proyecto.
- 4) Diseño de la aplicación: planificación de la estructura de la aplicación en base a los objetivos establecidos. Se definirán las entidades necesarias, las relaciones entre ellas y los procesos a nivel de negocio que permitirán implementar los algoritmos de solución a los problemas de partida.
- 5) Implementación: codificación del diseño mediante el entorno de programación seleccionado.

- 6) Pruebas y desarrollos correctivos: más allá de las pruebas asociadas al proceso de implementación se dedicará una etapa exclusivamente al *testeo* de la aplicación. El objetivo será detectar posibles errores en tiempo de ejecución y deficiencias funcionales y de seguridad. En caso de ser necesario se llevarán a cabo los correspondientes desarrollos correctivos.
- 7) Elaboración de la memoria del proyecto: documentación de todo el proceso de desarrollo del proyecto.

	i	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Análisis de productos similares	5 días	lun 12/01/09	vie 16/01/09	
2		Estudio de viabilidad	5 días	lun 19/01/09	vie 23/01/09	1
3		Elaboración del listado de control	5 días	lun 26/01/09	vie 30/01/09	2
4		Diseño	25 días	lun 02/02/09	vie 06/03/09	3
5		Implementación	35 días	lun 09/03/09	vie 24/04/09	4
6		Pruebas y desarrollos correctivos	7 días	lun 27/04/09	mar 05/05/09	5
7		Elaboración de la memoria del proyecto	15 días	mié 06/05/09	mar 26/05/09	6

Etapas del proyecto



Para elaborar la planificación se ha estimado que el tiempo de dedicación medio al proyecto por día será de 3 horas y se han excluido los fines de semana. Así, la previsión del coste en horas para la consecución del mismo queda como sigue:

Perfil	Actividad	Horas
Técnico júnior	Sondeo de mercado y estudio de productos similares.	15
Técnico sénior	Análisis de requerimientos y elaboración listado de control del proyecto.	15
Responsable del proyecto	Estudio de viabilidad, control de procesos y confección de la memoria.	60
Analista	Diseño modelo de base de datos y aplicación en base al listado de control.	75
Programador	Implementación en base al diseño.	105
Tester	Pruebas de la aplicación, elaboración de <i>feedback</i> para el equipo de desarrollo.	21
		291

2.5 Análisis de costes

Se clasificarán los costes de desarrollo del proyecto en tres categorías:





- costes de software
- costes de hardware
- costes por horas de trabajo

2.5.1 Software

Como se verá más adelante en la discusión de la tecnología a utilizar (capítulo 3 de esta memoria), para la realización del proyecto se requerirá el siguiente software:

- Sistema operativo Windows XP Home (1 licencia)
- Visual Web Developer 2005 Express Edition
- MS SQL Server 2005 Express Edition

De los tres, el único por el que se tendrá que pagar será por el sistema operativo. Por Internet se puede adquirir Windows XP Home (Service Pack 3) a un precio de unos 90€

Windows XP Home		
Tienda	Enlace	Precio
PrecioDiez.com - http://www.preciodiez.com		90,00 €
GangaComputer.com - http://www.gangacomputer.net		95,42 €
Solmicra.es - http://www.solmicra.es		99,90 €
Newegg.com - http://secure.newegg.com		70,21 €
(el precio incluye IVA y envío, la última oferta es sensiblemente más barata puesto que el sitio opera en dólares USA)		

2.5.2 Hardware

En cuanto al hardware, se adquirirá un ordenador de sobremesa dedicado exclusivamente a este proyecto. Las especificaciones técnicas no tienen por qué ser elevadas, los requerimientos del sistema operativo seleccionado son muy asequibles:

- Procesador de 233 MHz
- 63 MB de RAM
- 1.5 GB libres en el disco duro
- Adaptador de video y monitor VGA
- CD-ROM o DVD

Para comprar el equipo se han cotejado ofertas de dos tiendas: PC Coste (<http://www.pccoste.es>) y Datatech informática (<http://www.melollevo.es>). Desde el principio se ha apostado por una de estas dos tiendas por tres motivos:

- Su sitio Web permite la configuración online del equipo, pudiendo definir exactamente los componentes del mismo y ajustar el precio.
- No es venta exclusiva por Internet, se puede pasar a recoger el material por la tienda y ahorrar en gastos de envío.
- Ya se ha comprado en ambas tiendas anteriormente y con resultados muy satisfactorios.

Finalmente se decide comprar el siguiente equipo en Datatech informática:

Modelo: Eco Placa Base: Placa Base Asus M2N-MX SE PLUS Procesador: AMD X2 4200 AM2 Doble Nucleo 64Bits Ventilador CPU: Ventilador Original AMD Memoria: Memoria 2GB DDR2 667Mhz Kingston Caja: Caja Asus TA-86 Negra 450W Ventilador Caja: Ventilador Caja Normal No incluye Ventilador D.Duro Fuente de Alimentación: Fuente 450W Standar Tarjeta gráfica: NVIDIA GeForce6 256MB Integrada Share Tarjeta de audio: Sonido 7.1 Integrado Alta Definición Altavoces: Ninguno Sinto. TV: Ninguno Monitor: Monitor TFT 19 ASUS VW193D-B Sin Altavoz Disco duro: Disco Duro 320GB Sata II Seagate/Samsung/WD 2 Disco Duro: Ninguno Óptico 1: Regrabadora DVD Samsung SATA L.Scribe D.Capa Óptico 2: Ninguno Disketera: Ninguno Lector Tarjetas: Ninguno Tarjeta de red: Tarjeta De Red 10/100 Integrada Wireless: ninguno Firewire: Ninguno Bluetooth: Ninguno Teclado y ratón: Teclado Y Raton Logitech Con Cable PS2 Sistema operativo: Windows XP Home Instalado y Configurado
PRECIO TOTAL CON IVA: 498€

Es importante destacar que se ha decidido comprar el sistema operativo junto al ordenador, con lo que el precio de la licencia ha salido finalmente por 78,88 €, más barato que cualquiera de las ofertas analizadas por separado.

2.5.3 Horas de trabajo

Según la planificación propuesta en este estudio, el coste en horas requerido para la consecución del proyecto sería el siguiente:

Perfil	Horas	€/hora	Total
Técnico júnior	15	9,00	135,00 €
Técnico sénior	15	15,00	225,00 €
Responsable del proyecto	60	50,00	3000,00 €
Analista	75	40,00	3000,00 €
Programador	105	20,00	2100,00 €
Tester	21	0,00	0,00 €
			8460,00 €

2.5.4 Coste total

La siguiente tabla muestra el coste real que comportará el desarrollo de este proyecto:

Concepto	Importe
Hardware y software	489,00 €
Sueldo personal implicado	8460,00 €
TOTAL	8949,00 €

2.6 Análisis de riesgos

La principal amenaza para la consecución del proyecto es sin lugar a dudas la posible falta de dedicación al mismo por tener que ocuparse de otros proyectos. Este riesgo es especialmente crítico porque en gran medida depende de personas ajenas al proyecto y por lo tanto puede ser controlado sólo hasta cierto punto.

Otro riesgo muy a tener en cuenta en este tipo de proyectos es el hecho de que el listado de control esté mal confeccionado. Debido al modelo de desarrollo en cascada por el que se ha optado, la detección de requerimientos (sobre todo funcionales) a mitad del proyecto implicaría tener que volver a repetir gran parte de los procesos ya realizados.

Finalmente también cabe destacar el hecho de que el equipo de desarrollo tiene poca experiencia en el ámbito de la seguridad de sitios Web. Esto implica que el producto resultante, por lo menos en sus primeras versiones, puede llegar a ser bastante vulnerable a ataques informáticos.

2.7 Conclusiones

El coste de desarrollo del proyecto es elevado, tanto económicamente como en dedicación requerida. Pero a cambio se obtendrá una herramienta útil, se demostrará que desarrollar en entorno Microsoft no es tan caro y se conseguirá finalizar la carrera. Por estas razones se decide proceder con el proyecto.

CAPÍTULO 3: FUNDAMENTOS TEÓRICOS

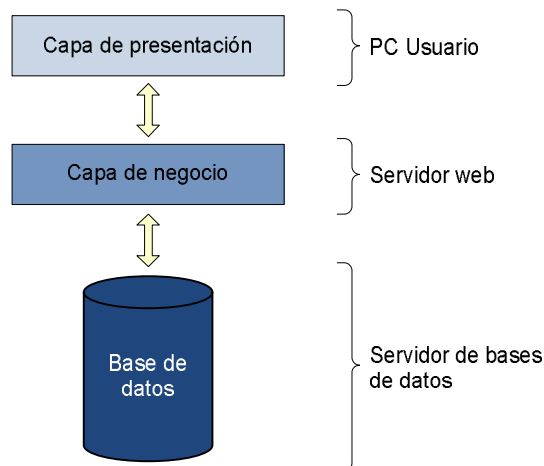
3.1 Introducción

Hoy en día es una práctica muy extendida en el campo del desarrollo de software el estructurar las aplicaciones en capas. Anteriormente a este planteamiento todo el código de la aplicación se hallaba al mismo nivel, con las desventajas que ello conllevaba.

En este capítulo se describe cómo se va a estructurar la aplicación, las alternativas que ofrece el mercado para implementar la estructura seleccionada y el proceso de toma de decisión en base el cuál se optará por una u otra opción.

3.2 Arquitectura de la aplicación

La herramienta que se va a desarrollar consistirá en una aplicación Web estructurada en tres capas: presentación, negocio y base de datos.



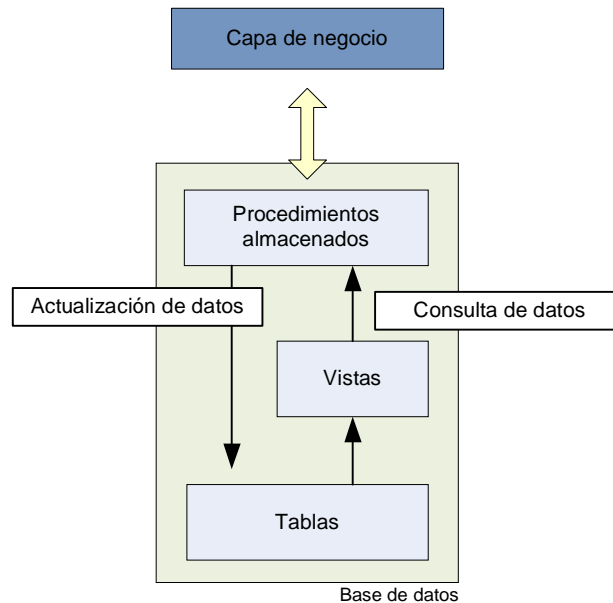
En este tipo de aplicaciones, cada capa se encarga de funciones específicas e interactúan entre ellas mediante interfaces muy concretas. Esta distribución presenta muchas ventajas, entre las que destacan las siguientes:

- Independencia entre los diferentes bloques del proyecto: se agiliza el trabajo en paralelo por parte de varios equipos sobre el mismo proyecto.
- Flujos de trabajo más claros y definidos: cada parte del programa tiene una función muy concreta y la forma de comunicarse con los bloques anterior y siguiente siempre es igual.
- Mantenimiento correctivo y evolutivo más sencillo: permite la depuración y actualización de las diferentes capas por separado, lo que evita tener que manejar todo el código a la vez.

La capa de presentación será la interficie que utilizará el usuario para interactuar con el sistema. Básicamente se tratará de páginas Web, hojas de estilos y *scripts* de cliente. Esta capa será procesada por el explorador web utilizado por el usuario.

La capa de negocio se ejecutará en el servidor Web y se encargará de gestionar toda la lógica de la aplicación. Cuando el usuario haga peticiones desde la capa de presentación, éstas serán procesadas por los objetos de negocio correspondientes, si es necesario se consultará o interactuará con la base de datos y finalmente se devolverá la respuesta a presentación en forma de página Web. Una característica que se pretenderá maximizar durante el desarrollo de esta capa será su independencia para con el sistema gestor de bases de datos utilizado. Es por ello por lo que se evitará la utilización de sentencias SQL dentro de la capa de negocio: todo el código SQL estará en la base de datos.

El sistema gestor de bases de datos se encargará de manejar la capa de datos. Internamente esta capa presentará la siguiente estructura:



El lenguaje SQL permanecerá en los procedimientos almacenados y éstos serán llamados desde la aplicación para ejecutar las diferentes acciones. Esto implica que si se desea migrar a otro sistema gestor de bases de datos o hacer la aplicación compatible con más de un sistema gestor de bases de datos, en el nuevo sistema únicamente tendrán que replicarse los objetos de la capa de datos, ni la capa de negocio y ni mucho menos la de presentación van a necesitar modificación alguna.

Por regla general, para cada entidad se dispondrá de un objeto de negocio, una tabla, una vista y los cuatro procedimientos almacenados correspondientes (uno para consulta y los otros tres para insertar, actualizar y eliminar registros).

3.3 Alternativas y tecnología utilizada

A continuación se presenta el proceso llevado a cabo para decidir qué tecnología utilizar, tanto a nivel de servidor Web como de sistema gestor de bases de datos.

3.3.1 Servidor Web

Para realizar sitios web sencillos no es necesario ningún servidor web: basta con crear un conjunto de páginas web, estructuradas en carpetas si se desea, y acceder a la página principal mediante el navegador. Pero si se quiere utilizar una base de datos y dotar al producto de más potencia y funcionalidad se tiene que recurrir a un servidor web. En este caso, el cliente interactúa con la página web, ésta se manda hasta el servidor web, quien procesa las acciones del usuario, consulta la base de datos si es necesario y retorna al cliente la respuesta en forma de página HTML. Para realizar estas

tareas, las páginas web, además de HTML, deben contener código de servidor. Este código se programará en diferentes lenguajes según el servidor web que lo deba procesar. En este sentido el mercado ofrece dos grandes alternativas: servidor web Internet Information Services (IIS a partir de ahora) y Apache. Decidir entre uno u otro determinará por completo la naturaleza de la aplicación a desarrollar (precio de desarrollo, sistema operativo, lenguaje a utilizar...). La siguiente tabla resume las principales diferencias entre ambos productos.

	IIS	Apache Server
Propietario	Microsoft	Software libre y de código abierto
Sistema operativo	Windows (en Windows 95 y 98 existía Personal Web Server, a partir de dichas versiones se introdujo IIS)	Unix (aunque se puede instalar en Linux, Windows, Macintosh y otros)
Lenguaje de servidor	ASP (ASP.NET actualmente)	PHP

Se podría simplificar mucho la comparativa concluyendo que IIS es la versión de pago y Apache es la versión gratuita.

En la tabla anterior se habla de lenguaje de servidor pero habría que puntualizar que ASP.NET no se trata de un lenguaje en si, sino de un entorno orientado al desarrollo de aplicaciones y servicios web. Se pueden implementar aplicaciones con arquitectura ASP.NET en diferentes lenguajes.

Para el desarrollo del proyecto se va a utilizar la opción de Microsoft por dos motivos fundamentales:

- 1) El equipo de desarrollo del proyecto tiene bastante experiencia en dicho entorno, si se decidiera trabajar con Apache lo que se ahorraría en licencias seguramente se gastaría en horas de investigación y formación ya que la experiencia que se tiene en Apache y PHP es prácticamente nula.
- 2) Mediante este proyecto se pretende crear una herramienta que, además de funcionar, sea diferente a la mayoría de productos CRM. Como se ha comentado en varias ocasiones y se comprobó en el análisis del estado del arte, la opción más común en este tipo de software es el entorno Apache/PHP/MySQL.

Para utilizar IIS como servidor web se va a necesitar un entorno Windows, el framework .NET y un editor de ASP.NET. Lo habitual en este tipo de proyectos es utilizar Windows XP Professional (oficialmente ésta es la versión mínima para disponer de IIS) y Visual Studio (versión 2005 o 2008).

En este caso, como se pretende abaratar costes y la envergadura del proyecto no es relativamente excesiva, se utilizará Windows XP Home y el editor gratuito Visual Web Developer 2005 Express Edition (VWDxe a partir de ahora). Las principales ventajas de VWDxe son dos: es gratuito (requiere activación gratuita) e incorpora para el desarrollo su propio servidor web que actúa a modo de IIS. Por este último motivo es por lo que no requerimos la versión Professional de XP. La principal desventaja de VWDxe es que no permite compilar la aplicación. A continuación se adjunta una breve explicación para ayudar a comprender mejor lo que este hándicap implica.

Las páginas Web en ASP.NET normalmente constan de tres archivos cuyas extensiones son: aspx, cs y resx. El archivo resx es un XML que contiene información acerca de los recursos utilizados en la página (por ejemplo imágenes). El archivo aspx contiene el código HTML y los *scripts* de cliente. Y el archivo cs contiene el código de servidor (esta extensión puede variar puesto que hace referencia al lenguaje de programación utilizado; así, cs se corresponde con el lenguaje C# (o C *sharp*), vb con Visual Basic, etc.). Es posible incrustar código de servidor en la página aspx pero no es habitual ni recomendable.

Durante el desarrollo de un proyecto se mantienen estos tres archivos por página pero cuando se desea ubicar la aplicación web en un servidor de producción se suele llevar a cabo lo que se denomina la publicación del sitio web. Esta publicación, entre otros, compila el código de los archivos cs y en su lugar genera una serie de archivos dll. Esto permite dos cosas: incrementar la velocidad de la aplicación web y preservar el código fuente para que no pueda ser revisado ni modificado por terceros.

VWDxe, como entorno de desarrollo, no incorpora la funcionalidad de publicación de sitio web. Este problema implica tener que compilar el sitio desde el intérprete de comandos utilizando las herramientas que incorpora .NET framework. Tal hecho no se considera especialmente grave puesto que el proceso de compilación sólo será necesario durante el cierre de cada versión. Adjunto a la memoria del proyecto se facilita un anexo explicando los pasos necesarios para compilar una aplicación Web .NET utilizando el ejecutable aspnet_compiler.

Así pues, para el desarrollo de la parte de aplicación del proyecto se utilizará VWDxe en su versión 2005 (por considerar que la 2008 aún no es lo suficientemente estable a tiempo de la toma de esta decisión).

3.3.2 Sistema gestor de bases de datos

Se contemplan cinco alternativas: PostgreSQL, MySQL, Oracle, MS Access y MS SQLServer.

PostgreSQL es un sistema gestor de bases de datos de código abierto. Se descarta su utilización por la completa falta de experiencia en su manejo.

MS Access no ofrece las prestaciones necesarias para implementar la estructura de base de datos descrita en el apartado anterior por lo que también se rechazará su utilización.

Se elegirá entre MySQL, Oracle y MS SQLServer en base a los siguientes criterios:

Criterio 1: Precio

La principal baza de MySQL es su precio: se distribuye bajo licencia GNU GPL (*) por lo que puede ser utilizado sin pagar.

En este sentido Oracle también ofrece una versión gratuita: Oracle 10g Express Edition (u Oracle XE). Esta versión se puede utilizar libremente para desarrollo y distribución, siempre que se respeten las curiosas cláusulas impuestas por Oracle, entre ellas no ser ciudadano de ningún país integrante del Eje del mal ni utilizar Oracle para desarrollar aplicaciones relacionadas con armas nucleares, químicas o biológicas. En el siguiente enlace se puede acceder al contrato de licencia completo:

http://www.oracle.com/technology/software/htdocs/xe_lic_prod.html

La alternativa gratuita de Microsoft es MS SQLServer Express. Existen dos versiones, la 2005 y la 2008, ambas totalmente gratuitas tanto para desarrollo como para distribución.

Así pues y, en contra de la creencia popular, en cuanto a costes de desarrollo el sistema gestor de bases de datos no va a ser relevante puesto que no vamos a tener que pagar por él. De todos modos es importante no perder de vista el objetivo final: generar un sitio Web. Este sitio Web se va a tener que alojar en algún servidor de Internet y en este aspecto sí que puede haber diferencias según la elección que se haya hecho. Existen multitud de servicios de hosting gratuitos para webs con MySQL. Para SQLServer Express se puede encontrar alguno (por ejemplo <http://www.aspspider.com/>) pero para Oracle es prácticamente imposible. De todos modos este va a ser un coste que asuma el usuario final del CMS y hospedar la web en un sitio gratuito no siempre es la mejor opción (propaganda abundante, velocidad lenta e irregular, garantía de servicio limitada...).

Criterio 2: Prestaciones

Tanto Oracle como SQLServer ofrecen la posibilidad de utilizar transacciones, procedimientos almacenados, triggers y funciones. Éstos son objetos necesarios que se utilizarán durante el desarrollo de la aplicación.

MySQL ha incorporado muchas de estas estructuras en versiones recientes, su funcionamiento aún no está optimizado. Este factor se tendrá en cuenta durante la toma de decisión final.

La velocidad no se prevé que sea determinante en este caso puesto que, contrariamente a lo que podría suceder si se trabajara por ejemplo con potentes ERP, el volumen de datos a manejar no va a ser muy grande. Los tres sistemas están perfectamente capacitados para dar soporte a un sitio web.

La versión Express de Oracle presenta las siguientes limitaciones: límite de uso de memoria (1Gb) y límite de uso de espacio (4Gb). Además, sólo se permite la instalación de una instancia por máquina y sólo utilizará un procesador de cuantos tenga el servidor.

SQL Server Express ofrece unas limitaciones similares: no sobrepasa 1Gb de RAM ni 4Gb de espacio en disco y sólo utiliza 1 procesador.

MySQL no presenta estos límites puesto que no se trata de una distribución Express.

La herramienta a desarrollar se va a alojar en un servidor contratado mediante hosting y en tales casos rara vez se permite al usuario sobrepasar estos límites por lo que tampoco se los considerará factores determinantes.

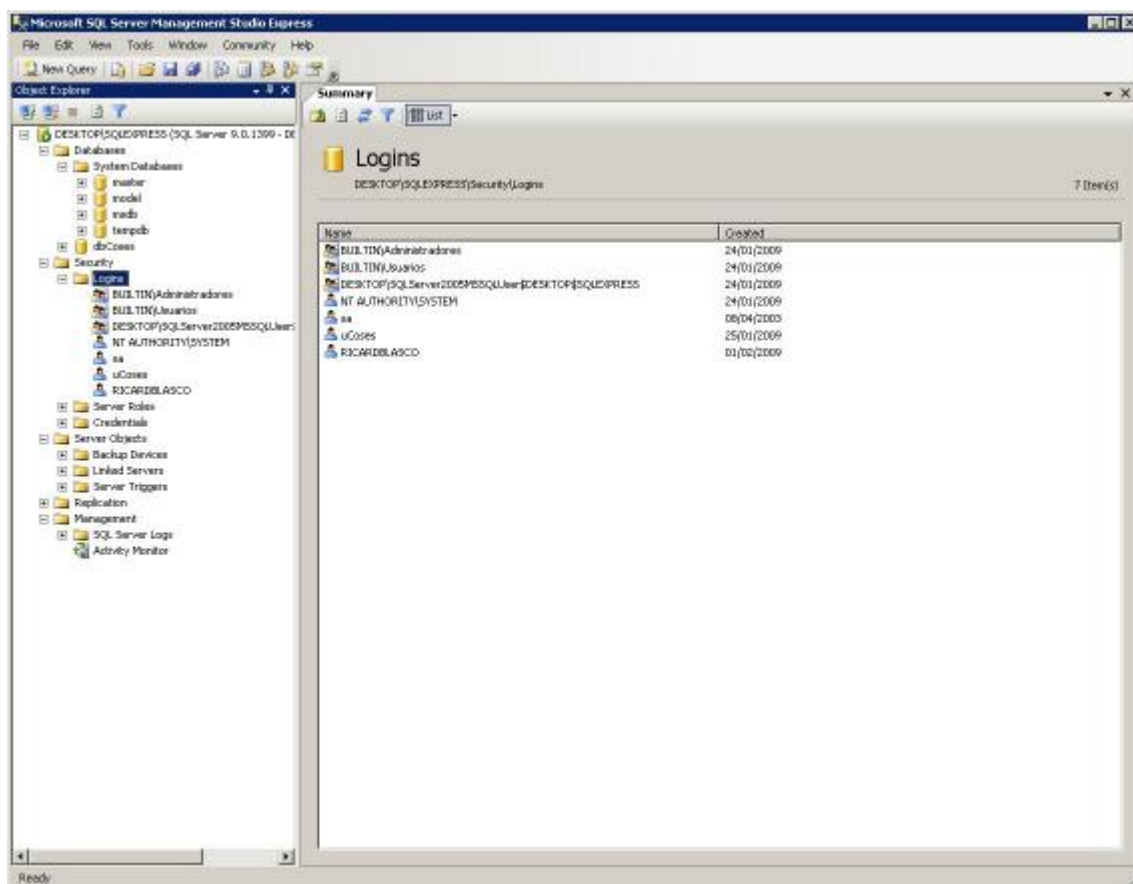
Critero 3: Experiencia y accesibilidad

El equipo de desarrollo del proyecto posee una amplia experiencia en administración y manejo tanto de MS SQLServer como de Oracle. Por el contrario, apenas se han realizado un par de proyectos con MySQL. Este será un factor de peso durante la toma de decisión final.

Para administrar la base de datos las tres alternativas ofrecen buenas soluciones.

SQL Server Express puede ser administrado mediante la herramienta MS SQL Server Management Studio. Esta herramienta es totalmente gratuita y se puede descargar de la página oficial:

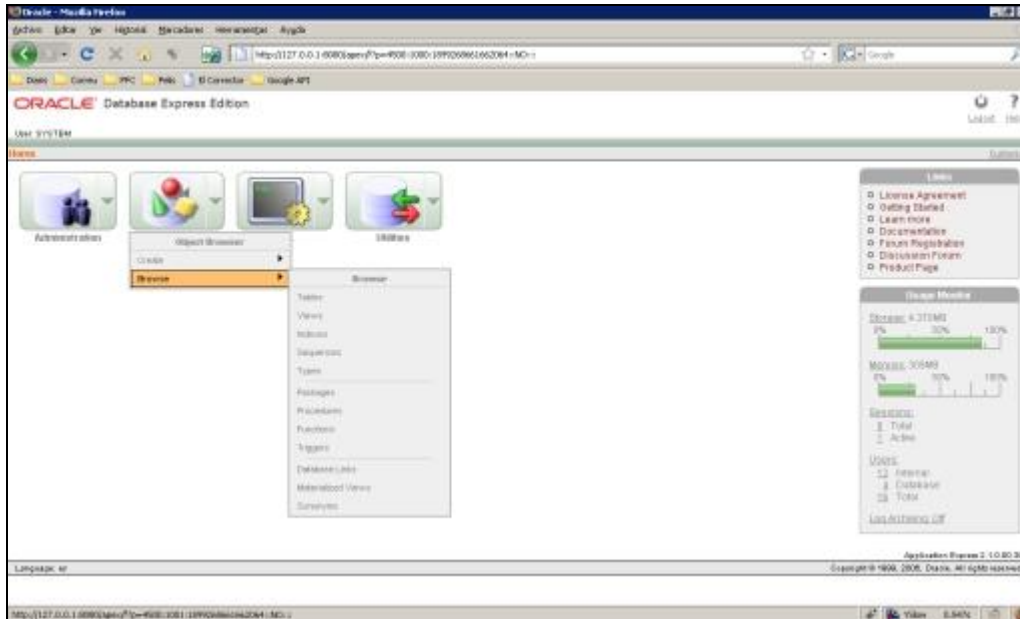
<http://www.microsoft.com/downloads/details.aspx?familyid=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=es>



MS SQL Server Management Studio Express

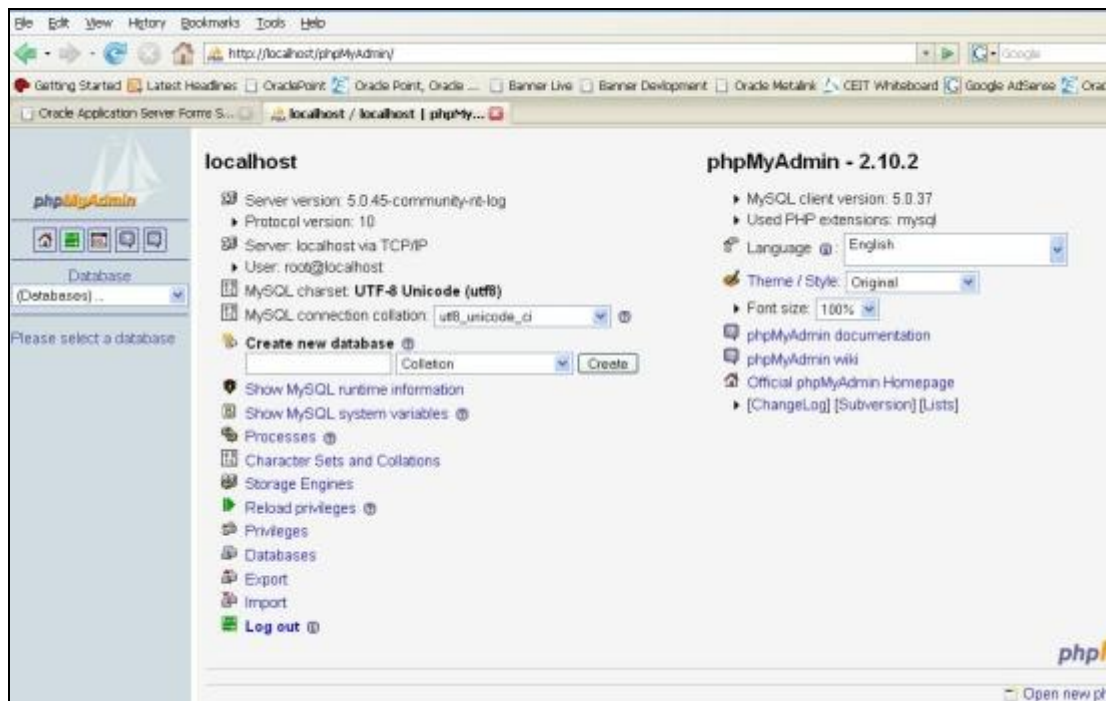
Desde siempre un gran hándicap de Oracle ha sido lo tedioso de su instalación y administración. Esto se solucionó en gran medida a partir de la versión 10 (incluida la Express). La instalación estándar se puede

realizar prácticamente utilizando el botón “Siguiente” tan característico de las instalaciones Windows. El mismo Oracle XE incorpora un panel de administración Web muy práctico y fácil de utilizar.



Oracle Express Edition: interfície de administració

Para la administración de MySQL existen numerosas alternativas gracias a que se trata de un producto de código abierto soportado por una gran comunidad de usuarios y desarrolladores. Una de las herramientas más utilizadas es phpMyAdmin. Se puede descargar de forma gratuita desde la Web oficial: http://www.phpmyadmin.net/home_page/index.php



phpMyAdmin interfície de administració

Elección

Resumiendo lo visto se pueden extraer las siguientes conclusiones:

- El precio de la base de datos a utilizar no es un factor relevante puesto que las tres opciones que se barajan disponen de versiones totalmente gratuitas y de libre distribución.
- El coste del alojamiento Web una vez finalizado el producto será más cuantioso en Oracle y SQL Server pero tampoco se considera un factor crítico puesto que se interpreta que si se desea un servicio de hosting de cierta calidad habrá que pagar de todos modos.
- Oracle y MS SQL Server presentan una estructura más robusta en lo que se refiere a opciones avanzadas de base de datos y los límites impuestos por sus versiones gratuitas no afectan sensiblemente cuando se trata de dar soporte a un sitio web como el que se pretende desarrollar.
- En lo referente a experiencia de desarrollo se tiene mucha más en Oracle y MS SQLServer que en MySQL.
- La administración de los tres sistemas no presenta diferencias significativas en cuanto a dificultad.

Se descartará MySQL por tres motivos:

- 1) Por falta de experiencia en relación con los otros dos sistemas.
- 2) Por estar menos orientado a aplicaciones web y más a sitios web convencionales.
- 3) Para hacer algo diferente a la gran mayoría de productos similares existentes hoy en día en el mercado.




Entre MS SQL Server y Oracle se va a apostar por la primera, dejando la puerta abierta a una futura compatibilidad con la segunda. Esta decisión se toma fundamentalmente por dos motivos:

- 1) Encontrar servicios de hosting para Oracle es bastante más caro y difícil.
- 2) Como se expuso en el apartado anterior, la aplicación se desarrollará utilizando el framework .NET de Microsoft por lo que ésta necesitará un entorno Windows. Este hecho no implica ni mucho menos la imposibilidad de utilizar Oracle pero para maximizar la compatibilidad de todo el sistema la mejor opción es MS SQL Server.




Así pues, el sistema gestor de bases de datos que se va a utilizar va a ser MS SQL Server Express.

Como puntualización es necesario añadir que, pese a la existencia de una versión 2008, se va a utilizar la 2005 para maximizar la compatibilidad con la versión del entorno de desarrollo que se va a utilizar (VWDxe 2005).

Mucha de la información utilizada durante este proceso de toma de decisión ha sido sacada de las respectivas páginas Web de los productos:

	MySQL
	http://www.mysql.com/
	MS SQL Server Express
	http://www.microsoft.com/spanish/msdn/articulos/archivo/261205/voices/sseoverview.msp
	Oracle 10g Express Edition
	http://www.oracle.com/lang/es/database/express_edition.html

Otras fuentes de información utilizadas:

	http://www.arsys.info/programacion/bases-de-datos/que-sistema-de-base-de-datos-usar
	Comparativa entre los diferentes gestores de bases de datos llevada a cabo por la empresa arsys.
	http://swik.net/MySQL/MySQL+vs+MS+SQL+Server
	Comparativa entre MySQL y MS SQL Server
	http://www.microsoft.com/technet/prodtechnol/sql/2005/msde2sqlexpress.msp
	Actualizar desde MSDE hacia MS SQL Server Express. MSDE (Microsoft SQL Server Desktop Engine) fue el precursor de la primera versión Express de SQL Server.

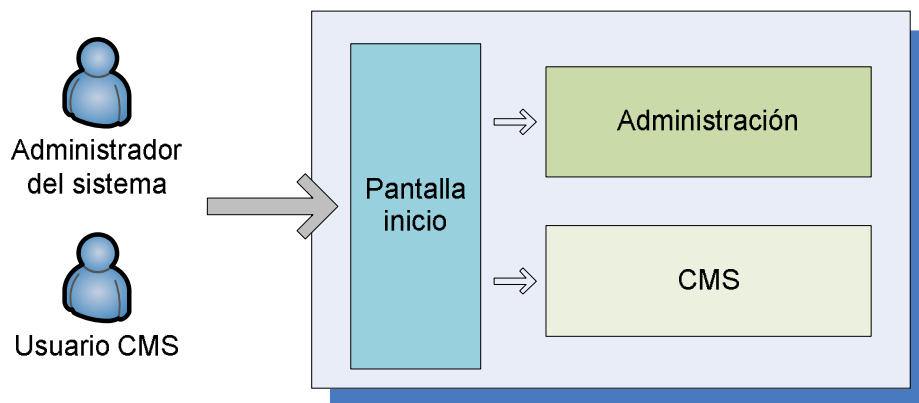
CAPÍTULO 4: ANÁLISIS

4.1 Introducción

En esta sección se detallarán los diagramas de flujo para los principales casos de uso de la aplicación, así como se explicarán los diferentes ámbitos que compondrán la aplicación y los tipos de usuario que los utilizarán.

4.2 Tipos de usuario y ámbitos de la aplicación

La aplicación se dividirá en dos ámbitos bien diferenciados: interficie de administración del sistema e interficie de uso del mismo. Esta diferenciación da lugar a dos perfiles de acceso: administrador del sistema y usuario del CMS.



La parte de administración tendrá como objeto la gestión de los usuarios CMS por parte de los administradores así como la consulta de estadísticas acerca del uso del sistema. La interficie CMS será el gestor de contenidos propiamente dicho, mediante el cual los usuarios CMS podrán configurar y mantener su sitio web.

La puerta de entrada a la aplicación o pantalla inicial permitirá identificarse como usuario y acceder a una u otra interficie. También permitirá registrarse como nuevo usuario y solicitar una nueva contraseña en caso de pérdida.

Mediante el presente análisis se detallan las funcionalidades que incorporarán cada una de estas interfaces, su comportamiento y sus posibilidades.

4.3 Especificaciones funcionales

4.3.1 Pantalla de entrada

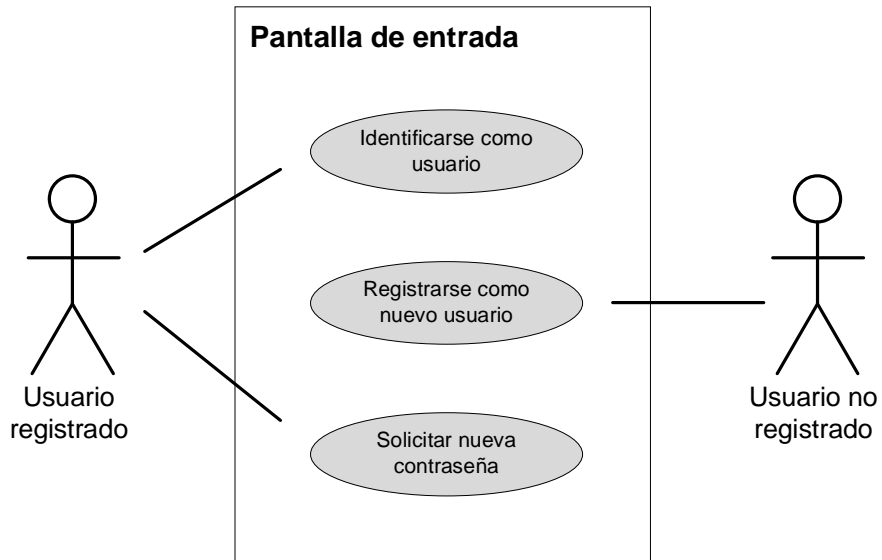


Diagrama de casos de uso de la pantalla de entrada

Caso de uso	Identificarse como usuario
Descripción	<p>Para poder acceder a la aplicación todo usuario deberá pasar inicialmente por este proceso de identificación. El usuario introducirá su nombre de usuario y su contraseña y pulsará el botón "Identificarse". El programa mostrará un error en los siguientes casos: usuario inexistente, contraseña errónea, usuario no activo o usuario bloqueado. En caso contrario se iniciará una nueva sesión y se redirigirá al usuario hacia la interficie correspondiente, según se trate de un usuario administrador o CMS.</p>
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B[Recuperar registro de la base de datos] B --> C{¿Existe?} C -- No --> D[Mostrar error usuario inexistente] C -- Sí --> E{¿Contraseña correcta?} E -- No --> F[Mostrar error contraseña incorrecta] E -- Sí --> G{¿Usuario bloqueado?} G -- Sí --> H[Mostrar error usuario bloqueado] G -- No --> I{¿Usuario activado?} I -- No --> J[Mostrar error usuario no activado] I -- Sí --> K[Inicio nueva sesión] K --> L{¿Usuario Administrador?} L -- Sí --> M[Redirección a interface de administración] L -- No --> N[Redirección a interface CMS] </pre>

Caso de uso	Registrarse como nuevo usuario
Descripción	Mediante esta opción se emitirán las solicitudes de creación de usuario. El visitante accederá a un formulario que le permitirá introducir su nombre, dirección de correo (con campo de confirmación), nombre de usuario, contraseña (con campo de confirmación) y nombre de su sitio web. Cuando pulse el botón de enviar solicitud se creará un registro en la tabla de usuarios cuyo estado será "pendiente de activación" y se le notificará que un administrador activará su usuario y le mandará un correo de conformación.
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Todos los campos obligatorios cumplimentados?} B -- No --> C[Mostrar error faltan campos] B -- Sí --> D{¿Nombre de usuario, email o nombre sitio web ya existentes?} D -- Sí --> E[Mostrar error datos ya existentes] D -- No --> F{¿Campos confirmación email y confirmación contraseña correctos?} F -- No --> G[Mostrar error confirmación incorrecta] F -- Sí --> H[Crear usuario pendiente de activación] H --> I[Mostrar mensaje operación realizada] </pre>

Caso de uso	Solicitar nueva contraseña
Descripción	<p>Cuando un usuario haya olvidado su contraseña pulsará el botón "Olvidé mi contraseña" de la pantalla de entrada. Será redirigido a una pantalla en la que tendrá que introducir el correo electrónico que utilizó para registrarse y pulsar el botón "enviar". Si no existe ningún usuario con el correo electrónico especificado o el usuario al que pertenece el correo no está activado, el programa notificará el error; de lo contrario, se generará una nueva contraseña aleatoria para el usuario, se actualizará el registro de la base de datos y se enviará por correo la nueva contraseña al usuario. Finalmente se mostrará un mensaje confirmando que la operación se ha realizado y que recibirá la nueva contraseña en su correo. La intervención de usuarios administradores no será necesaria durante este proceso.</p>
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Existe algún usuario con dicho email en la base de datos?} B -- No --> C[Mostrar error email desconocido] B -- Sí --> D{¿Está activado el usuario?} D -- No --> E[Mostrar error usuario no activado] D -- Sí --> F[Generar nueva contraseña] F --> G[Cambiar contraseña de usuario] G --> H[Enviar nueva contraseña por correo] H --> I[Mostrar mensaje operación realizada] </pre>

4.3.2 Administración

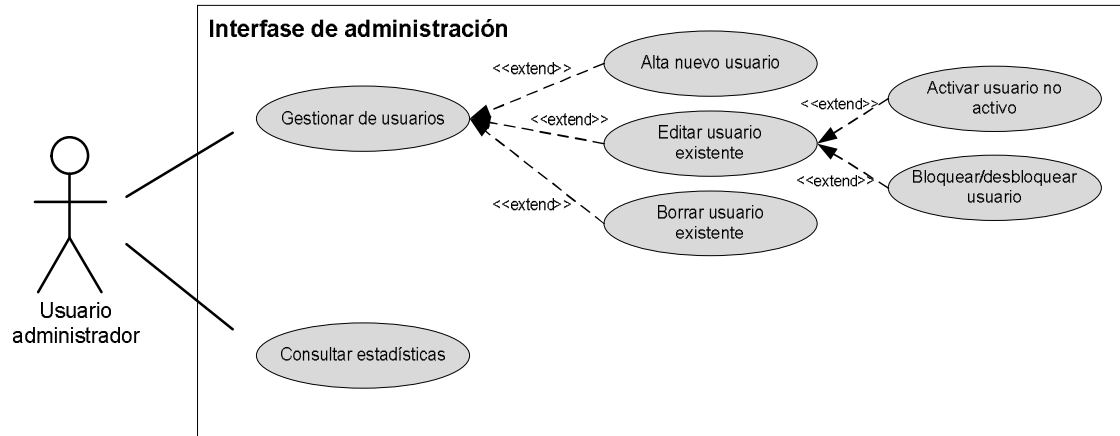


Diagrama de casos de uso de la interfase de administración

Caso de uso	Alta nuevo usuario
Descripción	<p>Mediante esta opción los usuarios administradores podrán crear nuevos usuarios con independencia del mecanismo de creación de usuarios mediante registro situado en la pantalla de entrada de la aplicación. Se podrán crear los dos tipos de usuario reconocidos por el sistema: administrador y usuario CMS. Sólo en el segundo caso, junto a la creación del usuario se creará también la estructura física y lógica correspondiente a su sitio web virtual.</p> <p>Para crear un usuario se pedirán los siguientes campos: nombre, dirección de correo, nombre de usuario, contraseña, tipo de usuario (administrador o CMS), nombre del sitio web y estado del usuario (sólo disponible valor "Usuario activo"). Todos los campos serán obligatorios.</p>
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Todos los campos obligatorios cumplimentados?} B -- No --> C[Mostrar error faltan campos] B -- Sí --> D{¿Nombre de usuario, email o nombre sitio web ya existentes?} D -- Sí --> E[Mostrar error datos ya existentes] D -- No --> F[Crear usuario] F --> G{¿El usuario es de tipo Administrador?} G -- Sí --> H[Mostrar mensaje operación realizada] G -- No --> I[Crear estructura sitio web virtual] I --> H </pre>

Caso de uso	Editar usuario existente
Descripción	Mediante esta acción el usuario administrador podrá cambiar los datos de usuarios existentes. El tipo de usuario no se podrá modificar y, en caso de usuario CMS, el nombre del sitio web virtual tampoco se podrá modificar. Los valores para el campo "Estado" serán: Activo y bloqueado (excepcionalmente existirá también el valor "pendiente de activación", presente cuando se esté editando un usuario creado desde la pantalla de inicio a través del proceso de registro de nuevo usuario).
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Todos los campos obligatorios cumplimentados?} B -- No --> C[Mostrar error faltan campos] B -- Sí --> D{¿Nombre de usuario o email ya existentes?} D -- Sí --> E[Mostrar error datos ya existentes] D -- No --> F[Modificar usuario] F --> G[Mostrar mensaje operación realizada] </pre>

Caso de uso	Borrar usuario existente
Descripción	Mediante esta opción se podrán eliminar usuarios del programa. En caso de tratarse de usuarios tipo CMS, durante el proceso de eliminación también se tendrá que eliminar la estructura correspondiente a su sitio web virtual. Un usuario no se podrá eliminar a si mismo.
Diagrama de flujo	<pre> graph TD A{¿El usuario es de tipo administrador?} -- Sí --> B[Borrar usuario] A -- No --> C[Borrar sitio web virtual] C --> B B --> D[Mostrar mensaje operación realizada] </pre>

Caso de uso	Consultar estadísticas
Descripción	<p>Mediante esta opción el usuario administrador podrá acceder a los siguientes datos (por sitio web virtual y en total): espacio de disco consumido, número y tipo de páginas creadas, y número y tipo de módulos utilizados. Esta información se presentará en modo tabla y en formato gráfico.</p> <p>La pantalla de consulta de estadísticas dispondrá de un desplegable en la parte superior en el que se listarán todos los sitios web virtuales existentes, más la opción "todos los sitios web". De este modo, el usuario seleccionará el sitio web para el que se mostrarán las estadísticas. Tras seleccionar un valor, el usuario pulsará un botón "consultar estadísticas" y se refrescará la información de pantalla aplicando el nuevo filtro. Cuando se acceda a la pantalla de consulta de estadísticas por primera vez, el desplegable tendrá seleccionado como valor por defecto "todos los sitios web", de modo que se mostrarán las estadísticas generales (sin filtro por sitio web).</p>
Diagrama de flujo	<pre> graph TD A[Leer valor del desplegable "sitio web"] --> B{¿Valor seleccionado es igual a "todos los sitios web"?} B -- Sí --> C[Recuperar estadísticas sin filtrar por sitio web] B -- No --> D[Recuperar estadísticas aplicando filtro sitio web] C --> E[Mostrar estadísticas] D --> E </pre>

4.3.3 CMS

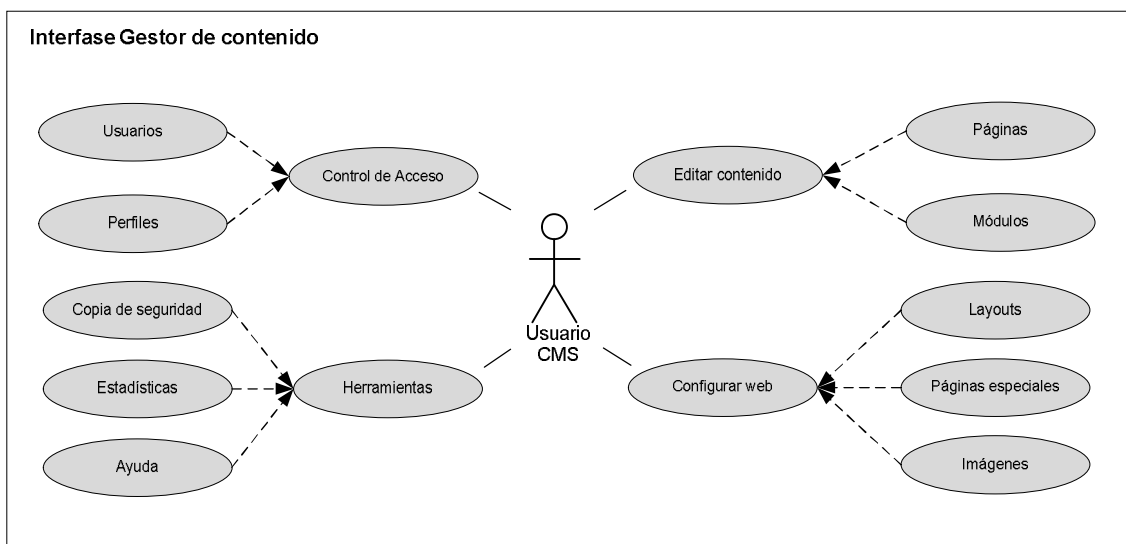
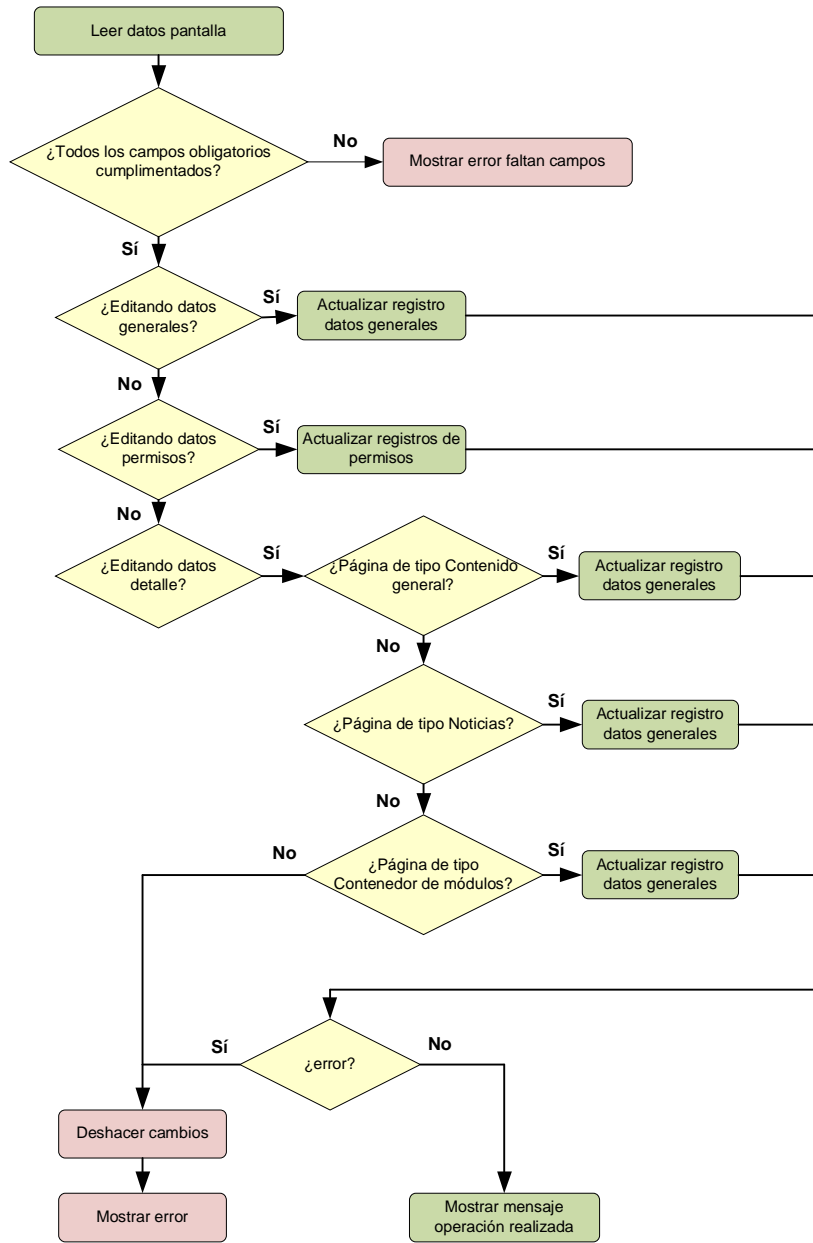


Diagrama de casos de uso de la interfase CMS

Caso de uso	Editar contenido: Páginas (Alta nueva página)
Descripción	<p>El usuario podrá añadir a su sitio web tantas páginas web como desee. Cuando se cree una página se deberá especificar obligatoriamente el nombre y el tipo de página. Los tipos de página disponibles serán los siguientes:</p> <ul style="list-style-type: none"> • Contenido general: página sin estructura previa. • Noticias: página con entradas tipo <i>blog</i>. • Contenedor de módulos: páginas para mostrar módulos. <p>El usuario también deberá especificar la ubicación de la página (carpeta) y opcionalmente una descripción.</p> <p>Durante el proceso de alta de la página todos los botones de la pantalla permanecerán deshabilitados salvo el botón de guardar y el de cancelar. El alta en la base de datos se realizará en dos pasos: alta del registro en la tabla de páginas (con los datos generales de la página) y alta del registro en la correspondiente tabla de detalle según el tipo de página. Estas operaciones se englobarán dentro de una misma transacción para que si hay error durante alguna de ellas se puedan deshacer todos los cambios.</p>
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Todos los campos obligatorios cumplimentados?} B -- No --> C[Mostrar error faltan campos] B -- Sí --> D[Crear página (datos generales)] D --> E{¿error?} E -- Sí --> F[Deshacer cambios] E -- No --> G[Crear página (datos detalle según tipo página)] F --> H[Mostrar error] G --> I{¿error?} I -- Sí --> F I -- No --> J[Volver a pantalla de selección de página] </pre>

Caso de uso	Editar contenido: Páginas (Edición página)
Descripción	<p>La edición de una página existente se realizará a través de tres pantallas: datos generales, datos detalle y permisos.</p> <p>La pantalla de datos generales permitirá editar los campos Denominación, Ubicación y Descripción. El tipo de página no se podrá cambiar una vez la página haya sido creada.</p> <p>La pantalla de detalle cambiará según el tipo de página:</p> <ul style="list-style-type: none"> • Páginas de contenido general: se mostrará un editor que ayude al usuario a insertar de forma fácil contenido en la página web. El editor permitirá insertar texto con formato, imágenes, tablas y otros elementos como objetos <i>flash</i>, y también incluirá funciones básicas de editor de textos: cortar, copiar, pegar, alinear, etc. En el caso de las imágenes, el usuario podrá especificar una <i>url</i> o escoger entre las imágenes que previamente haya subido a su propio sitio web. • Páginas de noticias: el usuario podrá especificar si las noticias de la página admiten o no comentarios de los usuarios. También podrá especificar el número de noticias por página, así como consultar las noticias incluidas en la página y los respectivos comentarios. Desde aquí no se podrán añadir y editar ni noticias ni comentarios puesto que para estas funciones existirán roles específicos que se explicarán más adelante en este mismo documento. Cuando los usuarios del sitio web accedan a una página de noticias, según su perfil y la configuración del sitio web, podrán añadir, editar y eliminar noticias y comentarios. Para una noticia se guardará la siguiente información: título, autor, fecha, contenido y hasta tres categorías o etiquetas (utilizadas posteriormente para filtrar noticias). Cuando se edite una noticia adicionalmente se podrá especificar el motivo de la edición. Para un comentario se guardará la siguiente información: autor, fecha y contenido. Como en las noticias, cuando se edite un comentario se podrá especificar el motivo de la edición. Las páginas de noticias presentarán sus noticias paginadas y la navegación se realizará mediante una barra situada en la parte inferior de la pantalla. • Páginas contenedoras de módulos: durante la edición de este tipo de pantallas, al usuario se le mostrará una rejilla sobre la que podrá arrastrar y soltar todos los módulos que previamente haya creado en su sitio web. Cuando los visitantes del sitio accedan a una página de este tipo, los módulos se mostrarán en las posiciones establecidas durante el diseño. Un mismo módulo solo podrá aparecer una vez por página. Un mismo módulo podrá aparecer en tantas páginas contenedoras de módulos como se desee. <p>Finalmente, la pantalla de permisos permitirá definir, para una página en concreto, los permisos de acceso según perfil. Esta pantalla mostrará todos los perfiles creados y mediante un desplegable permitirá especificar para cada perfil si tiene o no acceso a la página que se está editando. Por defecto todos los perfiles tendrán acceso a todas las páginas.</p>

Diagrama de flujo



Caso de uso	Editar contenido: Páginas (Eliminación página)
Descripción	Cuando el usuario CMS elimine una página de su sitio web se deberán eliminar automáticamente tanto los registros de detalle según el tipo de página como los registros de permisos. Será responsabilidad del usuario CMS el no eliminar páginas que estén referenciadas mediante hipervínculo desde otras páginas.
Diagrama de flujo	<pre> graph TD Start(()) --> Conf{¿Confirmar borrar?} Conf -- No --> Cancel[Operación cancelada] Conf -- Si --> BorrarPer[Borrar registros Permisos] BorrarPer --> Error1{¿error?} Error1 -- Si --> Deshacer[Deshacer cambios] Deshacer --> MostrarError[Mostrar error] Error1 -- No --> BorrarDetalle[Borrar registros Detalle] BorrarDetalle --> Error2{¿error?} Error2 -- Si --> Deshacer Deshacer --> MostrarError Error2 -- No --> BorrarGeneral[Borrar registro General] BorrarGeneral --> Error3{¿error?} Error3 -- Si --> Deshacer Deshacer --> MostrarError Error3 -- No --> Volver[Volver a pantalla de selección de página] </pre>

Caso de uso	Editar contenido: Módulos
Descripción	<p>Los módulos serán bloques funcionales que se podrán incrustar y mostrar en las páginas del sitio web que sean del tipo Contenedor de módulos. El usuario CMS podrá crear tantos módulos como desee. Los tipos de módulo disponibles inicialmente serán los siguientes:</p> <ul style="list-style-type: none"> • Imagen: para mostrar una imagen en una página contenedora de módulos. El usuario especificará el origen de la imagen (url o una imagen subida en el propio sitio web) y su tamaño (automático o ancho y alto específicos). Durante la configuración de este tipo de módulos el usuario podrá previsualizar el aspecto que tendrá la imagen en el sitio web. • Conexión de usuarios: este módulo permitirá a los usuarios visitantes del sitio web identificarse. Los usuarios CMS que deseen establecer permisos por perfil dentro de su sitio web necesitarán mostrar un módulo de conexión de usuarios para que sus usuarios web se puedan identificar. • Menú: este tipo de módulos servirán para poder mostrar menús en el sitio web. Un usuario se podrá crear tantos menús como desee y luego mostrarlos en su sitio web mediante las pantallas contenedoras de módulos. Para cada menú, el usuario podrá definir si será un menú horizontal o vertical, y añadir tantas entradas de menú como desee. Para cada entrada de menú, podrá definir el texto, la posición dentro del menú y el vínculo o página que se abrirá cuando se seleccione el menú (url o una página del propio sitio web). Sólo se permitirán menús con un nivel de profundidad. Para los módulos de tipo menú el usuario

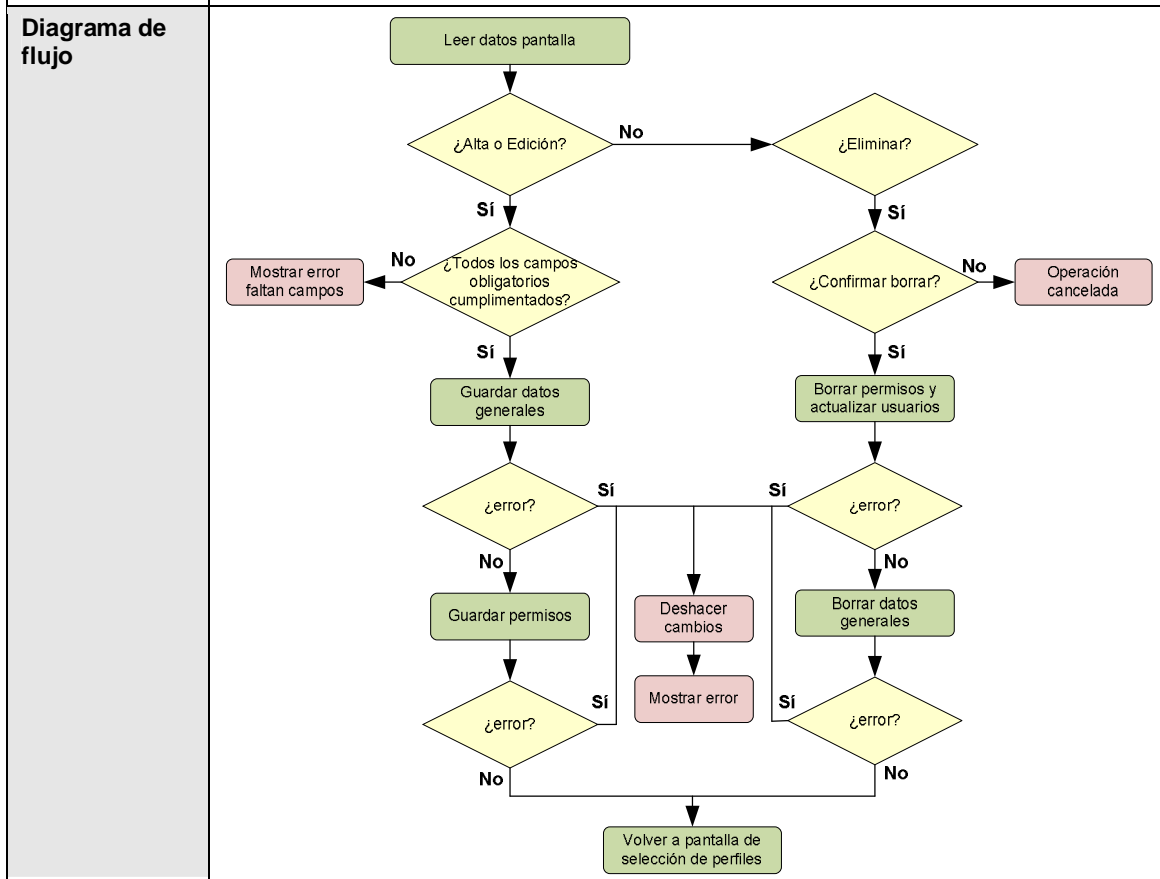
configuraciones de un mismo sitio web.

Además de estos datos generales, para cada perfil se podrán definir los siguientes aspectos:

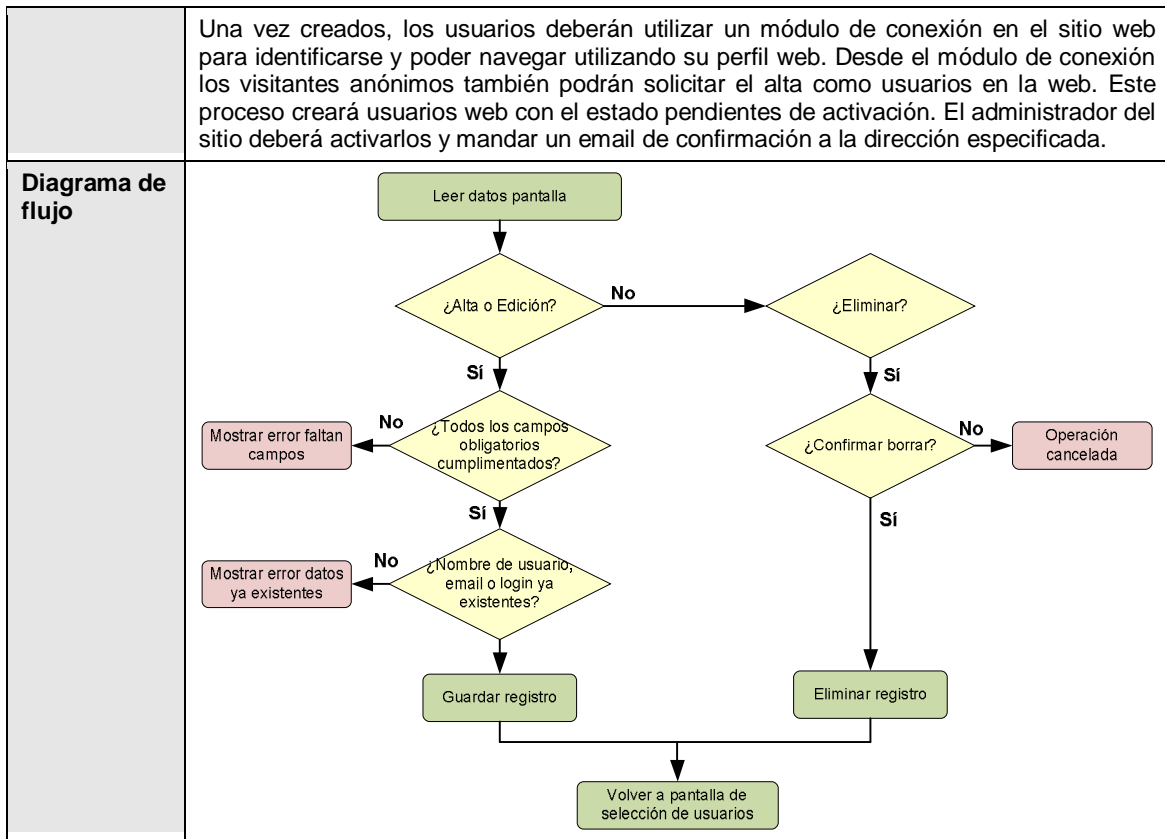
Permisos por tipo de página: cada tipo de página tendrá asociados una serie de permisos que se podrán definir para cada perfil. Esta característica será importante para las páginas de tipo Noticias ya que permitirá definir, para cierto perfil, si tiene permiso de alta y/o edición de noticias y alta y/o edición de comentarios. Los permisos de alta implicarán que los usuarios con dicho perfil pueden añadir noticias y comentarios a las páginas de tipo noticias. Los permisos de edición harán referencia al hecho de que un usuario pueda modificar y/o eliminar noticias y comentarios añadidos por otros usuarios. Un usuario siempre podrá editar y/o eliminar noticias y comentarios creados por él mismo.

Permisos por página: para cada una de las páginas del sitio web se podrá definir si los usuarios de cierto perfil tienen o no acceso a la misma. Cuando se añada una nueva página por defecto todos los perfiles tendrán acceso a ella. Cuando se añada un nuevo perfil, por defecto tendrá acceso a todas las páginas. Esta relación se podrá configurar tanto desde perfiles como desde páginas (como ya se comentó).

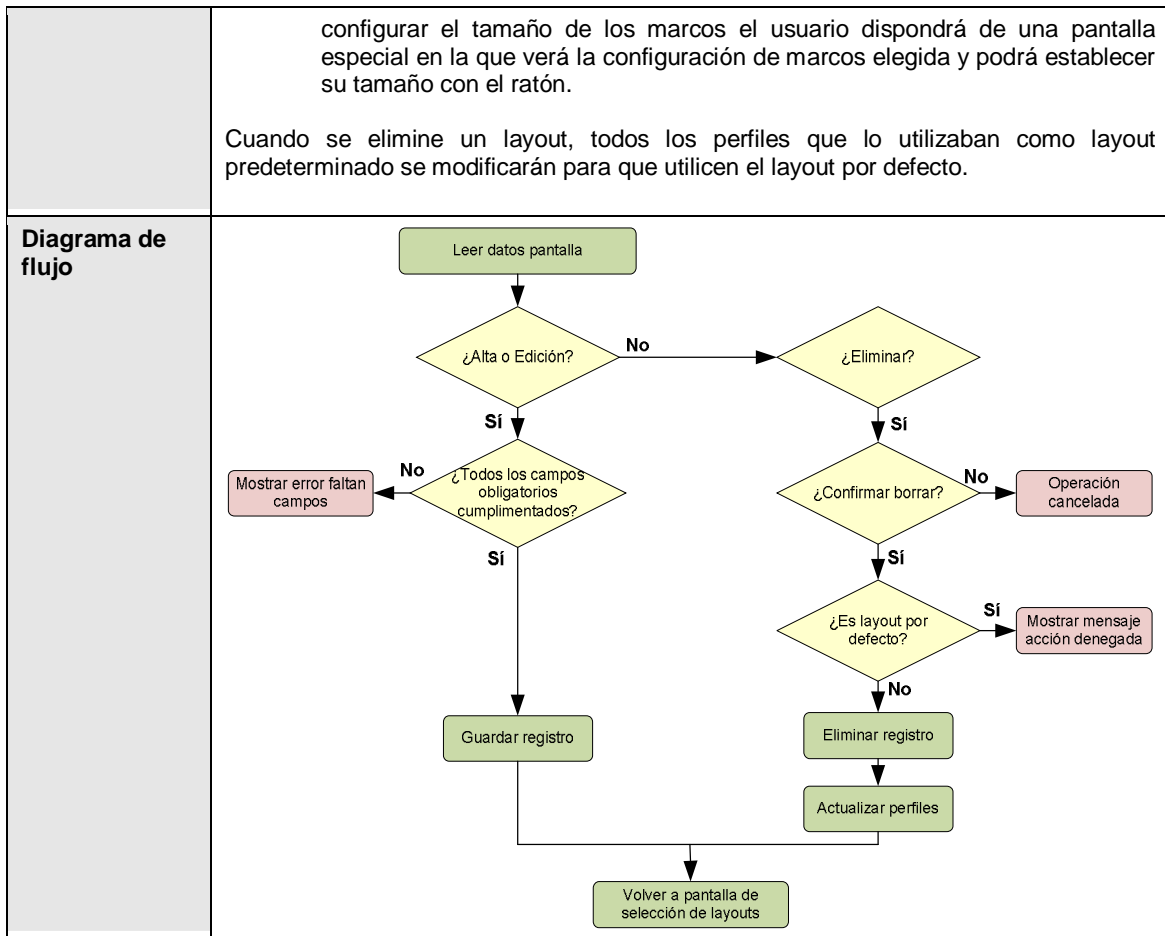
Cuando se elimine un perfil, todos los usuarios web que lo utilizaban pasaran a tener perfil Anónimo.



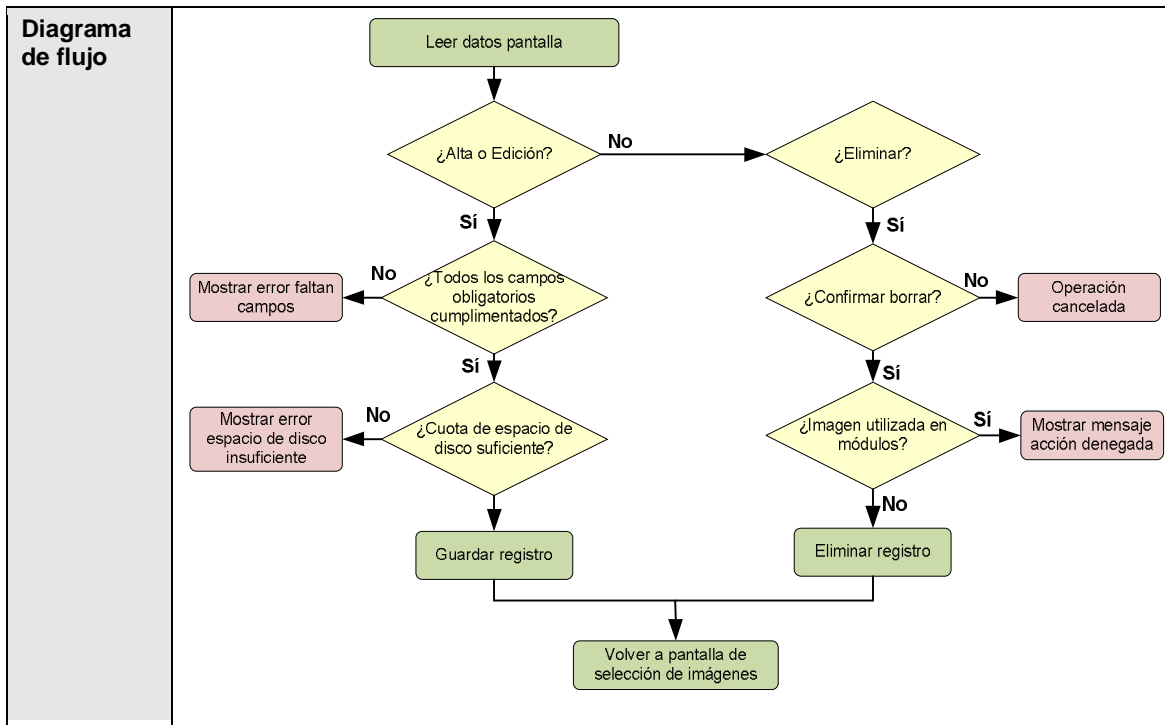
Caso de uso	Control de acceso: usuarios
Descripción	El sistema permitirá al usuario CMS crear tantos usuarios web para su sitio web como desee. Para cada usuario deberá introducir la siguiente información: nombre (obligatorio), email (obligatorio), login (obligatorio), contraseña (obligatorio), perfil web (obligatorio) y estado (obligatorio, valores posibles: activo, bloqueado y a la espera de activación).



<p>Caso de uso</p>	<p>Configurar web: layouts</p>
<p>Descripción</p>	<p>Un layout representa una distribución de elementos en la pantalla. El sistema permitirá crear, para un sitio web, tantos layouts como se desee, pudiendo relacionar estos layouts con los diferentes perfiles de acceso para permitir que la apariencia y el contenido de un mismo sitio web puedan ser totalmente distintos según el perfil de acceso.</p> <p>Siempre habrá un layout por defecto, que será el que se mostrará cuando el usuario que acceda al sitio web no esté registrado o cuando el perfil del usuario no tenga un layout especificado.</p> <p>Para un layout, el sistema permitirá especificar las siguientes características:</p> <ul style="list-style-type: none"> • Denominación y descripción: datos para identificar el layout, el primero de ellos obligatorio. • Layout por defecto: sólo podrá haber un layout por defecto (el que desee el usuario), mientras un layout esté marcado como por defecto, no podrá ser eliminado. • Con marcos o sin marcos: especifica si el layout constará de una única página principal (que a su vez podrá contener menús y todo tipo de contenidos) o se dividirá en marcos o <i>frames</i>. En el caso de no utilizar marcos el usuario deberá especificar la página principal de entre todas las que haya creado previamente. En el caso de utilizar marcos el usuario podrá especificar qué marcos desea mostrar (se dividirá la pantalla en 7 marcos, el marco central o principal siempre se mostrará, los demás serán opcionales). Para cada marco que se muestre el usuario deberá especificar qué página contendrá. Adicionalmente, si el layout se divide en marcos, el usuario podrá especificar si se tiene que mostrar borde entre los marcos o no. En caso afirmativo, podrá especificar el tamaño, color y comportamiento del borde (fijo o móvil). Para

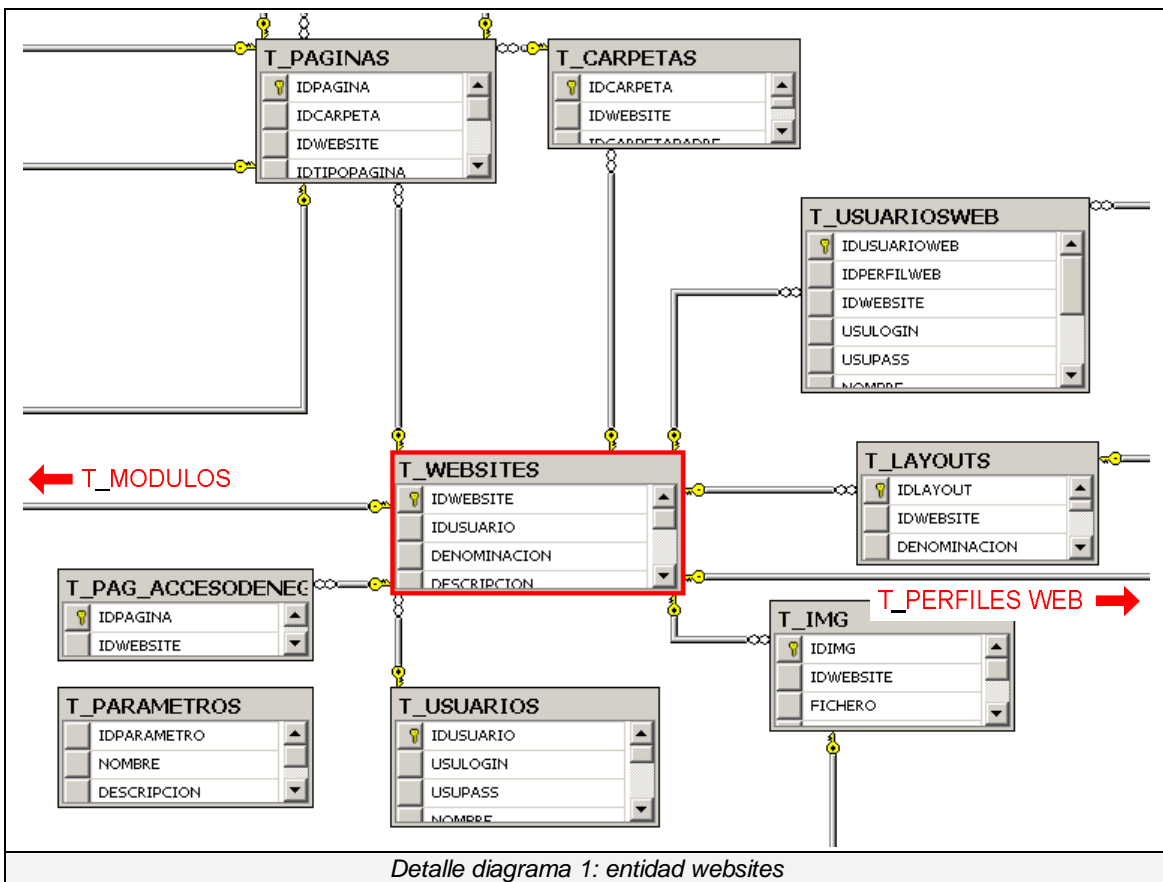
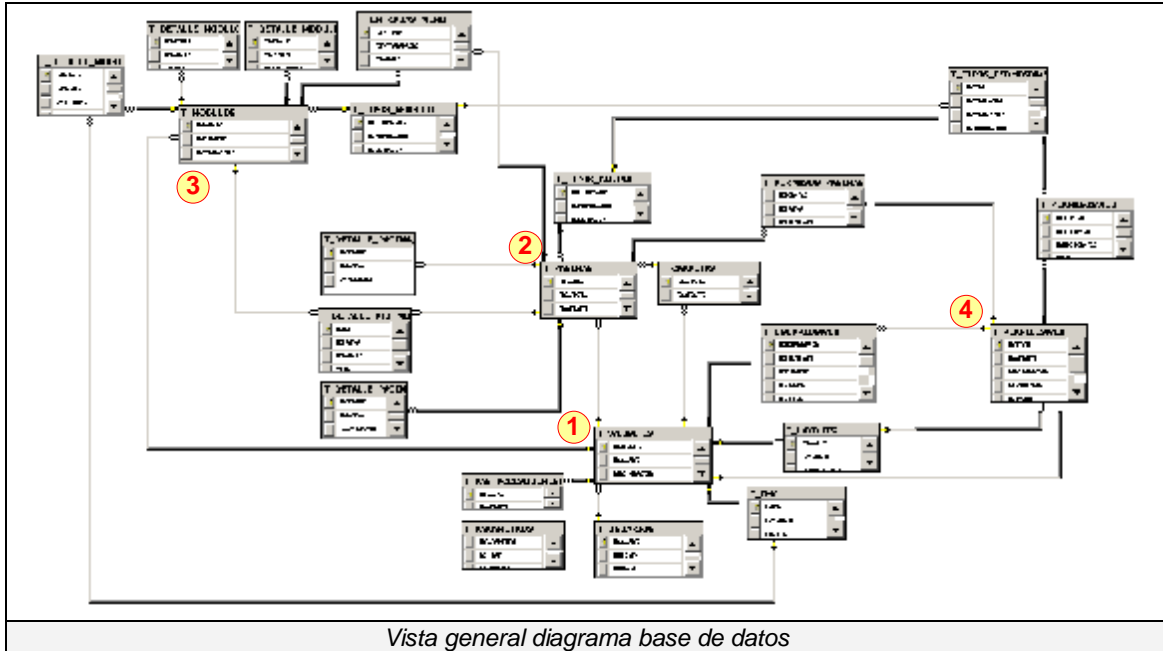


Caso de uso	Configurar web: imágenes
Descripción	<p>Cuando un usuario desee mostrar imágenes en su sitio web lo podrá hacer de dos formas distintas: especificando una url (dirección de internet en la que se encuentra la imagen) o subiendo la imagen al propio sitio web. El segundo método será más seguro puesto que entonces el sitio web del usuario no dependerá de imágenes alojadas en otros servidores. Por el contrario, el espacio por usuario estará limitado y no podrá sobrepasar este límite de espacio.</p> <p>Para gestionar sus imágenes el usuario CMS dispondrá de una interfice en la que se mostraran todas las imágenes subidas y podrá navegar por ellas con los cursores o el ratón. Para cada imagen se guardará la siguiente información: denominación (obligatorio), descripción, tamaño (solo lectura) y el propio archivo de imagen. Desde la pantalla de alta-edición, el usuario podrá cargar archivos de imagen que tenga en su ordenador y previsualizarlas. Si el sistema detecta que se ha excedido el espacio de disco asignado, no permitirá subir la imagen.</p> <p>El sistema no permitirá eliminar imágenes que estén siendo utilizadas por módulos de tipo imagen.</p>



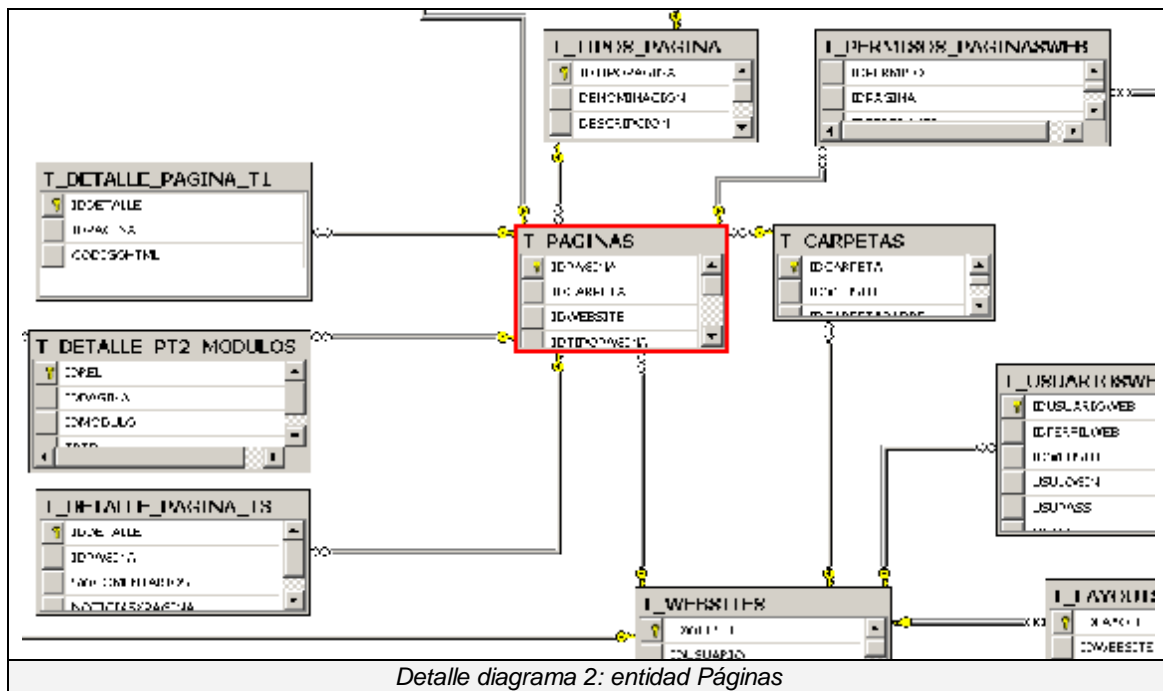
Caso de uso	Configurar web: páginas especiales
Descripción	En todos los sitios web virtuales alojados en la aplicación existirán una serie de páginas especiales que los usuarios CMS no podrán eliminar pero sí personalizar. En una primera versión las páginas especiales disponibles serán: página de acceso denegado y página de alta de nuevo usuario. El usuario podrá modificar el contenido de estas páginas con un editor similar al utilizado para gestionar el contenido de las páginas de carácter general.
Diagrama de flujo	<pre> graph TD A[Leer datos pantalla] --> B{¿Todos los campos obligatorios cumplimentados?} B -- No --> C[Mostrar error faltan campos] B -- Sí --> D[Guardar registro] D --> E[Volver a pantalla de selección de páginas especiales] </pre>

4.4 Diseño de la base de datos



La entidad *websites* (tabla T_WEBSITES) es una de las principales entidades de la aplicación, cada registro de esta tabla se corresponde con un web site virtual dentro de la aplicación. Esta entidad se relaciona 1 a 1 con la entidad *usuarios* (tabla T_USUARIOS) puesto que en la primera versión no está previsto que un mismo usuario pueda tener varios websites. Las tablas que guardan los registros de páginas especiales también tienen esta misma relación 1 a 1 con la tabla de websites ya que sólo habrá un registro de cada página especial por website. El resto de entidades relacionadas con la tabla T_WEBSITES lo hacen 1 a N: habrá varios registros en cada tabla relacionados con un único website. Estas entidades son las siguientes:

- **Usuarios (tabla T_USUARIOS):** se encarga de almacenar la información referente a los usuarios CMS (cada registro de esta tabla es un usuario CMS de la aplicación). El contenido de esta tabla es el que se gestiona desde la interficie de administración de la aplicación.
- **Imágenes (tabla T_IMG):** almacena un registro por cada imagen subida a los websites alojados en la aplicación.
- **Layouts (tabla T_LAYOUTS):** almacena un registro por cada Layout que se crea dentro de los diferentes websites alojados en la aplicación. Esta entidad a su vez está relacionada con los perfiles web que se hayan creado dentro del mismo website (esta tabla queda fuera del diagrama 1, su relación con la tabla T_WEBSITES también es 1 a N).
- **Usuarios web (tabla T_USUARIOSWEB):** para cada usuario que el usuario CMS cree dentro de su sitio web se insertará un registro en esta tabla. Esta tabla está relacionada N a 1 con la tabla de perfiles web que queda fuera del diagrama 1 (un perfil web podrá ser utilizado por varios usuarios web pero un usuario web sólo podrá tener 1 perfil web).
- **Carpetas (tabla T_CARPETAS):** esta entidad simplemente se utiliza para clasificar las páginas contenidas en los respectivos sitios web virtuales. Los usuarios CMS podrán crear carpetas y organizar sus páginas web mediante las mismas.
- **Páginas (tabla T_PAGINAS):** otra de las entidades principales de la aplicación. La tabla T_PAGINAS almacenará los datos generales de cada página web creada dentro de los sitios web virtuales. Toda página tendrá su correspondiente registro en la tabla T_PAGINAS. Además, según el tipo de página, tendrá registros relacionados en las diferentes tablas de detalle que se describirán más adelante.
- **Módulos (tabla T_MODULOS):** esta tabla queda fuera del diagrama 1 pero su función y relación con la entidad websites es similar a la de las páginas: relación N a 1 con la entidad websites y almacenamiento de los datos generales de todos los módulos creados dentro los websites. Esta parte de la base de datos se describirá más adelante.



La entidad *paginas* (tabla T_PAGINAS) almacena los datos generales de cada página web creada desde la interfaz CMS. Además de estos datos generales, según el tipo de página, se guardará información de la página en las diferentes tablas de detalle.

La tabla T_DETALLE_PAGINA_T1 guarda la información de detalle de las páginas de tipo contenido general.

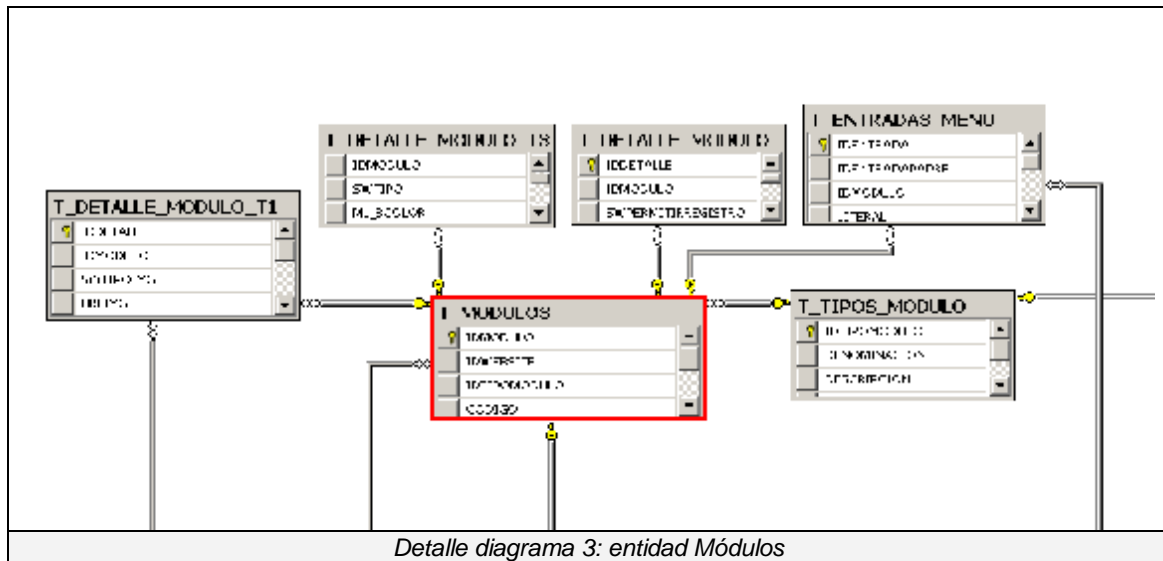
La tabla T_DETALLE_PT2_MODULOS se utiliza para las páginas de tipo contenedor de módulos: en realidad es una tabla N a N entre T_PAGINAS y T_MODULOS (esta última queda fuera del diagrama 2).

La tabla T_DETALLE_PAGINA_T3 guarda la información de detalle de las páginas de tipo noticias.

La tabla T_NOTICIAS no aparece en el diagrama 2, sus registros están relacionados con las páginas de tipo 3 en calidad N a 1 (para una misma página puede haber N Noticias).

La tabla T_COMENTARIOS no aparece en el diagrama 2, sus registros están relacionados con la tabla T_NOTICIAS en calidad N a 1 (varios comentarios por noticia, un comentario es exclusivo de una noticia).

La tabla T_PERMISOS_PAGINASWEB relaciona la tabla T_PAGINAS con la tabla T_PERFILESWEB y sirve para almacenar los permisos de acceso por página y perfil web.



La entidad *módulos* (tabla T_MODULOS) funciona de forma similar a la entidad *páginas*: la información general se almacena en la tabla T_MODULO y la información detalle en las tablas detalle según el tipo de módulo.

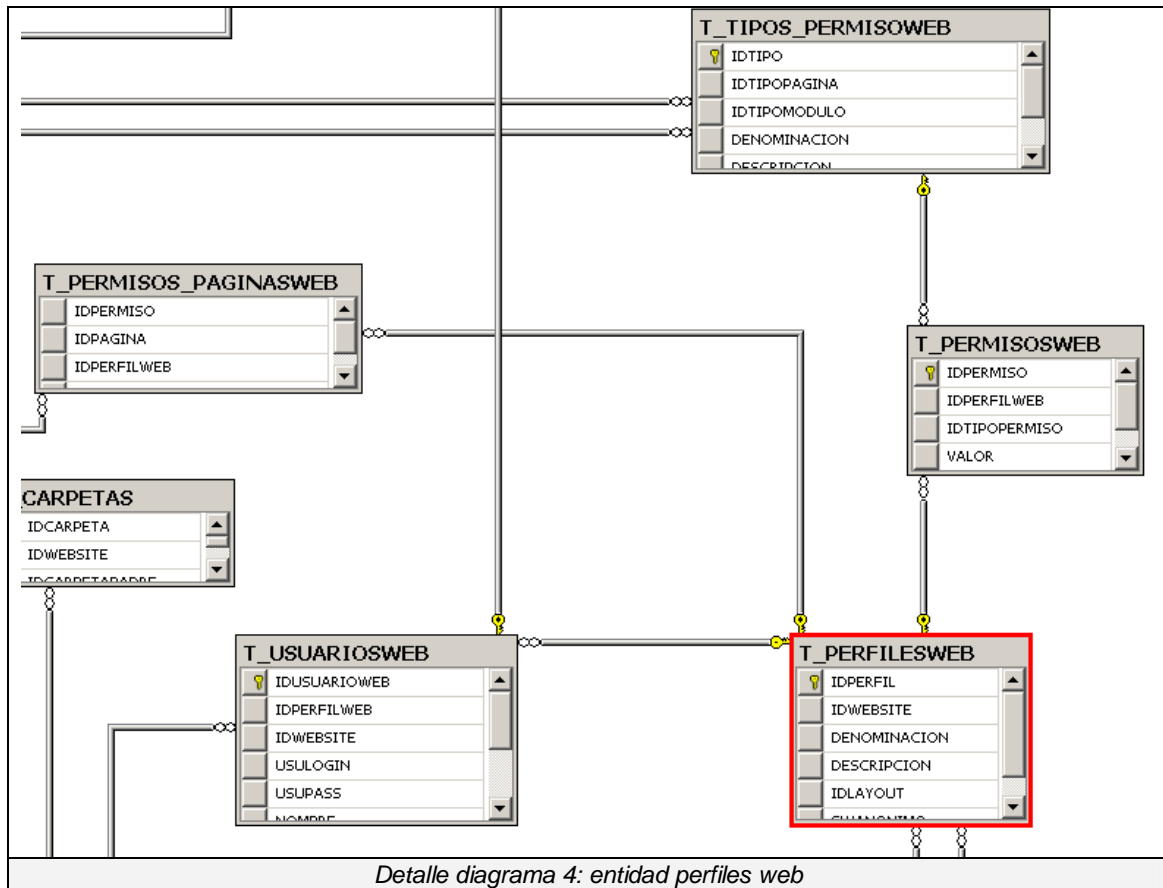
La tabla T_DETALLE_MODULO_T1 almacena la información detalle de los módulos de tipo imagen.

La tabla T_DETALLE_MODULO_T3 almacena la información detalle de los módulos de tipo menú (información referente a la apariencia del menú y a su comportamiento).

La tabla T_ENTRADAS_MENU implementa la relación N a N entre los módulos de tipo menú y las páginas web.

La tabla T_DETALLE_MODULO_T2 almacena la información detalle de los módulos de tipo Conexión de usuario.

En el diagrama 2 se puede ver la tabla T_TIPOS_PAGINA y en el diagrama 3 la tabla T_TIPOS_MODULO. Estas dos tablas sirven para relacionar cierto módulo y cierta página con la tabla T_TIPOS_PERMISOWEB. La tabla T_TIPOS_PERMISOWEB no es editable por los usuarios y almacena, para cada tipo de página y tipo de módulo, los permisos que se podrán conceder o denegar a los diferentes perfiles web. Ejemplo: las páginas de tipo Noticias tienen en la tabla T_TIPOS_PERMISOWEB registros correspondientes a las acciones: alta y edición de noticias, y alta y edición de comentarios. El usuario CMS no puede cambiar esto, pero sí puede definir la relación entre estos permisos y cada perfil web de su sitio web. Este sistema permite que cada nuevo tipo de página o tipo de módulo que se agregue en el futuro pueda presentar su propia interficie de permisos que el usuario CMS personalizará según perfil.



Dentro de cada sitio web virtual existirá un sistema de usuarios (usuarios web) y perfiles (perfiles web) en base al cual el usuario CMS definirá los permisos de acceso a las diferentes zonas de su web. Básicamente estos permisos se definen mediante dos tablas: T_PERMISOSWEB y T_PERMISOS_PAGINASWEB.

T_PERMISOSWEB almacena la relación N a N entre perfiles web y tipos de permisos web. Ejemplo: en esta tabla se almacena la información que determina si cierto perfil web puede o no añadir noticias en TODAS las páginas de tipo Noticias.

T_PERMISOS_PAGINASWEB almacena la relación N a N entre perfiles web y páginas web. Ejemplo: en esta tabla se almacena la información que determina si cierto perfil puede o no acceder a cierta página web.

La combinación de estos dos mecanismos de permisos aporta total flexibilidad a la hora de definir los accesos a su sitio web virtual por parte del usuario CMS. Por ejemplo, podrá incluir varias páginas de noticias, cada una moderada por un usuario web diferente de forma que el moderador de una página ni tan siquiera pueda acceder a la otra o pueda acceder pero no pueda añadir ni editar noticias ni comentarios de otros usuarios.

La tabla T_LAYOUTS no se muestra en el diagrama 4 pero, como ya se comentó, está relacionada 1 a N con la tabla de perfiles (un perfil tiene 1 Layout asociado, 1 Layout puede tener N perfiles asociados) de manera que cada usuario web acceda a una configuración del sitio web diferente según su perfil web.

CAPÍTULO 5: DISEÑO

En todas las pantallas del proyecto se aplicaran los mismos criterios de diseño con el fin de que el uso del programa resulte fácil e intuitivo.

Siempre que el usuario situe el ratón encima de un botón o un enlace, se cambiará puntero para mostrar una mano, indicando que puede hacer click sobre ese elemento para desencadenar un evento. Como fuente se utilizará Arial o Verdana, de tamaño 10px y color azul mate.

La pantalla se dividirá en tres regiones (o *frames*) separadas por un borde que cambiará de color según se trate de la interficie de administración o de la interficie CMS. Estas tres regiones serán las siguientes:

- **Frame superior:** mostrará el nombre de la aplicación, el usuario conectado y, en la interficie CMS, un enlace al sitio web virtual del usuario conectado. Junto al nombre del usuario se mostrará un botón para cerrar sesión.
- **Frame izquierdo:** mostrará el menú para navegar por las diferentes pantallas de la aplicación. Todas las opciones del menú constarán de un texto y un icono representativo para hacer la interficie más amigable. El texto de la opción seleccionada permanecerá en negrita.
- **Frame central o principal:** mostrará las pantallas de la aplicación propiamente dichas.

Habrán principalmente dos tipos de pantalla: las pantallas de selección y las pantallas de edición.

Pantalla de selección



The screenshot shows a web application interface. At the top left is the logo 'PFC'. At the top right, it displays 'Webster: FERRETERIA' and 'Usuario: FERRETERIA'. On the left side, there is a navigation menu with items like 'Inicio', 'Configuración', 'Contenido web', 'Control de Acceso', 'Usuarios', 'Perfiles', and 'Herramientas'. The main content area is titled 'USUARIOS' and contains a table with the following data:

Login	Nombre completo	Email	Estado
PEPE	Pepe Garcia	pepe.garcia@gmail.com	●
MARIA	Maria Rodriguez	maria.rodriguez@gmail.com	●
JUANITO	Juanito Fernández	juanito.fernandez@gmail.com	●

Pantalla de selección

Las pantallas de selección serán las que se abrirán desde el menú. Este tipo de pantallas tendrán como finalidad listar registros por pantalla. Su estructura será la siguiente: una cabecera en la parte superior, con el título o nombre de la entidad cuyos registros se están listando. Debajo de la cabecera una serie de botones correspondientes a las opciones principales que ofrece la pantalla. A continuación el listado de registros propiamente dicho. Los registros se mostrarán mediante una rejilla o *grid* con el siguiente comportamiento: cuando se presione sobre la cabecera de cierta columna, el listado se ordenará según el contenido de la misma; cuando se pase el ratón por encima de las filas, éstas se iluminarán para resaltar el registro seleccionado; cuando se seleccione un registro haciendo click en cualquier columna, se accederá a la pantalla de edición que permitirá modificar y/o eliminar el registro seleccionado.

Además de las pantallas de selección estándar, para algunas entidades se realizarán adaptaciones:

- **Pantalla de selección de páginas web:** debido a que las páginas se podrán clasificar en carpetas, no se utilizará el mecanismo antes citado mediante el cual haciendo click sobre un registro se accede a su pantalla de edición. En su lugar, en el listado de páginas web se habilitará una columna que permitirá, en caso de tratarse de una carpeta entrar a la misma, y en caso de tratarse de una página web mostrará qué tipo de página es. Adicionalmente para cada registro se mostrará una columna para acceder a la pantalla de edición y una para eliminar la página o carpeta. Además, debajo de la sección de botones pero encima del listado, se mostrará la ruta en la que se encuentra el usuario en relación a la navegación por las diferentes carpetas.
- **Pantalla de selección de imágenes:** las imágenes no se mostrarán como un listado sino en una tabla para que el usuario las pueda previsualizar. El usuario podrá navegar por las imágenes con los cursores o el ratón, la imagen seleccionada se mostrará en su tamaño original y junto a ella se mostrarán los botones para editarla y eliminarla.
- **Pantalla de selección de pantallas especiales:** al tratarse de un listado cerrado en el que el usuario no podrá añadir ni eliminar registros, no se utilizará un *grid*, simplemente se dispondrán unos iconos grandes correspondientes a cada pantalla especial para que el usuario las pueda identificar y seleccionar rápidamente.

Pantalla de edición

Mediante las pantallas de edición el usuario podrá dar de alta nuevos registros o editar y eliminar el registro previamente seleccionado desde la pantalla de selección.



Pantalla de edición

Las pantallas de edición constarán de una barra de título en la parte superior, seguida de una barra con botones correspondientes a las opciones principales y opcionalmente otra barra para la navegación tipo pestañas en las entidades que así lo requieran (cuando se precise se dividirá el contenido en pestañas para hacer la edición de registros más ágil y sencilla).

A continuación del título y las barras de botones se mostrará el formulario con los campos del registro que se esté editando. Los campos obligatorios se resaltarán con fondo amarillo. Se utilizará la zona inferior de la pantalla para mostrar mensajes de confirmación y/o error cuando se preciso. Para los mensajes de confirmación se utilizará letra verde, para los de error roja (ambos en negrita).

CAPITULO 6: IMPLEMENTACIÓN

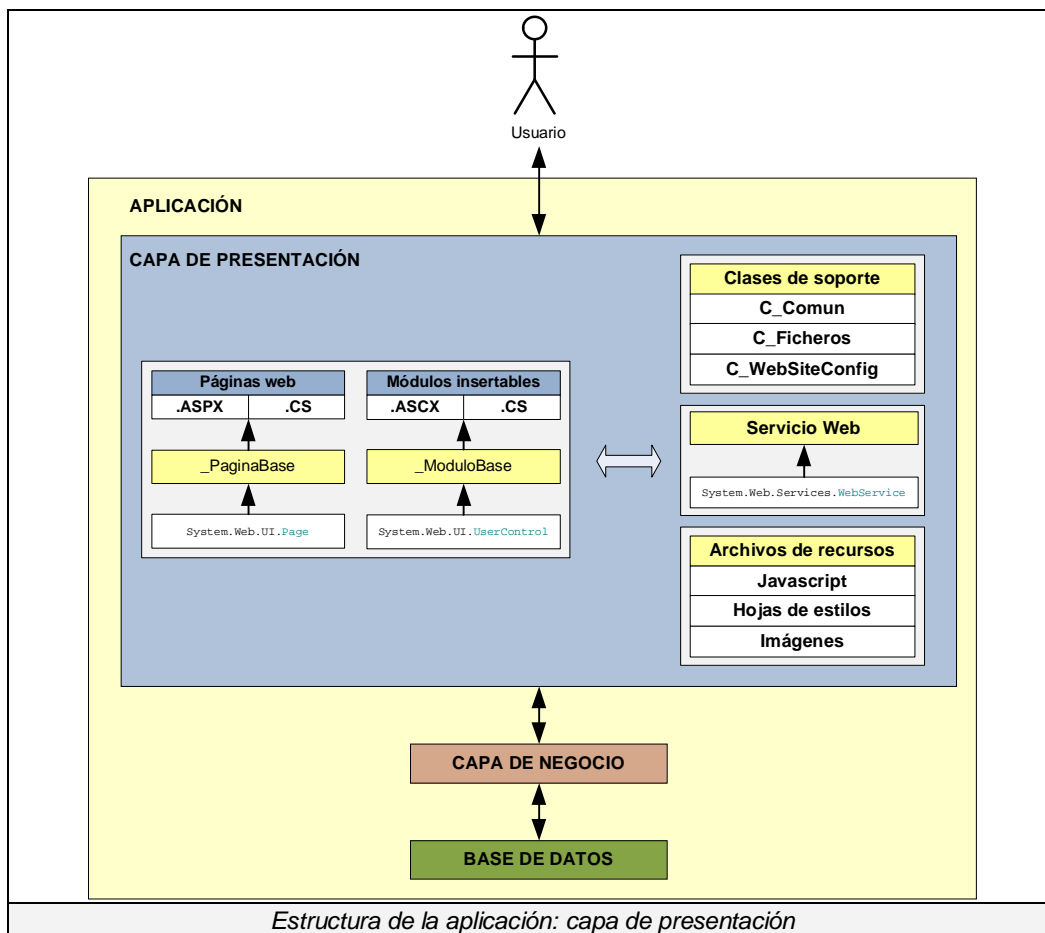
6.1 Descripción técnica

La aplicación se implementará mediante la tecnología ASP.NET de Microsoft. Concretamente se utilizará la versión 2.0 del *framework*. Como lenguaje de programación se utilizará C# y para la base de datos se ha elegido Microsoft SQL Server 2005 por su compatibilidad con el entorno de programación y por disponer de una edición gratuita. Al tratarse de una aplicación web, necesitará un servidor web para funcionar. Se utilizará el propio IIS que incorpora el sistema operativo Windows.

6.2 Estructura de la aplicación

Como ya se adelantó en el capítulo 3, la aplicación se dividirá en 3 capas: presentación, negocio y base de datos.

6.2.1 Capa de presentación



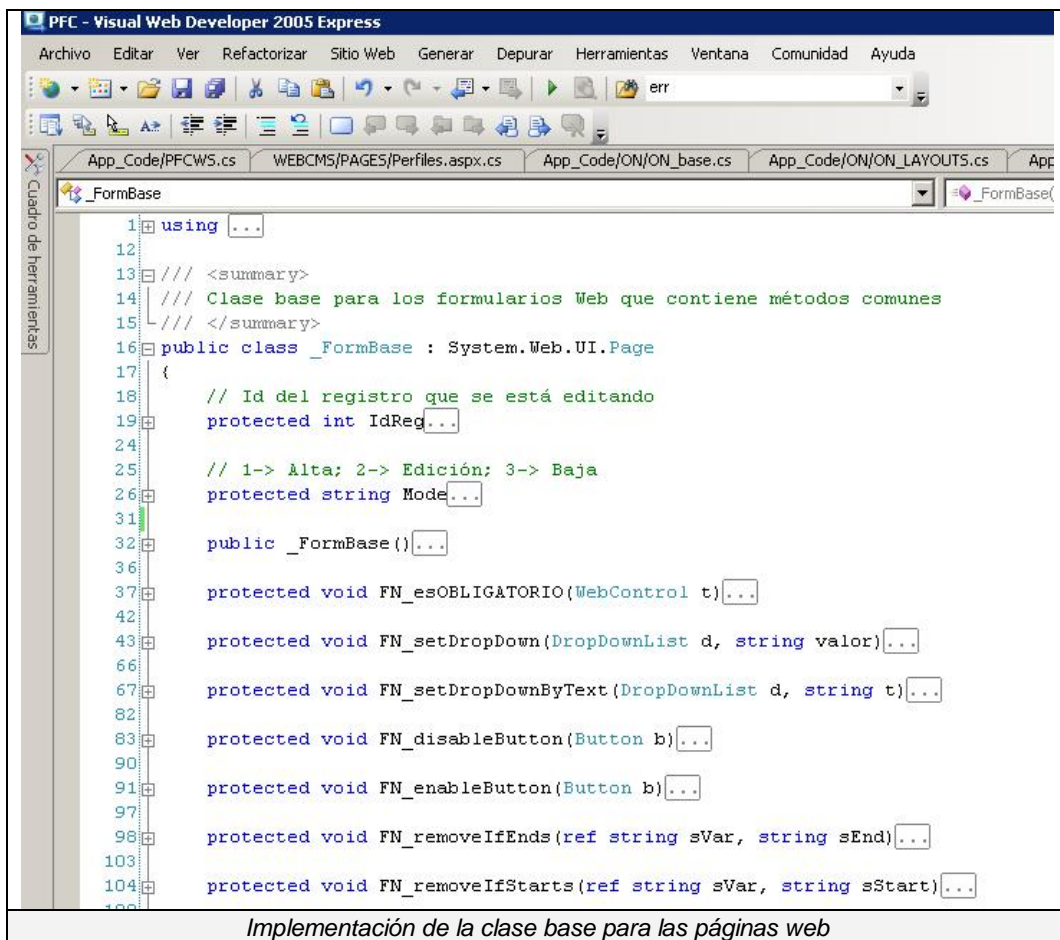
Funcionalidad

La capa de presentación será la parte de la aplicación con la que interactuarán los usuarios. Básicamente estará formada por páginas web y se comunicará con la base de datos a través de la capa de negocio.

Descripción

Las páginas web: Las páginas se separarán en carpetas según formen parte de la interficie de administración o de la interficie CMS. Todas las páginas web tendrán una estructura similar para facilitar la labor de programación. Se creará una clase base y todas las páginas web del proyecto heredaran de esta clase base. En la página base se definirán métodos comunes que se utilizarán posteriormente desde el resto de páginas. Este planteamiento ahorrará tiempo y trabajo tanto a la hora de programar como a la hora de realizar el mantenimiento de la aplicación.

El código de cada página se almacenará en dos archivos separados: en los archivos con extensión .cs se guardará el código de servidor (C#) mientras que en los archivos .aspx se guardará el código HTML. Este mecanismo permitirá conseguir un código mucho más claro y entendible.



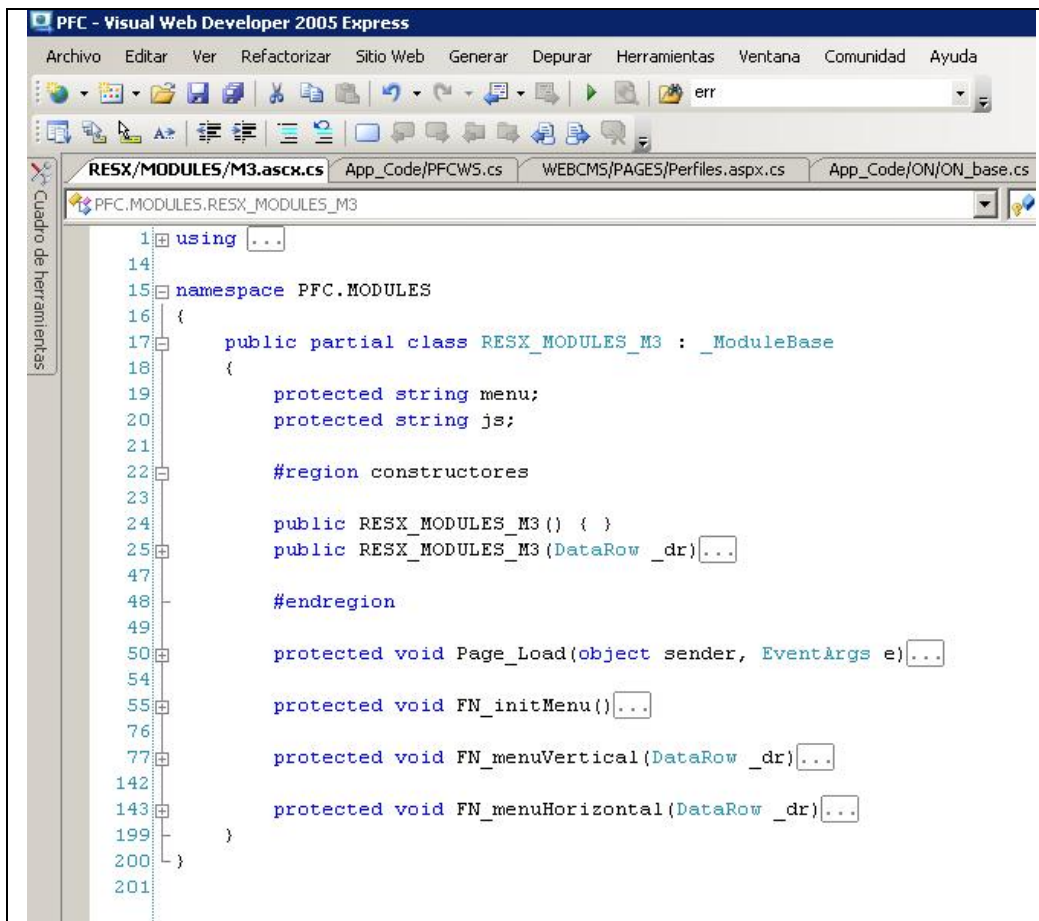
```

1  using ...
12
13  /// <summary>
14  /// Clase base para los formularios Web que contiene métodos comunes
15  /// </summary>
16  public class _FormBase : System.Web.UI.Page
17  {
18      // Id del registro que se está editando
19      protected int IdReg...
24
25      // 1-> Alta; 2-> Edición; 3-> Baja
26      protected string Mode...
31
32      public _FormBase()...
36
37      protected void FN_esOBLIGATORIO(WebControl t)...
42
43      protected void FN_setDropDown(DropDownList d, string valor)...
66
67      protected void FN_setDropDownByText(DropDownList d, string t)...
82
83      protected void FN_disableButton(Button b)...
90
91      protected void FN_enableButton(Button b)...
97
98      protected void FN_removeIfEnds(ref string sVar, string sEnd)...
103
104      protected void FN_removeIfStarts(ref string sVar, string sStart)...
109
  
```

Implementación de la clase base para las páginas web

Los módulos insertables: para cada tipo de módulo que el usuario CMS pueda incorporar en su web se creará un control de usuario de ASP.NET. Estos controles son bloques de código reutilizables en las diferentes páginas web. En cuanto a la programación los controles de usuario son similares a las páginas web. Así, también crearemos una clase base en la que definiremos métodos comunes y de la que todos los controles de usuario heredarán, y también guardaremos en archivos separados su código de servidor (extensión .cs) y su código HTML (en este caso extensión .ascx). Por ejemplo, cuando el usuario CMS decida insertar un menú en una página de su sitio web, lo que se hará será crear en la base de datos el registro correspondiente a ese menú y, cuando un visitante del sitio web acceda a la página, se instanciará el control de usuario encargado de mostrar módulos de tipo menú y se inicializará indicándole que registro de la base de datos tiene que mostrar. Organizar los módulos insertables mediante controles

de usuario permite un mantenimiento mucho más cómodo de la aplicación ya que modificando un control de usuario se modificarán automáticamente en todos los sitios web todos los tipos de módulo correspondientes. Este enfoque también repercutirá en un importante aumento de la escalabilidad: para añadir nuevas funcionalidades a la aplicación bastará con añadir nuevos controles de usuario o nuevas funciones a los controles de usuario ya existentes.



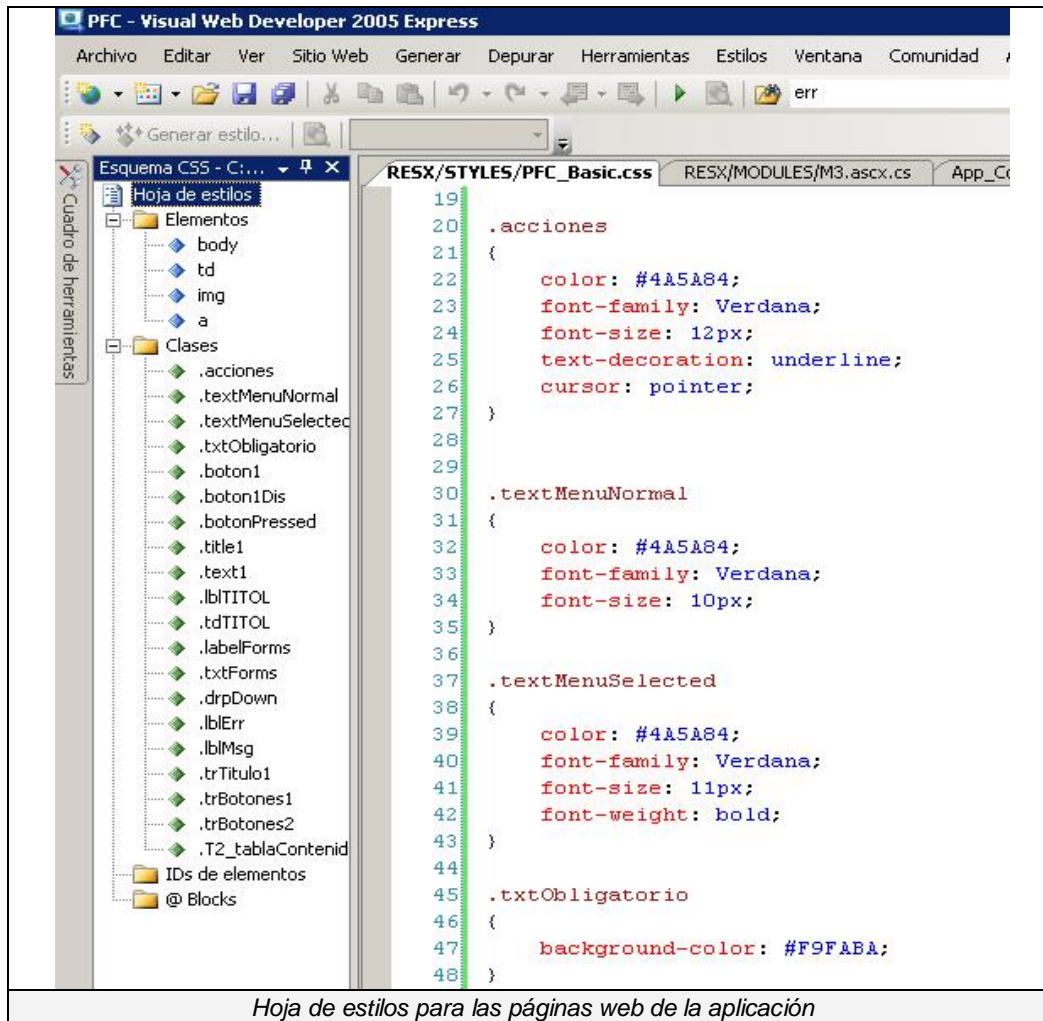
```

1  using ...
14
15 namespace PFC.MODULES
16 {
17     public partial class RESX_MODULES_M3 : _ModuleBase
18     {
19         protected string menu;
20         protected string js;
21
22         #region constructores
23
24         public RESX_MODULES_M3() { }
25         public RESX_MODULES_M3(DataRow _dr)...
26
27         #endregion
28
29         protected void Page_Load(object sender, EventArgs e)...
30
31         protected void FN_initMenu()...
32
33         protected void FN_menuVertical(DataRow _dr)...
34
35         protected void FN_menuHorizontal(DataRow _dr)...
36
37     }
38 }

```

Implementación del control de usuario de tipo 3, correspondiente a los Menús insertables

Archivos de recursos: para definir el aspecto visual de los formularios web se utilizarán hojas de estilo. Esto permitirá obtener un diseño mucho más homogéneo y más fácil de configurar y mantener. Concretamente, se creará una hoja de estilos para las páginas web y otra para cada tipo de módulo. Las funciones javascript permiten ejecutar código en el navegador del cliente evitando así tener que mandar consultas al servidor que lo que hacen es que la página web vaya y vuelva (lo que se conoce como hacer un *postback*). En aplicaciones web hay que evitar en la medida de lo posible los *postback* de las páginas ya que producen en el usuario molestos efectos de refresco de la pantalla, consumen tráfico de red y durante los mismos se puede perder información si no está debidamente almacenada. Pese a ello, el language script presenta algunas características que hacen desaconsejable abusar de él. La principal es que no es seguro y el usuario puede tener deshabilitada la ejecución de scripts en su navegador. Por lo tanto, en este proyecto se utilizará javascript para realizar tareas sencillas como validaciones de formato de los datos, validaciones de campos obligatorios cumplimentados, formateo de fechas y números u ordenación de listados. Todas las funciones javascript se definirán en un solo archivo al que se vincularán todas las páginas web de la aplicación. Otros archivos de recursos, como puedan ser las imágenes y los iconos, se almacenarán bajo una misma carpeta dentro del directorio de recursos de la aplicación.



Servicio web: como se ha mencionado en el punto anterior, no es conveniente hacer *postbacks* de las páginas web salvo cuando el usuario pulsa guardar o borrar o realiza alguna otra opción importante. De hecho, los *postbacks* se suelen realizar cuando el usuario lleva a cabo acciones que implican tener que acceder a la base de datos. Pese a ello, hay momentos en los se necesita acceder a la base de datos pero no es conveniente enviar la página al servidor (por ejemplo, cuando se muestran dos desplegables enlazados con la base de datos y el contenido del segundo se filtra según el registro seleccionado en el primero). Para solucionar estas situaciones se incluirá un objeto en la aplicación que heredará de la clase *WebService* (o Servicio Web). En los objetos de estas clases se pueden definir métodos web. Estos métodos tienen varias particularidades, son métodos que “expone” la aplicación y que pueden ser llamados (normalmente se utiliza la terminología “consumidos”) desde otras aplicaciones o desde el propio código javascript. De este modo, mediante el servicio web de la aplicación, podremos ejecutar código de servidor y de acceso a la base de datos desde las funciones javascript mediante llamadas a los métodos web que contenga, sin tener que realizar *postbacks* de las páginas. Estas llamadas a métodos web desde código javascript se conocen como llamadas asíncronas ya que después de la llamada se ejecuta el código del método web en el servidor mientras que la página web se sigue ejecutando en el cliente, independientemente uno del otro y con los riesgos que ello conlleva.

```

7 using System.Data;
8 using System.Web.Script.Services;
9
10 using PFC.ON;
11
12 /// <summary>
13 /// Metodos utilizados de forma asincrona desde Javascript
14 /// </summary>
15 [WebService(Namespace = "http://PFCWS/")]
16 [WebServiceBinding(ConformsTo = UriProfiles.BasicProfile1_1)]
17 [ScriptService]
18 public class PFCWS : System.Web.Services.WebService
19 {
20
21     public PFCWS()
22     {
23     }
24
25     [WebMethod(EnableSession = true)]
26     [ScriptMethod]
27     public string WH_guardarLayoutTemp(string par, string par2)
28     {
29         // Creamos una T_LAYOUT.DataRow temporal en session para la página LayoutsConfig.aspx
30         // Este método se llama de forma asincrona desde el botón btnConfig de la página LayoutsEdit.aspx
31
32         string[] parametros = par.Split(':');
33         string[] parametros2 = par2.Split('|');
34
35         ON_LAYOUTS onL = new ON_LAYOUTS();

```

Implementación del servicio web PFCWS

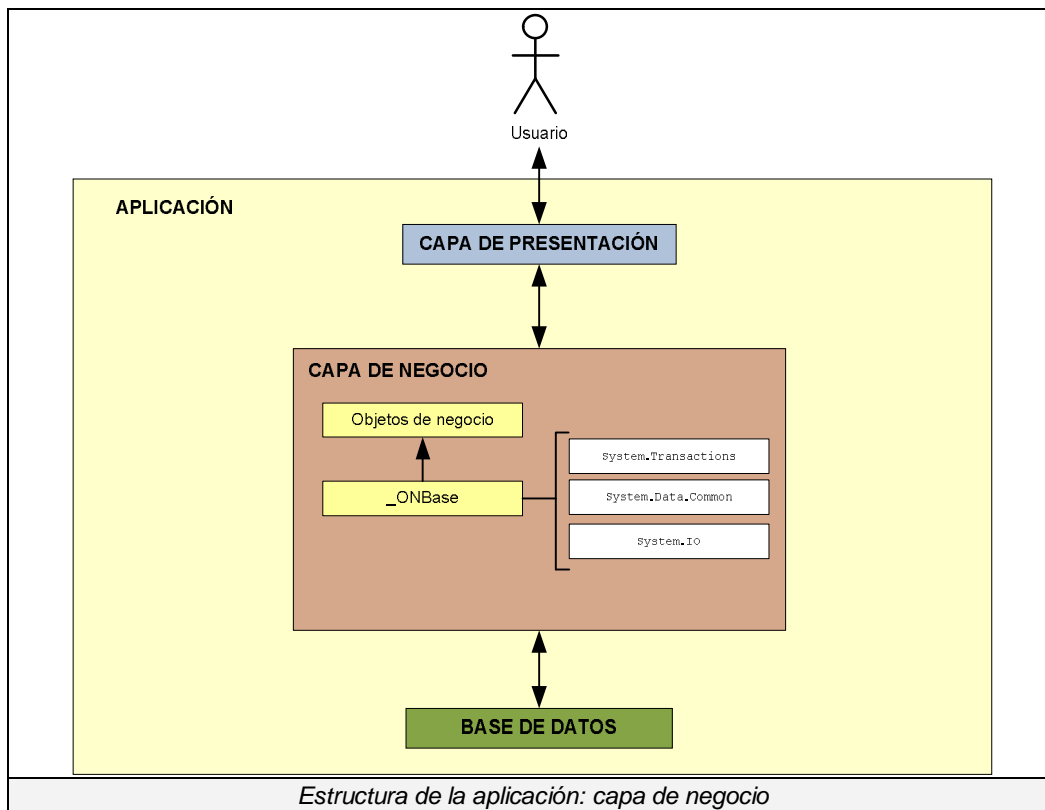
Clases de soporte: además de los diferentes elementos de la capa de presentación descritos, también se crearán una serie de clases que agruparán métodos según su funcionalidad y que podrán ser utilizadas desde cualquier parte de la aplicación. La clase C_Comun definirá métodos estáticos, no será necesario crear objetos de la misma para utilizarlos. Por ejemplo un método que contendrá esta clase será el que permitirá acceder de forma rápida a los diferentes valores de configuración de la aplicación definidos en el archivo web.config. Otra clase que se creará será la clase C_Ficheros, que ofrecerá métodos para trabajar con carpetas y ficheros físicos. Se clase se utilizará durante la creación y destrucción de sitios web, tareas que implicarán la correspondiente creación y destrucción del árbol de directorios de cada sitio web virtual. Por último se creará una clase _WebSiteConfig, que ayudará a consultar y editar el archivo de configuración que se asociará a cada sitio web virtual.


```

PFC - Visual Web Developer 2005 Express
Archivo  Editar  Ver  Refactorizar  Sitio Web  Generar  Depurar  Herramientas  Ventana  Comunidad  Ayuda
App_Code\PFC_Ficheros.cs  RESX\STYLES\PFC_Base.resx  RESX\MOOLES\M3.asax.resx  App_Code\PFCOWS.cs  WEB\ONES\PAGES\Pafiles.aspx.cs  App_Code\ION\ION_Base.cs
PFC_Ficheros
6  using System.Web.UI;
7  using System.Web.UI.WebControls;
8  using System.Web.UI.WebControls.WebParts;
9  using System.Web.UI.HtmlControls;
10 using System.IO;
11
12 /// <summary>
13 /// Funciones para el manejo de carpetas y ficheros fisicos
14 /// </summary>
15
16 namespace PFC
17 {
18     public class PFC_Ficheros
19     {
20
21         private string PR_ServerPath;
22
23         /// <summary>
24         /// PFC_Ficheros f = new PFC_Ficheros(Server.MapPath("../"));
25         /// </summary>
26         /// <param name="ServerPath"></param>
27         public PFC_Ficheros(string _ServerPath)
28         {
29
30
31
32
33         // Duplica la carpeta CODEBASE para crear la estructura de ficheros fisica de un nuevo sitio web
34         public Exception FH_creaWebSite(string _subdominio)
35         {
36
37
38
39
40
41
42
43         // Elimina los archivos fisicos correspondientes al webSite de un usuario dado
44         public Exception FH_borraWebSite(string _subdominio)
45         {
46
47
48
49
50
51
52
53         // Función extraida de la página web http://www.codenplanet.com/MS/files/copydirectoriosrecursivos.aspx
54         private void copyDirectory(string Src, string Dst)
55         {
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
    
```

Implementación de la clase de soporte PFC_Ficheros

6.2.2 Capa de negocio



Estructura de la aplicación: capa de negocio

Funcionalidad

La capa de negocio se encargará de atender las solicitudes provenientes de la capa de presentación, operar con la base de datos y devolver los resultados al usuario. Esta capa contendrá la mayor parte de lógica de la aplicación: sabrá cómo se relacionan las entidades de base de datos entre ellas y el flujo de acciones a seguir para ejecutar los diferentes casos de uso. A este nivel también se implementarán los controles de transacciones y de errores de la aplicación.

Descripción

Se creará una clase base abstracta para todos los objetos de negocio. Esta clase base contendrá los miembros necesarios (propiedades y métodos) para realizar el control de transacciones, el control de errores y el acceso a base de datos. En esta misma clase se declararán los métodos abstractos básicos de acceso a datos, posteriormente cada objeto de negocio que herede de esta clase deberá implementarlos. Se creará un objeto de negocio por cada entidad de base de datos que se necesite gestionar. Estos objetos básicamente definirán los métodos abstractos heredados de la clase base y adicionalmente implementarán métodos específicos propios de la entidad.

```

1  using ...
13
14
15 namespace PFC.ON
16 {
17
18     /// <summary>
19     /// Clase base abstracta para las clases de Negocio
20     /// No permite instanciación directa
21     /// Obliga a definir los métodos abstractos
22     /// </summary>
23
24     public abstract class ON_base
25     {
26         protected DbConnection con;
27         protected DbProviderFactory factory;
28         protected Exception err;
29
30
31         // Control de transacciones ...
34         protected TransactionOptions transOpt;
35
36         constructor
81
82         Funciones de acceso de datos
93
94         control de transacciones
105
106         Funciones variable Error
119
120         Funciones auxiliares
219

```

Implementación de la clase base para los objetos de negocio

Acceso a base de datos: para acceder a la base de datos se utilizarán las clases del espacio de nombres System.Data.Common del *framework* de .NET. Básicamente, estas clases permiten implementar código de acceso a datos no específico para un tipo de base de datos en concreto. El funcionamiento es el siguiente: cuando se inicializa el sistema se especifica el proveedor de base de datos y la cadena de conexión a utilizar. A partir de este punto, el código de la capa de negocio trabaja con clases genéricas del espacio de nombres Data.Common y el propio *framework* se encarga de instanciar internamente los

objetos específicos para el tipo de base de datos utilizada. Se optará por este planteamiento con la intención de dotar a la aplicación de la mayor compatibilidad posible y pensando en que algún día se trabajará con otras bases de datos como MySQL u Oracle además de con la utilizada (SQL Server). Tanto el proveedor de base de datos como la cadena de conexión se ubicarán en el archivo XML de configuración de la aplicación lo que permitirá no tener que compilar el código cada vez que se necesiten modificar las propiedades de acceso a datos.

```
<add key="dbProvider" value="System.Data.SqlClient"/>
<add key="dbConnStr" value="Data Source=SERVIDOR_PFC\SQLEXPRESS;
    Initial Catalog=PFCdb;User Id=PFCAdmin;Password=XXXXXX;"/>
```

.NET *framework* ofrece un método estático (se puede ejecutar sin necesidad de instanciar un objeto) que permite conocer todos los proveedores de base de datos compatibles con el espacio de nombres System.Data.Common:

```
DataTable dt = System.Data.Common.DbProviderFactories.GetFactoryClasses();
```

Los objetos de negocio dispondrán de dos variables miembro para gestionar el acceso a datos. Estas variables se inicializarán en el constructor a partir de la información contenida en el archivo de configuración y a partir de ahí podrán ser utilizadas por los métodos de acceso a datos para interactuar con la base de datos.

```
protected DbConnection con;
protected DbProviderFactory factory;
```

Variables miembro para el acceso a datos

```
AppSettingsReader appReader = new AppSettingsReader();
string provider = appReader.GetValue("dbProvider", typeof(string)).ToString();
string connStr = appReader.GetValue("dbConnStr", typeof(string)).ToString();

factory = DbProviderFactories.GetFactory(provider);

con = factory.CreateConnection();
con.ConnectionString = connStr;
```

Inicialización del acceso a datos dentro del constructor del objeto de negocio

```
public override DataSet FN_getAll(string psWHERE)
{
    DbCommand cmd = factory.CreateCommand();
    cmd.Connection = con;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "SP_USUARIOSWEB_GET";

    base.FN_addParameter(ref cmd, "@pWHERE", psWHERE, DbType.String);

    DbDataAdapter adap = factory.CreateDataAdapter();
    adap.SelectCommand = cmd;
    DataSet ds = new DataSet();

    try
    {
        adap.Fill(ds, "USUARIOSWEB");
    }
    catch (Exception er)
    {
        err = er;
        return null;
    }

    return ds;
}
```

Ejemplo de método de acceso a datos

Control de transacciones: el principal objetivo del control de transacciones es permitir deshacer los cambios cuando algún proceso de los que se suceden dentro de una misma transacción produce un error. Por ejemplo cuando se elimina un registro de la tabla Noticias de la base de datos también hay que eliminar todos sus comentarios. Siguiendo el orden marcado por la integridad referencial primero se borran los comentarios y luego la noticia. Estas operaciones tienen que enmarcarse dentro de una misma transacción puesto que si el proceso de borrar la noticia da error, se tendrá que cancelar el borrado de comentarios hecho previamente.

Para implementar este mecanismo se utilizarán las clases del espacio de nombres System.Transactions de .NET framework. El funcionamiento es sencillo: se abre un ámbito de transacción, se realizan las operaciones contra la base de datos y, si todo ha ido bien, se ejecutan los cambios, si no, se deshacen. Se habilitará un método en la clase base de los objetos de negocio encargado de abrir un ámbito de transacción.

```
protected TransactionScope FN_abrirTransaccion()
{
    return new TransactionScope(TransactionScopeOption.Required, transOpt);
}
```

Método encargado de abrir ámbitos de transacción

Cada ámbito de transacción está representado por un objeto de la clase TransactionScope. Es muy importante inicializar debidamente estos objetos para obtener el comportamiento deseado. En este caso se utilizará el parámetro TransactionScopeOption.Required, que implica que, si cuando se abre un ámbito de transacción ya existe otro abierto (ya existe una transacción ambiente), el nuevo ámbito queda subordinado al primero de forma que sólo la validación del ámbito raíz provocará que los cambios sobre la base de datos se hagan efectivos. De este modo, estando dentro de un ámbito de transacción, se podrán invocar otros métodos que a su vez también abran ámbitos de transacción.

```
public void FN_DeleteFull(int pID)
{
    // Borra los registros de la tabla T_COMENTARIOS i luego borra T_NOTICIAS
    int rc = 0;

    using (TransactionScope transScope = FN_abrirTransaccion())
    {

        #region Borramos los registros de la tabla T_COMENTARIOS

        ON_COMENTARIOS onC = new ON_COMENTARIOS();
        DataSet dsC = new DataSet();
        dsC = onC.FN_getAll("IDNOTICIA=" + pID.ToString());
        rc = 0;
        while (!onC.FN_hayError() && rc < dsC.Tables[0].Rows.Count)
        {
            onC.FN_Delete(Convert.ToInt32(dsC.Tables[0].Rows[rc]["IDCOMENTARIO"]));
            rc++;
        }

        if (onC.FN_hayError())
            err = onC.FN_getErr();

        #endregion

        #region Borramos el registro de la tabla T_NOTICIAS

        if (!FN_hayError())
            FN_Delete(pID);

        #endregion

        #region fin de la transacción i retorno de la función
        // Si no ha habido errores completamos la transacción
        // Si los ha habido el error quedará en this.err, liberamos la transacción y retornamos
        if (err == null)
            transScope.Complete();

        transScope.Dispose();

        #endregion
    }
}
```

Ejemplo de método de objeto de negocio utilizando control de transacción

Para efectuar el control de transacción se abre un ámbito de transacción. Dentro de ese ámbito se ejecutan las operaciones necesarias sobre la base de datos. Finalmente, si no ha habido error, se llama al método *Complete()* del objeto *TransaccioScope* que se está utilizando. Por último se liberan los recursos mediante el método *Dispose()*. Si algo va mal y hay errores basta con salir del ámbito de transacción sin llamar a *Complete()* para que los cambios no se hagan efectivos.

Control de errores: la aplicación dispondrá de un archivo interno en el que se irán registrando los errores en tiempo de ejecución que se puedan producir. El flujo de los errores será el siguiente: si se produce un error en la base de datos pasará a la capa de negocio; si se produce un error en la capa de negocio se guardará en una variable miembro de tipo *Exception* de la propia clase de negocio; la capa de presentación, tras utilizar la capa de negocio, comprobará el estado de dicha variable para saber si ha habido errores, en caso negativo seguirá la ejecución normal y en caso afirmativo los lanzará para que se registren en el log de errores de la aplicación.

```
protected Exception err;
public Exception FN_getErr()
{
    return err;
}
public bool FN_hayError()
{
    return (!(err == null));
}
```

Código para el control de errores de la clase base de los objetos de negocio

Todos los objetos de negocio heredarán de la clase base una variable miembro de tipo *Exception* y dos métodos públicos para ser utilizados desde la capa de presentación (una para saber si ha habido error durante la ejecución de la última función y otro para obtener el error propiamente dicho).

En las páginas web (capa de presentación), tras utilizar un método de un objeto de negocio, se comprobará si ha habido error y en caso afirmativo se decidirá qué hacer: mostrar el error por pantalla y/o registrar el error en el fichero log de errores.

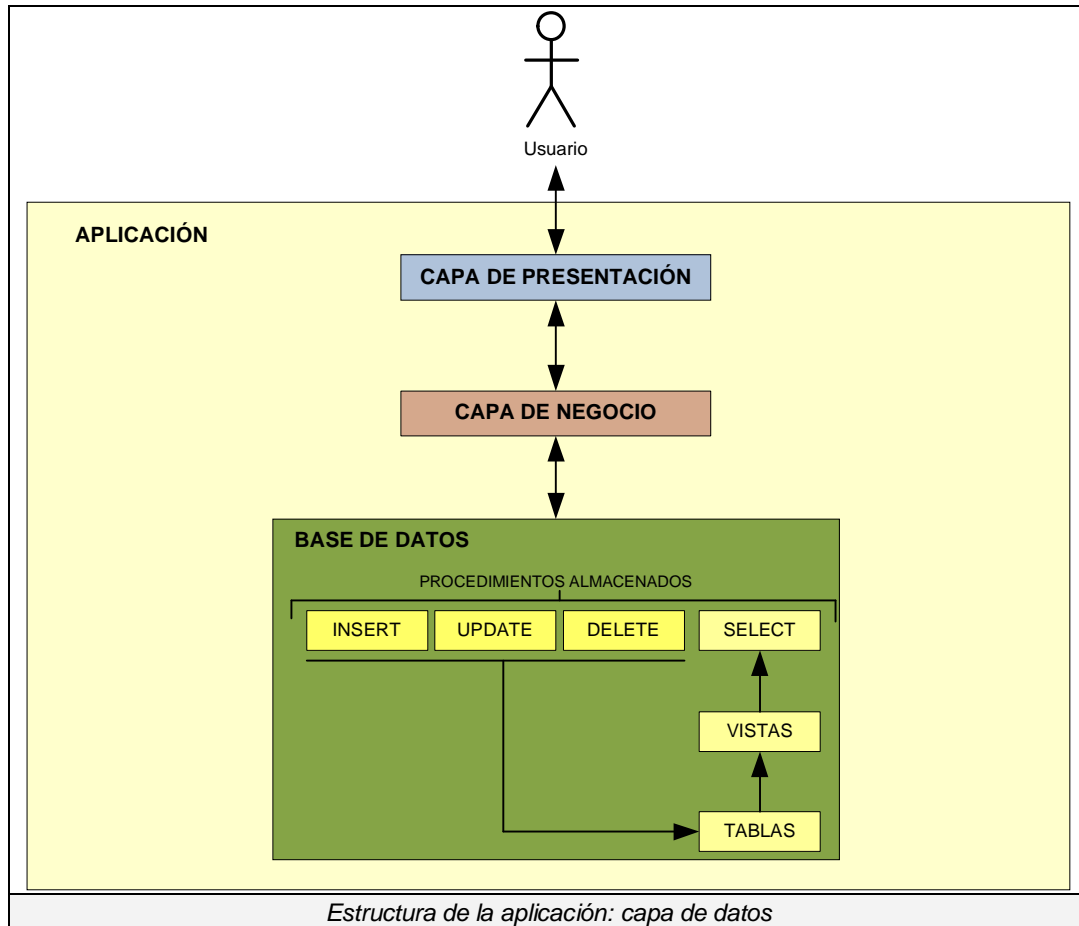
El mecanismo de registro de errores en el fichero log se implementará mediante la clase *Global*. La aplicación utilizará el archivo *Global.asax* (denominado *Global.asa* en versiones de ASP anteriores) para interceptar los errores lanzados desde cualquier punto de la aplicación y reflejarlos en el log de errores. La clase *Global.cs* es muy útil para definir el comportamiento de la aplicación durante determinados eventos, como pueden ser inicio y fin de sesión o el ya mencionado error en tiempo de ejecución.

```
11 using System.IO;
12 |
13 public partial class Global : System.Web.HttpApplication
14 {
15     void Application_Error(object sender, EventArgs e)
16     {
17
18         Exception ex = Server.GetLastError();
19
20         if (ex.InnerException != null)
21             ex = ex.InnerException;
22
23         Random objeto = new Random((int)DateTime.Now.Ticks);
24         int numero = objeto.Next(100000);
25
26         this.WriteToLog(ex, numero);
27
28     }
29
30
31
32     private void WriteToLog(Exception err, int numero) { ... }
33 }
```

Código encargado de registrar los errores perteneciente a la clase Global

Cuando desde cualquier punto de la capa de presentación se lance un error, se invocará automáticamente el método `Application_Error` de la clase `Global`. Este método asigna un número al error y lo registrará en el fichero de registro de errores.

6.2.3 Capa de datos



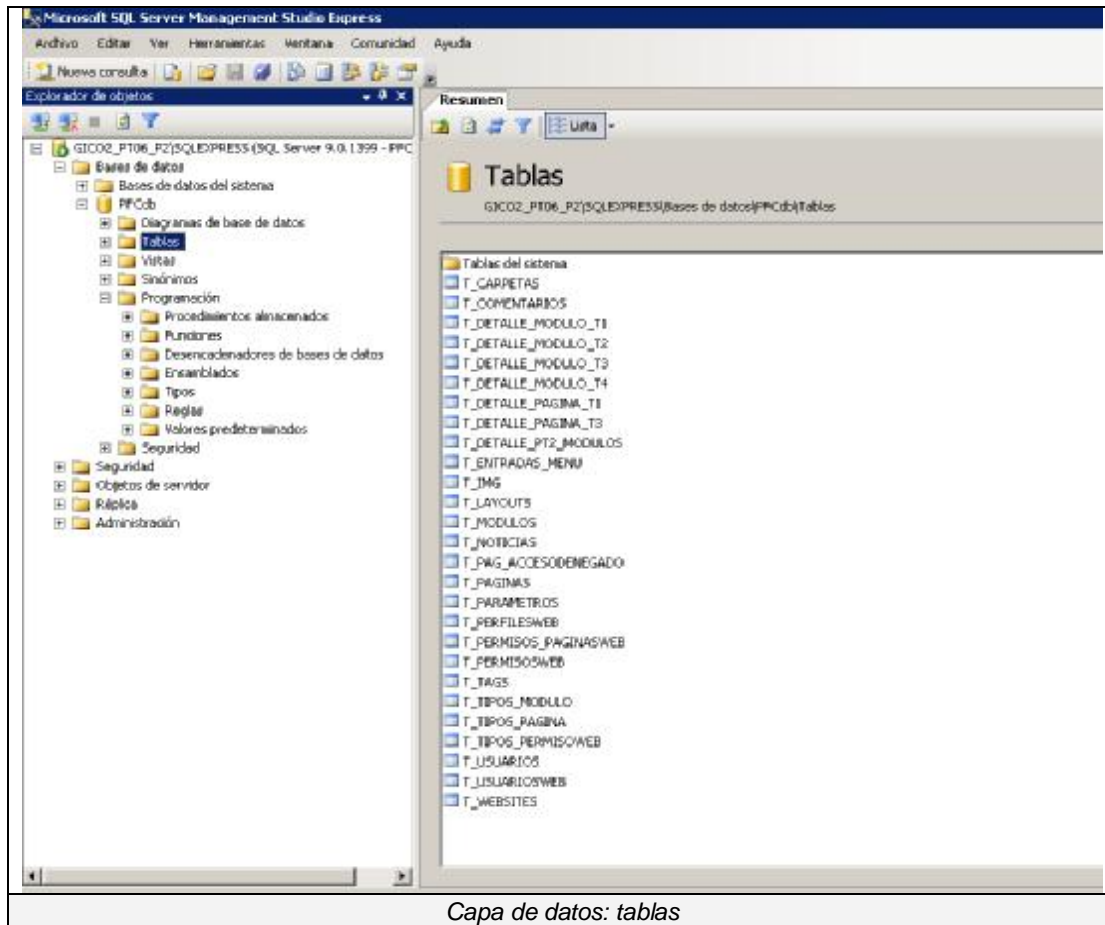
Durante la implementación de la aplicación se evitará en la medida de lo posible insertar código SQL tanto en la capa de negocio como ni mucho menos en la de presentación. Esto se hará por tres motivos:

- 1) Por seguridad (para evitar posibles ataques mediante técnicas de *SQL injection*).
- 2) Para mantener la compatibilidad con múltiples sistemas de bases de datos obtenida gracias al uso del espacio de nombres `System.Data.Common` (si se utiliza SQL en el código tiene que ser totalmente estándar y no específico de ningún sistema de bases de datos).
- 3) Para mejorar el rendimiento de la aplicación (el servidor web suele tener más carga de trabajo que el servidor de bases de datos, la intención es hacer todo el trabajo posible a nivel de base de datos para que el IIS pueda dedicar sus recursos a otras tareas como el proceso de la lógica de la aplicación).

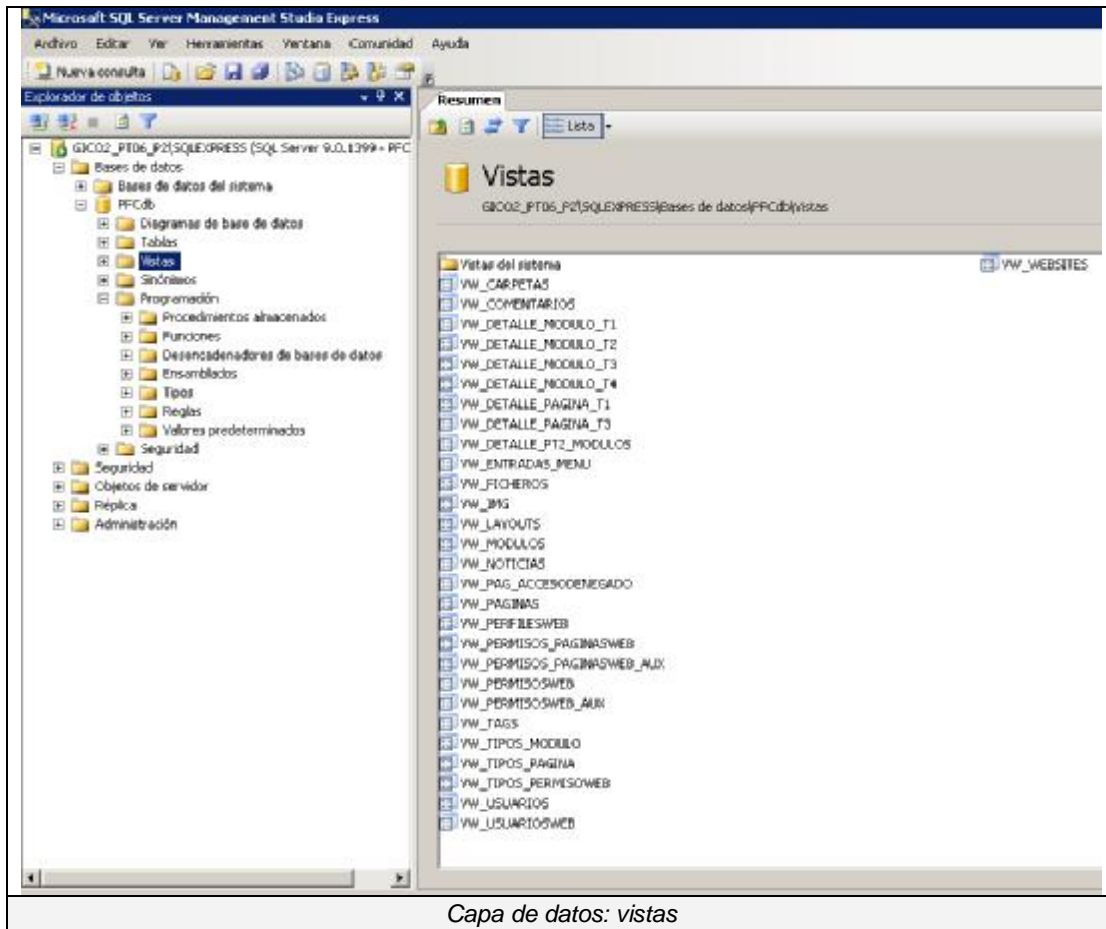
A nivel de base de datos cada tabla que se quiera gestionar desde la aplicación tendrá su correspondiente vista, que mostrará, además de todos los campos de la tabla, los campos de las tablas referenciadas que se necesiten. Para cada tabla también habrá los procedimientos almacenados necesarios para llevar a cabo las operaciones básicas: insertar un nuevo registro, editar un registro, eliminar un registro y consultar los registros de la tabla aplicando un filtro.

De este modo, desde la capa de negocio, el objeto de negocio correspondiente a la tabla X siempre hará llamadas a los procedimientos almacenados de la tabla X, pasándoles los parámetros necesarios, y nunca lanzará consultas SQL directamente. Si se necesitan efectuar operaciones más allá de las cuatro habituales (por ejemplo, obtener el código más alto de cierta tabla para calcular el código del próximo

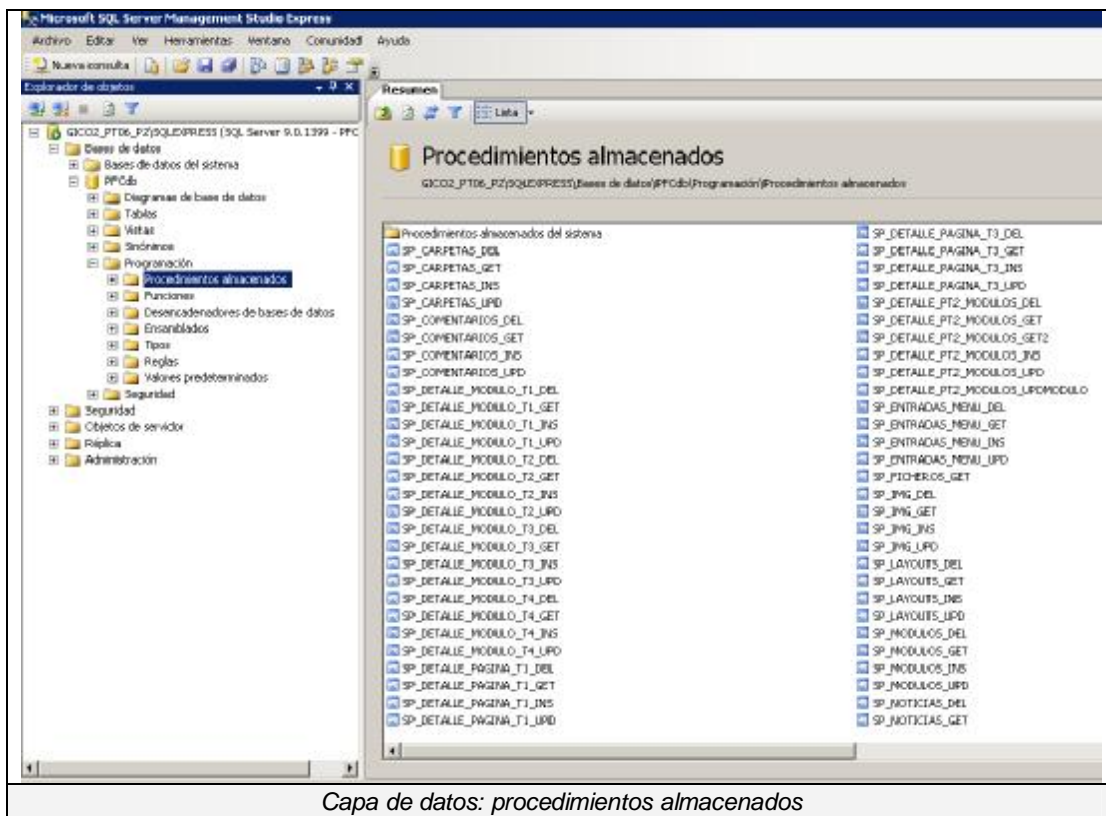
registro) se crearán procedimientos almacenados específicos y sus correspondientes métodos de llamada específicos dentro de los objetos de negocio que lo necesiten.



Capa de datos: tablas



Capa de datos: vistas



Capa de datos: procedimientos almacenados

Con este planteamiento, para migrar la aplicación hacia otro sistema gestor de bases de datos lo único que habrá que hacer será crear en el nuevo sistema una base de datos con la misma estructura que la original (tablas con los mismos campos, vistas y procedimientos almacenados con los mismos parámetros). Tras ello bastará con cambiar la cadena de conexión y el proveedor de base de datos en el archivo web.config de la aplicación. Los objetos de negocio seguirán llamando a procedimientos con el mismo nombre y pasando los mismos parámetros y la aplicación seguirá funcionando independientemente del sistema de base de datos que se esté utilizando.

6.3 Herramienta de soporte para la implementación

Debido a la estructura de las tres capas descrita, el código de la aplicación será muy homogéneo: a partir de cierta tabla, los procedimientos almacenados correspondientes serán siempre igual, variando solamente el número y tipo de campos. Y su objeto de negocio también será siempre igual, heredando del objeto de negocio base y cambiando sólo las partes que atañan a los campos de la tabla. Para evitar tareas repetitivas durante el proceso de implementación se creará una sencilla aplicación que, a partir de una tabla de la base de datos, genere el código para crear sus correspondientes procedimientos almacenados y su objeto de negocio. Ello permitirá ahorrar mucho tiempo y eliminar el riesgo de error durante esta parte de la implementación.



Interficie de la aplicaci3n de generaci3n de c3digo

Para generar el c3digo la aplicaci3n se basar3 en plantillas y en el nombre y los campos de la tabla seleccionada.

```
public override DataRow FN_Insert(DataRow prFILa)
{
    DbCommand cmd = factory.CreateCommand();
    cmd.Connection = con;
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "SP_%NOM_ENTITAT%_INS";

    #region Parametros SP

        %PARAMETRES_INSERT%

    #endregion

    DbDataAdapter adap = factory.CreateDataAdapter();
    adap.SelectCommand = cmd;
    DataSet ds = new DataSet();

    try
    {
        adap.Fill(ds, "%NOM_ENTITAT%");
    }
    catch (Exception er)
    {
        err = er;
        return null;
    }

    return ds.Tables["%NOM_ENTITAT%"].Rows[0];
}
```

Fragmento de plantilla encargado de generar la llamada al procedimiento almacenado de INSERT

```
CREATE PROCEDURE [dbo].[SP_%NOM_ENTITAT%_INS]
(
    %PARAMETRES_ENTRADA%
)
AS
SET NOCOUNT OFF;

INSERT INTO %NOM_TAULA%
    (%CAMPS_INSERT%)
VALUES
    (%VALORS_INSERT%);

SELECT
*
FROM %NOM_TAULA%
WHERE
    ( %CAMP_ID% = IDENT_CURRENT('%NOM_TAULA%') )
```

Fragmento de plantilla encargado de generar el procedimiento almacenado de INSERT

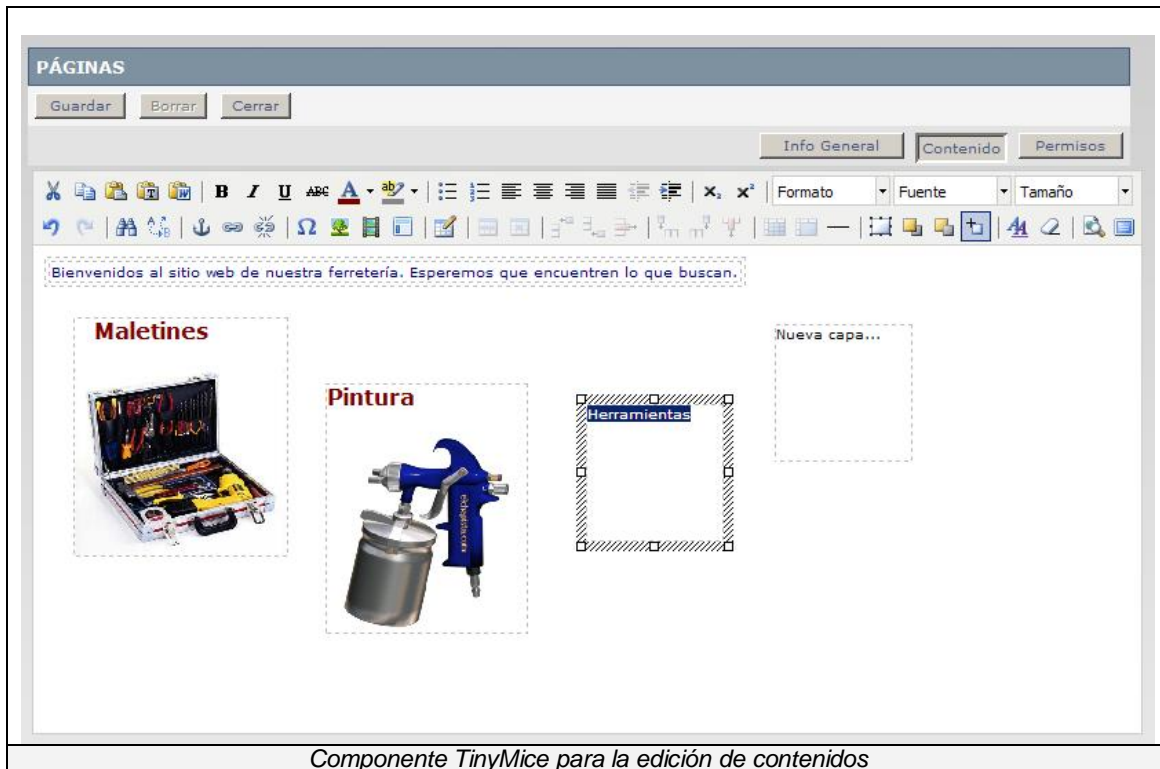
Lógicamente, si se cambia el sistema gestor de base de datos, se tendrán que cambiar las plantillas para que el código SQL generado se corresponda con el nuevo sistema.

Los objetos de negocio generados mediante esta aplicación contendrán las llamadas básicas a base de datos, si se necesitan métodos específicos habrá que añadirlos posteriormente de forma manual (previa creación de los procedimientos almacenados correspondientes).

6.4 Componentes externos utilizados

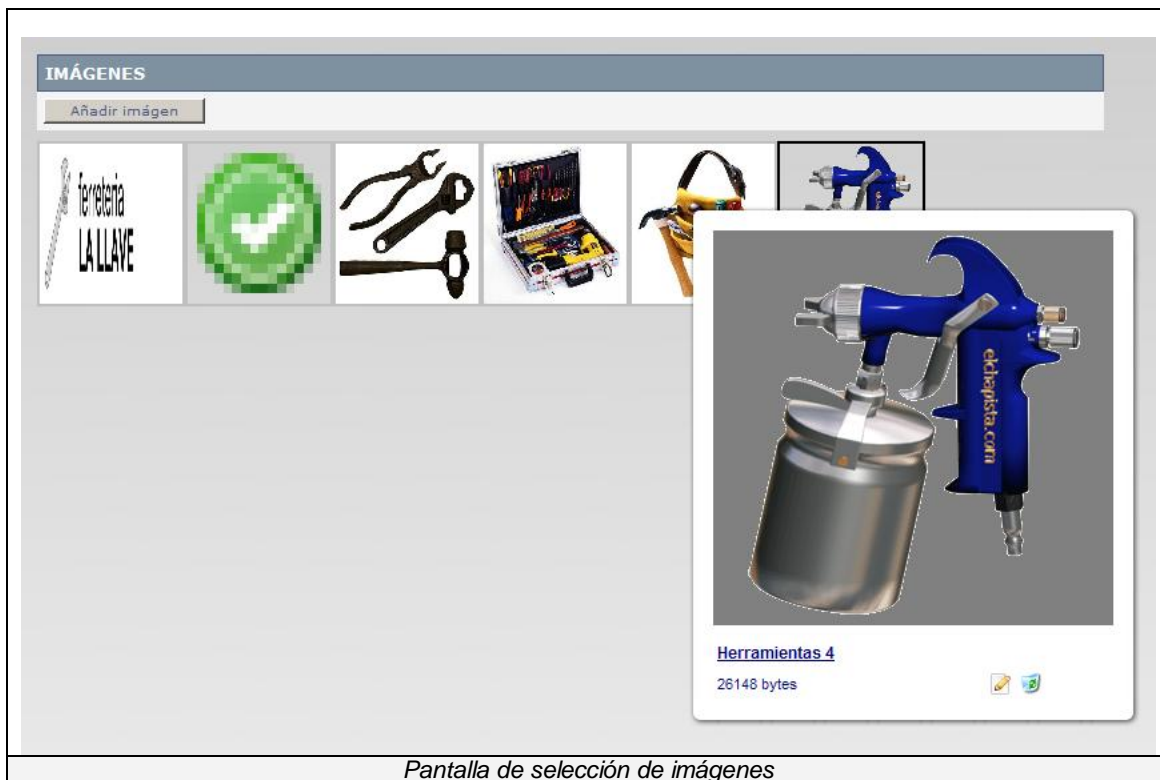
Durante el desarrollo de la aplicación se utilizarán ciertos componentes de libre distribución que para cubrir funcionalidades muy concretas.

TinyMice: editor de código HTML utilizado en la pantalla de edición de contenido de las páginas web de carácter general. Página web oficial: <http://tinymce.moxiecode.com>



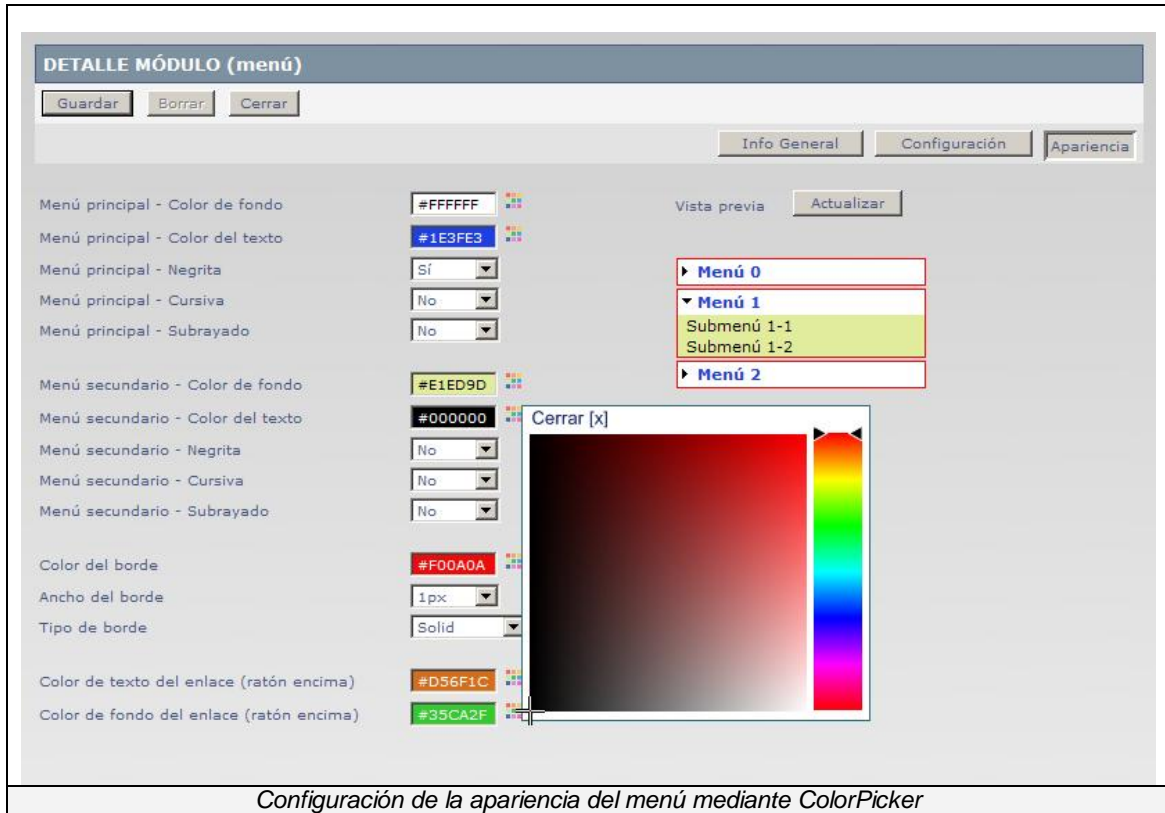
Componente TinyMice para la edición de contenidos

HighSlide JS: conjunto de scripts y hojas de estilo utilizados en la pantalla de selección de imágenes. Permite navegar por las imágenes de una forma fácil, práctica y atractiva para el usuario. Sitio web oficial: <http://highslide.com>.



Pantalla de selección de imágenes

Color Picker: elemento que permite abrir un cuadro de diálogo para que el usuario seleccione un color. Utilizado en las pestañas de configuración de la apariencia de los diferentes módulos.



Menú Expander: script base utilizado para crear los menús de los sitios web virtuales.

Redips Drag and Drop: scripts para arrastrar y soltar componentes por la pantalla mediante el ratón. Utilizado para las páginas contenedoras de módulos. Página web oficial: <http://www.redips.net/javascript/drag-and-drop-table-content>.



6.5 Instalación y configuración del servidor

La aplicación necesita un servidor web (IIS) y un servidor de bases de datos (SQL Server). Ambos servicios pueden estar en la misma máquina o en equipos separados, el proceso de instalación será en todo caso el mismo.

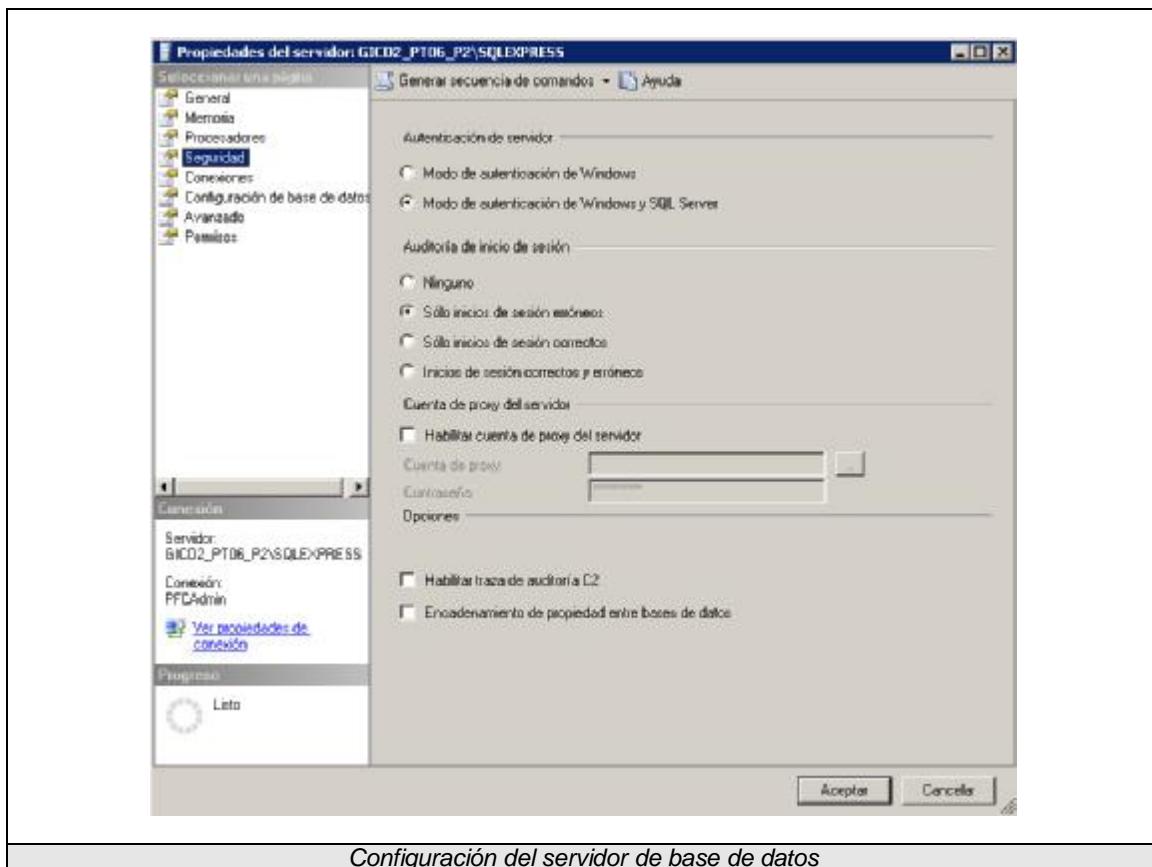
6.5.1 Base de datos

Para la base de datos se necesitará SQL Server 2005 Express Edition. Este software se puede descargar de forma gratuita desde su página oficial. Su instalación es muy sencilla. Durante la instalación se deberá especificar la contraseña del usuario administrados de la base de datos (en versiones anteriores de SQL Server se permitía dejar la contraseña de administrador en blanco, por motivos de seguridad en esta versión ya no se puede).

Para administrar la base de datos SQL Server 2005 Express es muy recomendable instalar la herramienta Microsoft SQL Server Management Studio Express (gratuita). Sin este componente, el servicio de base de datos estará activo pero para interactuar con él se debería hacer desde la línea de comandos.

Tras la instalación se deberán realizar las siguientes tareas sobre la base de datos:

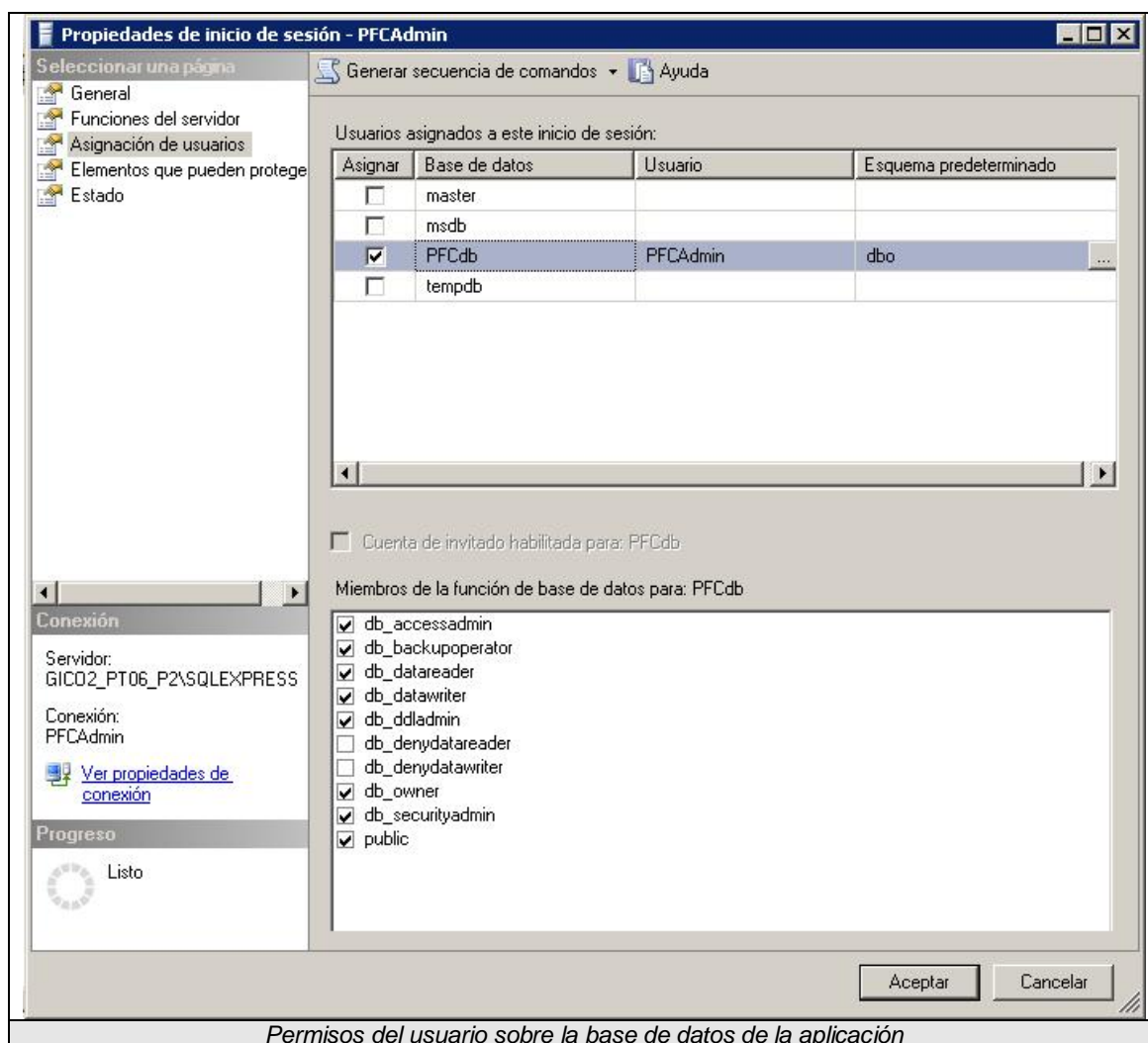
Configuración de parámetros generales: acceder a Management Studio, botón derecho sobre el nombre del servidor, opción Propiedades. En la pestaña **Seguridad**, el parámetro *Autenticación del servidor* deberá tener el valor *Modo de autenticación de Windows y SQL Server*. Esto permite que usuarios creados dentro de la base de datos se puedan conectar a la misma. De otro modo, solo podrían acceder a la base de datos los usuarios del sistema operativo habilitados. La aplicación que se está instalando utilizará su propio usuario de base de datos para realizar las conexiones con la misma por lo que se debe habilitar este tipo de conexión.



Si el servidor web se ubicará en otro equipo, en la pestaña **Conexiones** tendrá que marcarse la casilla *Permitir conexiones remotas con este servidor*.

Tras configurar estas propiedades se procederá a crear la base de datos. Para la instalación lo que se hará será partir de una copia vacía de la base de datos, no crear una base de datos en blanco. Botón derecho sobre *Bases de datos*, opción *Restaurar base de datos*. A la base de datos se le puede dar el nombre que se desee, luego se configurará el archivo *web.config* de la aplicación para que apunte al repositorio correcto. Se utilizará la opción *Restaurar desde dispositivo* para examinar y utilizar el archivo que contenga la copia de la base de datos. Esta copia inicial en realidad no estará vacía del todo ya que contendrá registros básicos para el funcionamiento de la aplicación (como los tipos de página y de módulo disponibles, los tipos de permiso disponibles o un usuario administrador ADMIN con contraseña PFC123).

Restaurada la base de datos solo quedará crear el usuario de base de datos que utilizará la aplicación. Dentro del menú Seguridad, botón derecho sobre la opción **Inicios de sesión**, opción *Nuevo inicio de sesión*. Se podrá utilizar el nombre y contraseña que se desee, posteriormente habrá que editar el archivo *web.config* de la aplicación en base a estos valores. Como base de datos predeterminada se seleccionará la base de datos de la aplicación. En la pestaña *Funciones de servidor* no hará falta seleccionar ninguna. En la pestaña *Asignación de usuarios* se seleccionará la base de datos de la aplicación y se concederán al nuevo inicio de sesión todos los permisos excepto *db_denydatareader* y *db_denydatawriter*.



Permisos del usuario sobre la base de datos de la aplicación

Será muy importante, en la pestaña **Estado**, conceder al usuario permiso de conexión a la base de datos y habilitar el inicio de sesión. Tras ello se pulsará Aceptar para crear el usuario.

Es importante recordar que, si el servidor en el que se ha instalado la base de datos no está sujeto a ninguna política de copias de seguridad, habrá que configurar los trabajos de copia de seguridad para evitar posibles desastres.

Con esta serie de pasos la base de datos quedará configurada, si algo no se ha hecho bien, cuando la aplicación trate de acceder a la misma se producirá un error y éste será mostrado por pantalla.

6.5.2 Servidor web

El servidor web constará de una máquina con sistema operativo windows, IIS y .NET framework versión 2.0.

El framework se instala con las actualizaciones automáticas de Windows. En caso de no tener el sistema actualizado, se puede descargar e instalar de forma gratuita.

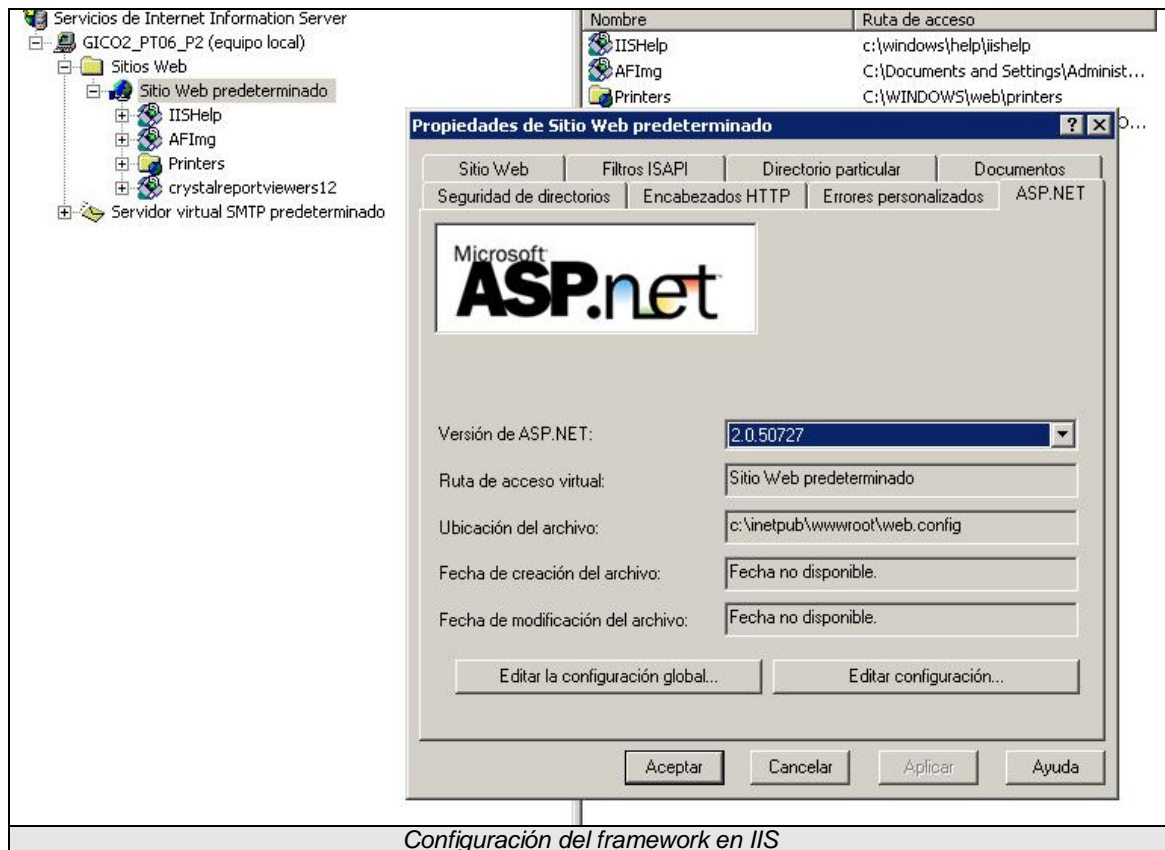
IIS es un componente del propio sistema operativo. Para instalarlo se deberán seguir estos pasos: acceder al panel de control, opción *Agregar o quitar programas*, *Agregar o quitar componentes de Windows*, **Servicios de Internet Information Services (IIS)**.



Instalación del servidor web

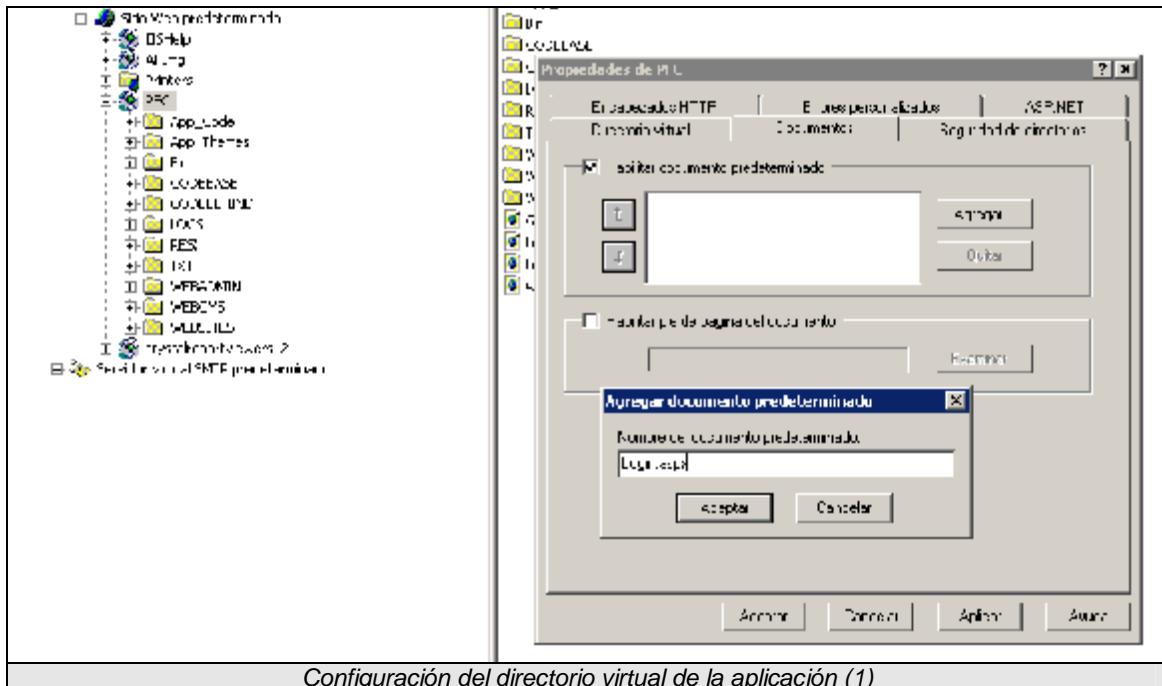
Para llevar a cabo la instalación se requerirá el CD o DVD original del sistema operativo.

Tras ello, se accederá a la consola de administración de IIS para instalar la aplicación. *Panel de control*, *Herramientas administrativas*, **Servicios de Internet Information Server**. Lo primero que se tendrá que hacer será acceder a las propiedades del sitio web predeterminado y establecer la versión de .NET framework a utilizar (la 2.0 en este caso).



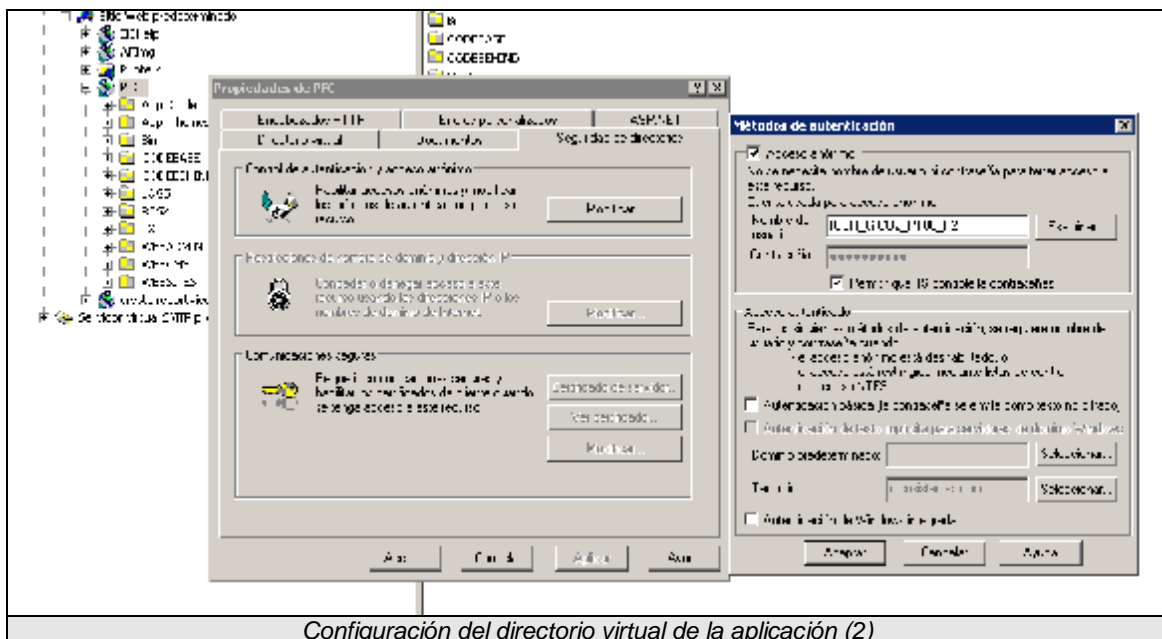
Configuración del framework en IIS

A continuación se procederá a crear el directorio virtual que servirá para hacer funcionar la aplicación. Botón derecho sobre el sitio web predeterminado, opción *Nuevo directorio virtual*. Se puede utilizar el *Alias* que se desee, teniendo en cuenta que la ruta final de la aplicación será del tipo `http://[nombre del servidor]/[Alias del directorio virtual]`. El directorio virtual tendrá que apuntar a la raíz del directorio físico donde se situen los archivos de la aplicación. Para que las páginas `aspx` se ejecuten bien será muy importante, en la siguiente pestaña, marcar la casilla *Ejecutar*. Finalmente se pulsa aceptar para crear el directorio virtual de la aplicación. A continuación se configurarán sus propiedades, pulsando botón derecho sobre el mismo, opción *Propiedades*. En la pestaña *Documentos*, habilitar documento predeterminado, eliminar los documentos que se ofrecen por defecto y añadir la pantalla `Login.aspx`.



Configuración del directorio virtual de la aplicación (1)

En la pestaña *Seguridad de directorios* habrá que habilitar el acceso anónimo a la aplicación y permitir que IIS controle la contraseña.



Configuración del directorio virtual de la aplicación (2)

Por último en la pestaña *ASP.NET* habrá que comprobar que efectivamente se esté utilizando el framework 2.0 para ejecutar la aplicación.

6.5.3 Otras configuraciones

Debido a que la aplicación utiliza control de transacciones para su funcionamiento, será indispensable que en el servidor web esté activo el servicio MSDTC (Coordinador de transacciones distribuidas de Microsoft, msdtc.exe). Para configurar este servicio se puede hacer desde *Panel de control, Herramientas administrativas, Servicios*, o desde *Panel de control, Herramientas administrativas, Servicios de componentes*.

Como último pasó se deberá editar el archivo web.config ubicado en el directorio raíz de la aplicación para especificar el nombre de la base de datos y el usuario y contraseña de conexión. Web.config es un archivo XML por lo que se puede utilizar el propio Notepad de Windows para realizar esta tarea. Estos valores se encuentran en la clave denominada `dbConnStr`.

Finalizada la instalación, se comprobará su funcionamiento accediendo mediante el navegador web a la dirección `http://[nombre del servidor]/[Alias del directorio virtual]`.

Enlaces de las descargas:

Producto	Enlace
SQL Server 2005 Express	http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=220549b5-0b07-4448-8848-dcc397514b41
SQL Server Management Studio Express	http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=c243a5ae-4bd1-4e3d-94b8-5a0f62bf7796
.NET framework 2.0	http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displayLang=es

CAPÍTULO 7: PRUEBAS

El testeo de la aplicación se ha llevado a cabo en dos etapas: la primera durante el desarrollo de la misma y la segunda una vez finalizada.

Durante la primera etapa, para cada entidad funcional se han realizado las siguientes pruebas básicas:

- Crear un nuevo registro.
- Modificar un registro existente.
- Eliminar un registro existente.

Adicionalmente, se ha puesto especial interés en verificar los siguientes aspectos:

- Campos obligatorios cumplimentados.
- Caracteres especiales en los datos de entrada.
- Control de la integridad referencial.
- Funcionamiento correcto de la capa de presentación tanto en Internet Explorer como en Mozilla Firefox.

Se detectaron algunos problemas de vulneración de la integridad referencial o de campos obligatorios no cumplimentados que fueron corregidos.

```
*****
# 46302
-----
12/12/2009 18:23:59
-----
MISSATGE ERROR
-----
Instrucción DELETE en conflicto con la restricción REFERENCE "FK_T_DETALLE_MODULO_T1_T_IMG". El conflicto ha aparecido en la base de datos "PFCdb", tabla
"dbo.T_DETALLE_MODULO_T1", column 'IDIMG'.
Se terminó la instrucción.
-----
STACKTRACE
-----
en WEBCMS_PAGES_ImgEdit.btnEliminar_Click(Object sender, EventArgs e) en c:\Documents and Settings\Administrador.GICO2_PT06_P2\Mis documentos\Visual
Studio 2005\WebSites\VWD\PFC\WEBCMS\PAGES\ImgEdit.aspx.cs:línea 185
en System.Web.UI.WebControls.Button.OnClick(EventArgs e)
en System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)
en System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument)
en System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument)
en System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData)
en System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
*****
```

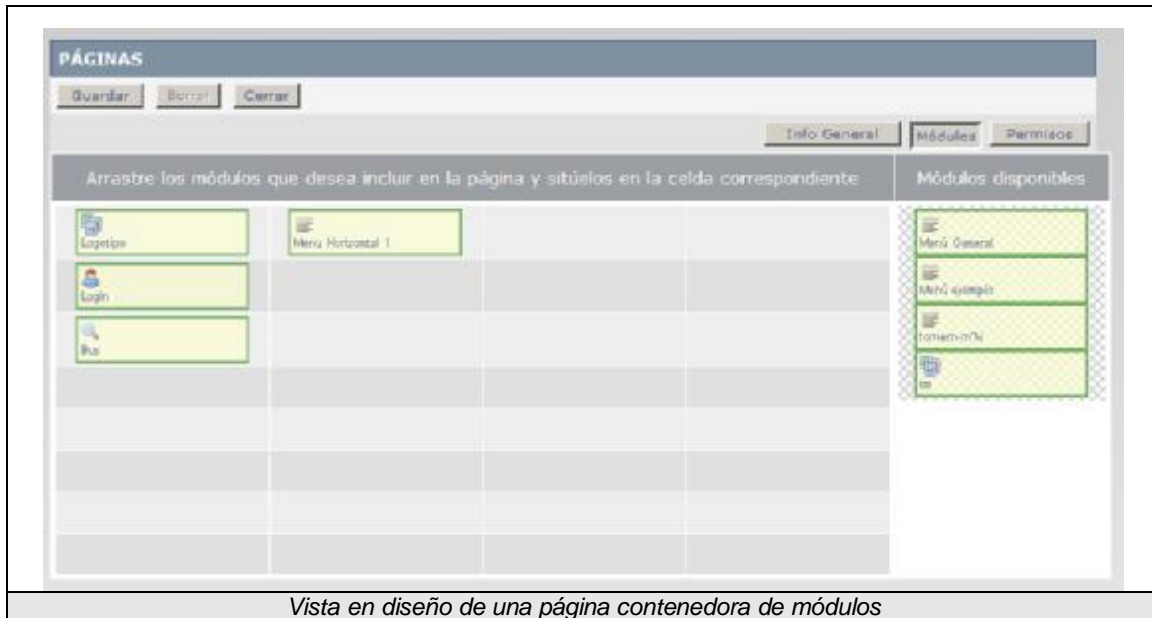
Muestra del archivo de registro de errores: problema de integridad referencial

```
*****
# 77426
-----
02/02/2010 19:43:22
-----
MISSATGE ERROR
-----
No se puede insertar el valor NULL en la columna 'USULOGIN', tabla 'PFCdb.dbo.T_USUARIOSWEB'. La columna no admite valores NULL. Error de INSERT.
Se terminó la instrucción.
-----
STACKTRACE
-----
en WEBCMS_PAGES_UsuariosEdit.btnGuardar_Click(Object sender, EventArgs e) en c:\Documents and Settings\Administrador.GICO2_PT06_P2\Mis
documentos\Visual Studio 2005\WebSites\VWD\PFC\WEBCMS\PAGES\UsuariosEdit.aspx.cs:línea 105
en System.Web.UI.WebControls.Button.OnClick(EventArgs e)
en System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument)
en System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument)
en System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument)
en System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData)
en System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
*****
```

Muestra del archivo de registro de errores: campos obligatorios no cumplimentados

Junto con estas pruebas técnicas durante el desarrollo del proyecto se han llevado a cabo las correspondientes pruebas funcionales.

El principal inconveniente detectado es el hecho de que en las páginas contenedoras de módulos es difícil establecer la ubicación exacta de los módulos. En las siguientes imágenes se muestra una página contenedora de módulos en diseño y cómo la ven los visitantes del sitio web.



Los módulos no se muestran exactamente donde deberían: el menú debería estar junto al logotipo y alineado abajo, no en medio. Otro problema de este tipo de páginas es que no permiten configurar el color de fondo por lo que siempre aparecen con fondo blanco. La solución a estos inconvenientes pasa por rediseñar el mecanismo de este tipo de páginas: en lugar de mostrarse como tablas HTML en cuyas celdas están los módulos, habría que mostrarlas como una capa contenedora en cuyo interior se dibujen las capas hijas, cada una con su posición relativa según las posiciones fijadas durante el diseño de la página. Adicionalmente se debería permitir seleccionar el color de fondo de la capa contenedora (mediante el control ColorPicker integrado en el entorno de desarrollo del proyecto). Éstos serían aspectos que sin duda formarían parte de la próxima actualización de la aplicación.

Se han realizado otras pruebas como provocar errores dentro de las transacciones para comprobar que se deshacen los cambios, o acceder a páginas web mediante usuarios que no tienen permisos de acceso: los resultados han sido satisfactorios.



Para la segunda etapa de la fase de pruebas se instaló la aplicación en un servidor público. Los objetivos de esta segunda etapa fueron medir el rendimiento de la aplicación en un entorno real y proponer a un usuario inexperto el uso de la herramienta para que aportara conclusiones.

No hubo problemas de rendimiento reseñables, con una conexión a internet de 3Mb la aplicación es perfectamente funcional y se puede utilizar sin ningún problema.

Mediante las pruebas realizadas por el usuario se alcanzaron diversas conclusiones, a continuación se listan las principales:

- No basta con un apartado de ayuda dentro del menú de la aplicación: el usuario necesita ver la ayuda para cada pantalla (necesidad de incorporar un icono de ayuda por pantalla que muestre una descripción resumida del funcionamiento de la misma).
- Elementos demasiado dispersos: tener que dar de alta primero un módulo, luego incluirlo en una página contenedora de módulos y luego tener que especificar que dicha página se muestre en una región concreta de la pantalla son muchos pasos. Quizá habría que sacrificar algo de flexibilidad en pos de un uso más sencillo y directo de la aplicación.
- Sistema de usuarios, perfiles y permisos demasiado complejo. Este sistema actualmente ofrece muchas posibilidades, pero un usuario poco experto en la administración de usuarios y perfiles lo único que desea es poder decir que a una página solo pueden acceder ciertos usuarios y que a otra pueden acceder todos. De todos modos se considera el planteamiento del sistema actual perfectamente válido y se sugiere al usuario que practique un poco más.
- Falta de variedad de módulos. Por falta de recursos (tiempo principalmente) no se han incluido módulos como las encuestas, un chat, algún tipo de tablón de ofertas, etc. Este problema se puede solucionar con el desarrollo de nuevas actualizaciones.

CAPÍTULO 8: CONCLUSIONES

Una vez finalizado el desarrollo del proyecto la valoración general es muy positiva. Se considera que el mayor logro han sido los conocimientos adquiridos, sobretudo en lo referente al control de transacciones y al enlace de la aplicación con la base de datos. Hasta antes de este desarrollo estos aspectos se enfocaban de un modo más rudimentario y menos optimizado. También se valora la experiencia adquirida en el entorno de programación Visual Web Developer y en el sistema gestor de bases de datos SQL Server, así como en la programación de aplicaciones web mediante ASP.NET compatibles con el navegador Mozilla Firefox. Por el contrario, el aspecto más negativo ha sido el fallo en la estimación de los tiempos requeridos para las diferentes etapas del proyecto. Ciertamente es que la disponibilidad del equipo de desarrollo no era del todo predecible por estar sujeta a factores externos, pero la estimación falló en tanto que no se acotaron lo suficiente los requisitos funcionales y se fijaron demasiados objetivos a cumplir.

A continuación se muestra una comparación entre la planificación inicial y el desarrollo real del proyecto, y se especifican los objetivos que por falta de tiempo no se han podido alcanzar y que se posponen para futuras actualizaciones de la aplicación.

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Análisis de productos similares	5 días	lun 12/01/09	vie 16/01/09	
2		Estudio de viabilidad	5 días	lun 19/01/09	vie 23/01/09	1
3		Elaboración del listado de control	5 días	lun 26/01/09	vie 30/01/09	2
4		Diseño	25 días	lun 02/02/09	vie 06/03/09	3
5		Implementación	35 días	lun 09/03/09	vie 24/04/09	4
6		Pruebas y desarrollos correctivos	7 días	lun 27/04/09	mar 05/05/09	5
7		Elaboración de la memoria del proyecto	15 días	mié 06/05/09	mar 26/05/09	6

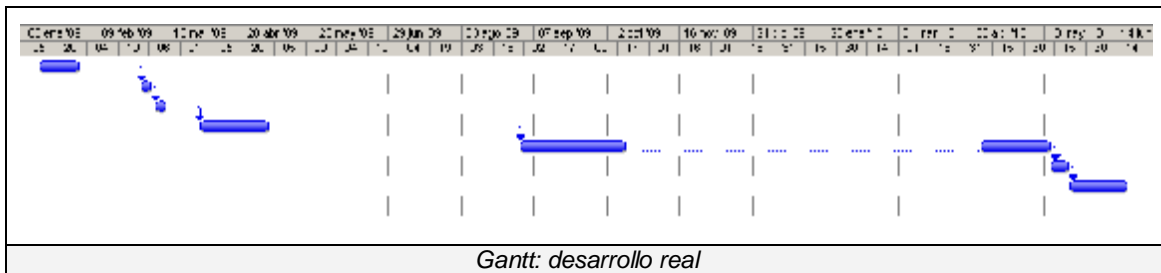
Etapas del proyecto: planificación inicial

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Análisis de productos similares	15 días	lun 12/01/09	vie 30/01/09	
2		Estudio de viabilidad	5 días	lun 02/02/09	vie 06/02/09	1
3		Elaboración del listado de control	5 días	lun 09/02/09	vie 13/02/09	2
4		Diseño	25 días	lun 16/02/09	vie 20/03/09	3
5		Implementación	60 días	lun 23/03/09	mar 15/12/09	4
6		Pruebas y desarrollos correctivos	7 días	mié 16/12/09	jue 24/12/09	5
7		Elaboración de la memoria del proyecto	20 días	vie 25/12/09	jue 21/01/10	6

Etapas del proyecto: desarrollo real



Gantt: planificación inicial



Por excesiva carga de trabajo no se han podido dedicar al proyecto las horas diarias planificadas, ello a provocado que algunas tareas se hayan desviado en el número de días. Además, la implementación tuvo que detenerse e incluso el proyecto en sí se tuvo que parar entre tarea y tarea en repetidas ocasiones. A estos problemas hay que añadir otros como por ejemplo que se calculó mal el tiempo a dedicar al estudio de mercado: el equipo de desarrollo del proyecto no había trabajado anteriormente con gestores de contenido por lo que se necesitó un tiempo extra para comprender qué son estas herramientas, qué tipos hay y qué ofrecen actualmente al usuario. El tiempo reservado a la implementación también se estimó mal: simplemente el desarrollo fue más lento de los esperado.

Con todo y, pese a obtener una aplicación perfectamente funcional y con muchas posibilidades, algunas de las especificaciones no pudieron ser completadas, quedando emplazadas para futuras actualizaciones.

Especificaciones no implementadas:

- Estadísticas (usuario administrador y usuario CMS).
- Copias de seguridad de los sitios web virtuales desde la propia interficie CMS.
- Sistema de ayuda online para la interficie CMS.
- Interficies de administración y CMS en varios idiomas.

Además de las citadas, se tendrán en cuenta las siguientes mejoras para futuras actualizaciones:

- Nuevos módulos insertables que aporten funcionalidad a los sitios web (encuestas, chat o inserción de gadgets de Google).
- Comunicación entre los usuarios vía correo electrónico (usuarios CMS con administradores y visitantes del sitio web con usuarios CMS).
- Portal de acceso común para todos los usuarios del sistema CMS, con clasificación de los sitios web más visitados y más frecuentemente actualizados.

Pese a las desviaciones de planificación y a las características que no se han podido implementar, se consideran los resultados obtenidos como muy satisfactorios: se ha desarrollado una aplicación perfectamente funcional, fácil de utilizar, muy escalable gracias a su estructura, y que ofrece unas posibilidades de creación de sitios web solo limitadas por el talento y la imaginación del usuario y por pequeñas carencias funcionales solucionables a corto plazo mediante actualizaciones.

BIBLIOGRAFÍA

Libros

Mantín Sierra, Antonio: *Desarrollo de aplicaciones web con ASP.NET 2.0*. RA-MA editorial, 2007.

Pérez López, César: *Microsoft Sql server 2005. Administración y análisis de bases de datos.* RA-MA editorial, 2006.

VV.AA.: *ASP.NET 2.0*. ANAYA-MULTIMEDIA, 2006.

Stephens, Rod: *Diseño de bases de datos*. ANAYA-MULTIMEDIA, 2009.

Internet

Google

Wikipedia - es.wikipedia.org

MSDN online - msdn.microsoft.com

Devjoker - www.devjoker.com

El blog del Guille - www.elguille.info

Microsoft TechNet - technet.microsoft.com

Codeguru - www.codeguru.com

CodeSnippets - codesnippets.joyent.com

Experts Exchange - www.experts-exchange.com

DotNetClubs - www.dotnetclubs.com

Dynamic Drive - www.dynamicdrive.com

Otros

Apuntes de clase

