



**Universitat Autònoma  
de Barcelona**

**DISEÑO E IMPLEMENTACIÓN  
DE UN PORTAL WEB PARA  
UNA EMPRESA DE SISTEMAS  
DE CONTROL DE ILUMINACIÓN**

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes  
realitzat per  
Joan Alsina Morillo  
i dirigit per  
Rafael Cortés i Fité

Escola Universitària d'Informàtica  
Sabadell, setembre de 2009

El/la sotasingnant, Rafael Cortés i Fité ,  
professor/a de l'Escola Universitària d'Informàtica de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present memòria  
ha estat realitzat sota la seva direcció  
per en Joan Alsina Morillo

I per a que consti firma la present.  
Sabadell, setembre de 2009

-----  
Signat: Rafael Cortés i Fité

## RESUMEN.

Este proyecto consiste en el diseño e implementación de un portal WEB para una empresa que se dedica al desarrollo de aplicaciones para el diseño y control de sistemas de iluminación. Esta empresa desea ampliar los servicios que oferta a sus clientes mediante una aplicación que facilite la descarga de las actualizaciones del software que la empresa desarrolla entre otras funcionalidades.

Además, la empresa diseña modelos de funcionamiento para dispositivos físicos, basados en el protocolo *DMX512* (Digital MultipleX). Estos modelos son de vital importancia para la expansión de la empresa ya que complementan el software de diseño de instalaciones de iluminación. Además, a día de hoy, existen pocas empresas que se dediquen al diseño de estos modelos. Por tanto, también se desea que estos modelos de funcionamiento se distribuyan a través del portal WEB, para mejorar el servicio a sus clientes.

Por otro lado, la implantación del portal WEB puede mejorar la imagen de empresa y ayudar a captar nuevos clientes, por lo que, el portal vendrá complementado con una serie de funcionalidades para la difusión de información relacionada con la empresa.

Por tanto, el objetivo principal del proyecto es facilitar a los clientes de la empresa el acceso a las actualizaciones de software y a los modelos de funcionamiento a través de un portal WEB. De esta manera se mejora el servicio que oferta la empresa en el momento de la solicitud del proyecto. Además, se desea mejorar la imagen de empresa que se ofrece a los clientes y futuros clientes incluyendo información relacionada con la empresa y sus productos.

Para cumplir con estos requisitos la aplicación hará uso del lenguaje de etiquetas *HTML* junto con *CSS*, el lenguaje de programación *PHP* y el un sistema gestor de datos *MySQL* para generar contenidos dinámicos. Además se utilizarán diferentes herramientas que ayuden a cumplir con los requerimientos especificados en el proyecto.

# ÍNDICE

1.	INTRODUCCIÓN.....	- 7 -
1.1.	RESUMEN DE CONTENIDOS.....	- 7 -
1.2.	OBJETIVO Y ALCANCE PREVISTO .....	- 8 -
1.3.	ESTADO DEL ARTE. ....	- 9 -
1.4.	ACERCAMIENTO A LA TECNOLOGÍA.....	- 10 -
2.	ESTUDIO DE VIABILIDAD.....	- 11 -
2.1.	INTRODUCCIÓN AL ESTUDIO DE VIABILIDAD .....	- 11 -
2.2.	OBJETO.....	- 12 -
2.2.1.	Situación actual.....	- 12 -
2.2.2.	Propuesta de mejora.....	- 12 -
2.2.3.	Perfiles de usuario del proyecto.....	- 12 -
2.2.4.	Objetivos .....	- 13 -
2.3.	DESCRIPCIÓN DEL SISTEMA.....	- 14 -
2.3.1.	Descripción.....	- 14 -
2.4.	RECURSOS .....	- 15 -
2.4.1.	Recursos humanos .....	- 15 -
2.4.2.	Recursos hardware .....	- 15 -
2.4.3.	Recursos software.....	- 16 -
2.5.	PLANIFICACIÓN DEL PROYECTO.....	- 17 -
2.5.1.	Planificación inicial .....	- 17 -
2.5.2.	Organización del proyecto .....	- 17 -
2.5.3.	Tareas .....	- 18 -
2.5.4.	Modelo de desarrollo .....	- 19 -
2.5.5.	Diagrama de Gantt .....	- 19 -
2.6.	ANÁLISIS DE COSTOS.....	- 20 -
2.6.1.	Coste de recursos humanos .....	- 20 -
2.6.2.	Coste de recursos hardware y software.....	- 21 -
2.7.	CONCLUSIONES.....	- 21 -

<b>3. ANÁLISIS DEL PROYECTO .....</b>	- 22 -
3.1. INTRODUCCIÓN.....	- 22 -
3.2. REQUERIMIENTOS FUNCIONALES.....	- 22 -
3.2.1. Visión general.....	- 23 -
3.2.2. Requisitos funcionales de usuarios .....	- 23 -
3.2.3. Diagramas de casos de usos .....	- 29 -
3.2.4. Diagramas de secuencias.....	- 32 -
3.2.5. Menús de navegación.....	- 41 -
3.3. REQUERIMIENTOS NO FUNCIONALES .....	- 41 -
3.3.1. RESTRICCIONES DE DISEÑO .....	- 41 -
3.3.2. OBJETIVOS DE DISEÑO.....	- 43 -
<b>4. DISEÑO .....</b>	- 44 -
4.1. INTRODUCCIÓN.....	- 44 -
4.1.1. Selección del entorno de desarrollo .....	- 45 -
4.1.2. Selección de bases de datos.....	- 46 -
4.2. CONFIGURACIÓN DE LA PLATAFORMA .....	- 46 -
4.3. CAPAS DE LA APLICACIÓN .....	- 47 -
4.3.1. Entorno de usuario .....	- 47 -
4.3.2. Motor de aplicación .....	- 49 -
4.3.3. Capa de datos.....	- 51 -
4.4. ESTRUCTURA DE LA BASE DE DATOS .....	- 52 -
4.5. ARQUITECTURA DE LA APLICACIÓN.....	- 57 -
4.6. INTERFASE .....	- 59 -
4.6.1. Definición de las Interfases para diferentes perfiles de usuario .....	- 59 -
4.7. ESTRUCTURA DE ARCHIVOS Y POLITICA DE BACKUP'S .....	- 65 -
<b>5. IMPLEMENTACIÓN .....</b>	- 67 -
5.1. INTRODUCCIÓN.....	- 67 -
5.2. CODIFICACION DE LAS DIFERENTES CAPAS.....	- 68 -
5.2.1. Entorno de usuario .....	- 68 -
5.2.2. Motor de la aplicación.....	- 69 -
5.2.3. Capa de datos.....	- 75 -
5.3. INTEGRACIÓN DE LAS HERRAMIENTAS DE APOYO .....	- 78 -

<b>6. PRUEBAS.....</b>	- 79 -
6.1. INTRODUCCIÓN.....	- 79 -
6.2. PRUEBAS DE COMPATIBILIDAD .....	- 79 -
6.3. PRUEBAS DE SEGURIDAD .....	- 80 -
6.4. PRUEBAS DE INTEGRACIÓN DE SERVICIO.....	- 80 -
 <b>7. CONCLUSIÓN Y RESULTADOS.....</b>	- 81 -
7.1. CONSECUCIÓN DE OBJETIVOS .....	- 81 -
7.2. DESVIACIONES OBSERVADAS .....	- 82 -
7.3. LINEAS DE AMPLIACIÓN .....	- 82 -
7.4. VALORACION PERSONAL DE LA EXPERIENCIA .....	- 83 -
 <b>BIBLIOGRAFÍA .....</b>	- 84 -
 <b>ANEXO .....</b>	- 86 -

# 1. INTRODUCCIÓN.

---

## 1.1. RESUMEN DE CONTENIDOS.

El presente proyecto consiste en el diseño e implementación de un portal WEB para una empresa de sistemas de control de iluminación.

El documento que se presenta, está dividido en siete capítulos, a los que se les añade un capítulo para la bibliografía y referencias electrónicas, y un anexo.

En este mismo capítulo, se realiza una pequeña introducción del proyecto, haciendo una breve exposición del propósito del mismo. Además se explican, de forma general, los objetivos y el alcance previsto.

En el segundo capítulo, el **Estudio de viabilidad**, se realiza un análisis del objeto del proyecto, haciendo hincapié en la situación de estudio y en los objetivos del proyecto. Además, se especifican los recursos a utilizar tanto en el desarrollo del proyecto como en su futuro funcionamiento. En este capítulo también se incluye la planificación del desarrollo y un análisis de costo del proyecto.

En el tercer capítulo se realiza en **Análisis del proyecto**, donde se especifican los requerimientos funcionales y no funcionales del mismo. Además se incluyen las restricciones de diseño que condicionen el desarrollo del proyecto.

En el cuarto capítulo se detalla el **Diseño** del proyecto. En primer lugar se describen los entornos de desarrollo y el sistema de base de datos seleccionado. También se

detallan las diferentes capas que forman la aplicación, así como la arquitectura de la misma y la estructura de la BBDD. Por ultimo se describen el diseño para las interfases de usuario.

En el quinto capítulo, se describe la **Implementación**. Por tanto, en este capítulo se describe la codificación de las diferentes capas de la aplicación, como resultado de los requisitos funcionales previstos anteriormente y el diseño realizado.

En el capitulo sexto se describen las **Pruebas** hechas una vez finalizada la implementación de la aplicación. Estas pruebas nos aseguran el correcto funcionamiento del proyecto.

En el séptimo capítulo, **Conclusión y resultados**, se explica la consecución de los objetivos. Además se proponen líneas de mejora del proyecto.

Por ultimo se presenta la **Bibliografía**, donde se exponen las fuentes de información consultadas, y el **Anexo**.

## **1.2. OBJETIVO Y ALCANCE PREVISTO**

En este documento se presenta el proyecto que consiste en el diseño e implementación de un portal WEB para una empresa que se dedica al desarrollo de aplicaciones para el diseño y control de sistemas de iluminación.

La empresa solicitante desea ampliar los servicios que oferta a sus clientes mediante una aplicación que facilite la descarga de las actualizaciones del software que la empresa desarrolla.

Además, la empresa diseña modelos de funcionamiento para dispositivos físicos, basados en el protocolo *DMX512* (Digital MultipleX). Estos modelos son de vital importancia para la expansión de la empresa ya que complementan el software de diseño de instalaciones de iluminación. Además, a día de hoy, existen pocas empresas que se dediquen al diseño de estos modelos. Por tanto, también se desea que estos modelos de funcionamiento se distribuyan a través del portal WEB, para mejorar el servicio a sus clientes.

Por otro lado, la implantación del portal WEB puede mejorar la imagen de empresa y ayudar a captar nuevos clientes, por lo que, el portal vendrá complementado con una serie de funcionalidades para la difusión de información relacionada con la empresa.

El objetivo principal del proyecto es facilitar a los clientes de la empresa el acceso a las actualizaciones de software y a los modelos de funcionamiento a través de un portal WEB. De esta manera se mejora el servicio que oferta la empresa en el momento de la solicitud del proyecto.

Además, se desea mejorar la imagen de empresa que se ofrece a los clientes y futuros clientes incluyendo información relacionada con la empresa y sus productos.

Por otro lado, se desea tener una base de datos completa con los usuarios del portal, catálogo de software, actualizaciones y modelos de funcionamiento disponibles, etc.

### **1.3. ESTADO DEL ARTE.**

El portal WEB que se presenta con este proyecto, es básicamente un portal para la descarga de actualizaciones y complementos de software. Existen diversos portales que ofrecen el mismo o parecido servicio a otras empresas. *Microsoft, Adobe, MathWorks*, son ejemplos de grandes empresas que se dedican al desarrollo de software que ofrecen descargas de actualizaciones a través de sus portales WEBS. Además de estas compañías, existen otras empresas con un predominio menor en sus respectivos mercados, que también ofertan este servicio a sus clientes. Ejemplo de estas ultimas serían *PoikoSoft* o *Emule*.

A continuación vemos un pequeño análisis de los portales de estas empresas:

- <http://www.microsoft.com>: se trata de una WEB con un gran diseño. Es posible acceder al portal en diferentes idiomas. Existe una amplia información de todos los productos de la empresa y un gestor de noticias. Hay un foro para la comunicación entre usuarios. Tiene un sistema de detección e instalación automática de actualizaciones, con un sistema claro y de fácil manejo. Es posible adquirir sus productos a través del portal.
- <http://www.adobe.com/es/>: portal con un diseño claro e intuitivo. El portal esta disponible en varios idiomas. A través del portal se puede consultar información ampliada de los productos de la empresa, complementado con un gestor de noticias. La comunidad de usuarios dispone de un foro. El usuario ha de buscar y seleccionar los archivos para la actualización del

software, para, una vez descargados, realizar la actualización. Tiene un sistema de venta *on-line*.

- <http://www.mathworks.com/>: se trata de un portal WEB con un diseño sencillo y elegante. Únicamente disponible en inglés. Contiene una amplia información de los productos. El usuario ha de buscar y seleccionar los archivos para la actualización del software, para, una vez descargados, realizar la actualización. Dispone de un sistema de adquisición de licencias.
- <http://www.emule-project.net>: es un portal con un diseño informal. Disponible en varios idiomas. El usuario ha de buscar los archivos para la descarga de actualizaciones. Contiene un gestor de noticias y un foro para la comunidad de usuarios.
- <http://www.poikosoft.com/>: se trata de un portal sencillo pero intuitivo. El usuario ha de buscar los archivos para la descarga de actualizaciones. Contiene un foro para la comunidad de usuario. Tiene un sistema de adquisición de licencias.

#### **1.4. ACERCAMIENTO A LA TECNOLOGÍA.**

La empresa que solicita el proyecto, se dedica al desarrollo de software para el diseño y control de sistemas de iluminación. Este software, trabaja con el protocolo DMX512 (**Digital MultipleX**), que permite la comunicación entre controles de iluminación y dispositivos DMX. Estos dispositivos pueden realizar funciones (movimientos, aumento/disminución intensidad, cambiar el color de la luz, etc.) de manera programada.

Uno de los servicios que suministra la empresa, es el de poder controlar un sistema de iluminación de manera automática y que dependa exclusivamente de parámetros dados por el sonido de la sala donde se encuentre el sistema. Para el funcionamiento de este software es necesario el uso de modelos de funcionamiento de los dispositivos, es decir, un conjunto de instrucciones, basadas en el protocolo DMX512, que condicione el comportamiento de los dispositivos en función, en este caso de parámetros de la música como pueden ser la intensidad o la frecuencia.

## **2. ESTUDIO DE VIABILIDAD.**

---

### ***2.1. INTRODUCCIÓN AL ESTUDIO DE VIABILIDAD.***

En el presente capítulo se realiza un análisis de diferentes aspectos del proyecto, a fin de determinar la viabilidad del proyecto.

En primer lugar se describe el objeto del proyecto, haciendo hincapié en la situación de estudio, proponiendo una mejora a esta, y en los objetivos del proyecto. Seguidamente se especifican los perfiles de usuario que harán uso de la aplicación resultante del proyecto. A continuación se especifican los objetivos que se pretenden conseguir con la aplicación del proyecto. En el siguiente apartado se da una descripción del sistema a realizar, exponiendo de forma general las funcionalidades con las que contara la aplicación. En el siguiente apartado se estudian los recursos hardware, humanos y software necesarios para el desarrollo, implantación y funcionamiento del proyecto. Teniendo en cuenta estos recursos y el sistema a implantar, se describe la planificación del proyecto, ordenando y describiendo las tareas a realizar. El análisis de costos aparece a continuación, a partir del uso de recursos de la planificación del proyecto. Finalmente se exponen las conclusiones del presente estudio de viabilidad.

## **2.2. OBJETO.**

### **2.2.1. Situación actual.**

Actualmente la empresa distribuye las actualizaciones del software que desarrolla y de los modelos de funcionamiento de forma física, es decir, entregando un soporte físico al cliente. Esta distribución se hace a través del servicio de Correos o de una empresa de mensajería, lo que conlleva un gasto extra, que en ocasiones debe asumir el cliente.

Además a día de hoy la firma no dispone de una WEB que potencie la imagen de la empresa.

### **2.2.2. Propuesta de mejora.**

La mejora que se propone con este proyecto es la de diseñar un portal WEB. En este se podrá descargar las actualizaciones y los modelos de funcionamiento vía Internet. Además se añadirán diversas funcionalidades para que los futuros clientes puedan obtener información de la empresa y ampliar así la distribución de sus productos.

### **2.2.3. Perfiles de usuario del proyecto.**

La aplicación a desarrollar esta destinada a varios perfiles de usuario:

- **Perfil usuario no registrado.** Podrá acceder cualquier persona que desee obtener información de la empresa así de los servicios que esta ofrece. Además el usuario podrá darse de alta en la WEB para tener acceso a otras funcionalidades. Este perfil es de usuario con conocimientos no expertos.
- **Perfil de usuario registrado.** Accederán todos aquellos clientes de la empresa que estén registrados en la WEB y deseen acceder a la descarga de actualizaciones y modelos de funcionamiento de los que hayan obtenido licencia. Se trata de un perfil de usuario con conocimientos no expertos y por tanto las funcionalidades a las que pueda acceder deben ser de fácil manejo.
- **Perfil gestor de noticias.** Se permitirá a los usuarios leer noticias relacionadas con la empresa, con el sector a la que se dedica o de clientes con los que trabaja. Para ello se encargarán una o varias personas que gestionarán las noticias que a los usuarios puedan interesar.

- **Perfil gestor de contenidos.** Este usuario se encargara de actualizar los catálogos de **productos y servicios** de la empresa, de actualizaciones y de modelos de funcionamiento.
- **Perfil administrador.** La aplicación necesita de un administrador para las tareas de realizar copias de seguridad, mantenimiento servidor, agregar, borrar y modificar información de las cuentas de usuarios, reestablecer contraseñas, responder consultas técnicas, etc. Por tanto se trata de un usuario con conocimientos expertos de informática.

#### **2.2.4. Objetivos.**

La implantación del portal WEB que se presenta en este proyecto tiene como objetivo principal permitir a los clientes de la empresa la descarga de las actualizaciones del software que distribuye y de los modelos de funcionamiento que desarrolla, a través de Internet. Esto mejoraría los servicios que oferta la empresa hasta ahora. Este servicio debe ser cómodo y de fácil manejo para el usuario. Esto permitirá dinamizar el intercambio de información con los usuarios del producto software y, así mismo, se alcanza un objetivo no menos importante que es el de incrementar el uso de la herramienta software de la empresa, ya que esta aporta un gran valor añadido al usuario.

Por otro lado con el portal WEB se pretende tener una base de datos actualizada tanto de clientes como de productos y modelos de funcionamiento. Esta BBDD puede dar información a la empresa de cual es su situación dentro del mercado y orientar a la empresa para futuros proyectos.

Otro objetivo a largo plazo es el de establecer un sistema de consultoría para terceras empresas instaladoras de servicios para locales de ocio utilizando esta plataforma, abriendo así nuevas líneas de negocio para la empresa.

Por ultimo, con la implantación del portal se pretende mejorar la imagen de la empresa. Hasta ahora la empresa no disponía de un WEB donde los clientes o futuros clientes obtuvieran información. Además, este portal, puede facilitar que la empresa se de a conocer entre usuarios que estén interesados en adquirir productos relacionados con el diseño de instalaciones de iluminación.

## **2.3. DESCRIPCIÓN DEL SISTEMA.**

### **2.3.1. Descripción.**

La aplicación que se presentará al final del proyecto será una WEB. Esta tendrá una parte estática, donde se expondrán contenidos informativos de la empresa, y otra parte dinámica donde los usuarios registrados podrán descargar actualizaciones de software y modelos de funcionamiento de equipos de iluminación.

Para la actualización de software se accederá a un archivo, en el terminal del cliente, donde se especifica que software tiene instalado en su equipo y la versión de éste. Para la descarga de modelos de funcionamiento, se accederá a la base de datos del cliente para consultar la lista de modelos que ya posee, y se cotejaran los resultados a la lista de modelos que dispone la empresa en la BBDD del portal WEB. Al cliente se le notificarán los resultados de la consulta para que pueda actualizar su BBDD con los modelos que no disponga.

Para la funcionalidad de gestión de noticias, los usuarios gestores de noticias, accederán a la WEB y cargarán las noticias relacionadas con el sector de la empresa o que crean convenientes. Una vez cargadas las noticias en la base de datos los usuarios podrán acceder a ellas y visualizarlas el la funcionalidad de la WEB específica.

Finalmente, existirán una serie de funcionalidades que permitan la inserción, modificación o eliminación de contenidos en la base de datos de la aplicación.

### **Funcionalidades de la aplicación.**

La WEB que proponemos tendrá las siguientes funcionalidades básicas:

- **Descargar actualizaciones:** los usuarios que estén registrados y tengan la licencia correspondiente podrán descargar las actualizaciones de software que la empresa desarrolle.
- **Descargar modelos de funcionamiento:** los usuarios que estén registrados podrán actualizar su base de datos con los modelos de funcionamiento que la empresa desarrolle.
- **Gestión de contenidos:** esta funcionalidad permitirá al usuario gestor de contenidos agregar, eliminar o modificar un producto del catálogo de la empresa. Además permitirá agregar, eliminar o modificar las actualizaciones de software que estarán disponibles en el portal. Por ultimo, este usuario, podrá

agregar, eliminar o modificar los modelos de funcionamiento que se ofrecerán a tráves del WEB.

- **Consulta del catálogo de productos:** todos aquellos usuarios que accedan al portal podrán consultar el listado de productos que la empresa ha desarrollado.
- **Gestión de noticias:** el usuario gestor de noticias, mediante esta funcionalidad, podrá dar de alta, eliminar o modificar las noticias que aparezcan en el portal.
- **Consulta de noticias:** todos aquellos usuarios que accedan al portal podrán consultar las noticias que la empresa considere oportuno incluir en el portal.
- **Sistema control de acceso:** debido a que al portal accederán diferentes perfiles de usuario, con diferentes funcionalidades, es necesario aplicar un sistema de control de acceso que permita a los usuarios acceder a aquellas funcionalidades que les esta permitido. Además este sistema de control de acceso estará complementado con un sistema de menús que permita a los usuarios acceder a dichas funcionalidades.

## **2.4. RECURSOS.**

Para la realización del proyecto, se necesitará una serie de recursos software, hardware y humanos que se detallan a continuación.

### **2.4.1. Recursos humanos.**

Para la realización de este proyecto, en el apartado de recursos humanos, tan solo se contará con la aportación del alumno, que deberá realizar las tareas de analista, técnico programador y técnico de sistemas y con la participación del profesor director del proyecto que ofrecerá soporte al alumno.

### **2.4.2. Recursos hardware.**

Este proyecto se trata de la implementación de un portal WEB, por lo que se necesita de un servidor para el hospedaje del portal. Además es necesario un ordenador personal para el acceso al WEB.

#### **2.4.2.1. Recursos mínimos del cliente.**

Para el acceso a la WEB el cliente deberá hacerlo a través de su ordenador personal. Éste deberá contar con una conexión a Internet igual o superior a 56 Kbps, además de un sistema de video compatible con resoluciones de 800x600 píxeles.

#### **2.4.2.2. Recursos mínimos del servidor.**

Para el hospedaje del portal, se hará uso de un servidor WEB. Los recursos mínimos de este servidor, dependerán del volumen de visitas que la empresa tenga previstas para el portal WEB. Otro factor que condiciona estos recursos es el tamaño de las archivos de actualización del software de la empresa.

### **2.4.3. Recursos software.**

Tanto para el diseño y desarrollo del proyecto, como para su futuro funcionamiento, se requiere de un conjunto de recursos software que se describen a continuación. El uso de estos recursos esta justificado de manera más extensa en el apartado **4.1.1 Selección del entorno de desarrollo**.

#### **2.4.3.1. PC cliente.**

Es necesario que el cliente cuente con un Sistema Operativo basado en *Windows XP* o posterior. Además deberá acceder a la WEB mediante *Internet Explorer* o *Mozilla Firefox*, ya que el proyecto esta pensado para estos navegadores.

#### **2.4.3.2. Servidor.**

El servidor que se utilice para hospedar la WEB deberá soportar bases de datos diseñadas bajo MySQL y contar con funcionalidades que permitan el acceso a BBDD de *Microsoft Access*. Además se recomienda que funcione bajo un sistema operativo basado en Windows y una plataforma de Internet Apache.

#### **2.4.3.3. Entornos de programación.**

Para la implementación del proyecto se han seleccionado los siguientes entornos de programación:

- HTML (**H**yper**T**ext **M**arkup **L**anguage) y CSS (**C**ascading **S**tyle **S**heets) para el diseño de la aplicación.

- PHP (**PHP Hypertext Pre-processor**) y JavaScript para el motor de la aplicación.
- MySQL, para el sistema gestor de la base de datos.

Además se recurrirá al siguiente software para el apoyo en diferentes necesidades:

- WinSCP, para la transferencia de archivos con el servidor.
- Visual Paradigm UML, para el diseño de diagramas de casos de uso y diagramas de secuencia.
- Microsoft Office: Word, Project y Access para la documentación.
- Server2Go, para la simulación de un servidor.

## **2.5. PLANIFICACIÓN DEL PROYECTO.**

### **2.5.1. Planificación inicial.**

El proyecto empezará el 23 de febrero del 2009 y se prevé que acabará el 17 de agosto del 2009.

Durante este periodo se dedicará una media de 3 horas diarias durante 4 días a la semana. El uso de estos días se hará de manera flexible, de manera que no influya en el rendimiento de otras actividades.

### **2.5.2. Organización del proyecto.**

El proyecto se desarrollará siguiendo las siguientes etapas:

- 1. Capturar requerimientos de usuario:** en esta etapa se realizarán las entrevistas con el cliente con el fin de **establecer** los objetivos del proyecto y toda la información necesaria para la realización del mismo.
- 2. Establecer requisitos funcionales y no funcionales del sistema:** una vez obtenida toda la información del cliente, se redactarán los requisitos del proyecto a fin de identificar todas las funcionalidades y restricciones.
- 3. Proponer una solución:** tras especificar los requisitos del proyecto y estudiarlos se propone una solución.
- 4. Estudiar alternativas:** en el caso que existan alternativas a la solución propuesta se comparan estas para escoger la que mejor se adapte al problema.

- 5. Buscar y definir los recursos software y hardware para desarrollar el proyecto:** en esta parte del proyecto se seleccionaran los recursos que sean necesarios para la realización del proyecto en los plazos deseados.
- 6. Planificar las tareas a realizar que requiere el proyecto:** en esta etapa se realizara toda la planificación del proyecto, indicando las tareas y distribuyendo los recursos que se dispongan a dichas tareas.
- 7. Realizar el proyecto:** una vez realizada la planificación se procederá a la implementación del portal, así como todos los procesos para que éste funcione.

### 2.5.3. Tareas.

Tarea	Trabajo(h)
1- Diagnóstico	16
2- Redactado documento especificación y solución propuesta	6
3- Estudio de viabilidad	10
4- Diseño	60 <sup>1</sup>
4.1- Diseño Base de datos	12
4.2- Diseño interfaces	16
4.3- Diseño módulos de funcionalidades	32
5- Programación e implementación	121 <sup>1</sup>
5.1- Programación e implementación BBDD	8
5.2- Programación interficie	16
5.3- Programación módulo adquisición de datos	8
5.4- Programación módulo actualización software	32
5.5- Programación módulo descarga de modelos	42
5.6- Programación módulo gestor de noticias	15
6- Implantación del servicio	8
7- Pruebas finales	20
8- Elaboración de documentación	60

<sup>1</sup> Trabajo acumulado de las subtareas.

### 2.5.4. Modelo de desarrollo.

El modelo de desarrollo que se ha elegido para la ejecución del proyecto es el modelo lineal. Se ha optado por este modelo ya que únicamente se dispone de un recurso humano para la elaboración del diseño e implementación del proyecto y por tanto el trabajo simultáneo de varias tareas se hace poco viable y práctico. Además la realización de las etapas se puede realizar por separado sin que esto afecte al desarrollo de las demás etapas.

### 2.5.5. Diagrama de Gantt.

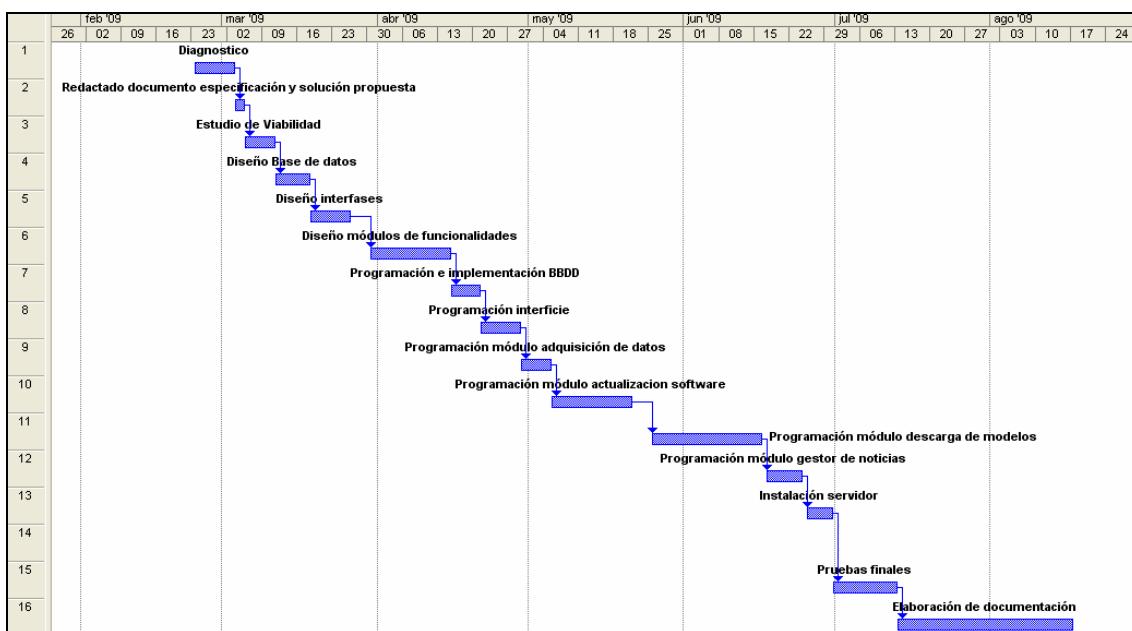


Fig. 1 Diagrama de Gantt.

## 2.6. ANÁLISIS DE COSTOS.

### 2.6.1. Coste de recursos humanos.

Como especifica el apartado **3.2.3 Recursos Humanos**, para la realización del proyecto únicamente se contará con el alumno y el profesor director del proyecto. A estos se le han asignado un coste de 10€/h y 30€/h respectivamente.

Tarea	Trabajo(h)	Costo
Diagnóstico	16	160,00€
Redactado documento especificación y solución propuesta	6	60,00€
Estudio de viabilidad	10	100,00€
Diseño	60 <sup>2</sup>	600 <sup>2</sup>
Diseño Base de datos	12	120,00€
Diseño interfaces	16	160,00€
Diseño módulos de funcionalidades	32	320,00€
Programación e implementación	121 <sup>2</sup>	1210 <sup>2</sup>
Programación e implementación BBDD	8	80,00€
Programación interficie	16	160,00€
Programación módulo adquisición de datos	8	80,00€
Programación módulo actualización software	32	320,00€
Programación módulo descarga de modelos	42	420,00€
Programación módulo gestor de noticias	15	150,00€
Instalación servidor	8	80,00€
Pruebas finales	20	20,00€
Elaboración de documentación	60	60,00€
<b>Total</b>	<b>301</b>	<b>3010,00€</b>

<sup>2</sup> Trabajo y costo acumulado de las subtareas.

## 2.6.2. Coste de recursos hardware y software.

En la siguiente tabla se presenta la relación de costos de material.

Recurso	Costo Total	Costo Mensual
Alquiler Servidor		96€ <sup>3</sup> /12meses <sup>3</sup>
Conexión Internet 20Mb		29,95 €
PHP 5.1.0	0,00 €	
MySQL 5,0	0,00 €	
Amortización PC programador	200,00 € <sup>4</sup>	
Amortización MSPROJECT analista	60,00 € <sup>4</sup>	
Amortización MSOffice 2003	40,0 € <sup>4</sup>	
Total	300,00 €	37,95 €

## 2.7. CONCLUSIONES.

Una vez realizado el estudio de viabilidad, se puede extraer las siguientes conclusiones:

- El proyecto presentará beneficios a los clientes de la empresa ya que facilitara el acceso a las actualizaciones de software y a los modelos de funcionamiento.
- Se reducirán los costes asociados al almacenaje, gestión y envío de soportes físicos en formato digital (CD) y al envío a los clientes de las actualizaciones de software y de modelos de funcionamiento.
- Permitirá a los clientes y futuros clientes obtener información de la empresa y de las actividades que realiza.
- La empresa dispondrá de una base de datos con los clientes que estén registrados.

A todas estas conclusiones, y valorando los apartados de este estudio de viabilidad, debemos añadir que el proyecto es viable.

<sup>3</sup> Coste del alquiler del servidor en [www.solo10.com](http://www.solo10.com). Especificaciones en el anexo. Pág. 86.

<sup>4</sup> Costo = (precio compra / meses para amortización) · meses que se utilizará en el proyecto.

## **3. ANÁLISIS DEL PROYECTO.**

---

### ***3.1. INTRODUCCIÓN.***

En el presente capítulo se describen los requerimientos funcionales de los diferentes perfiles de usuario que tiene el proyecto.

Como complemento a la descripción de los requerimientos funcionales se incluyen una serie de diagramas de casos de uso para cada perfil de usuario, además de diagramas de secuencia para las funcionalidades mas destacadas.

A continuación se detallan los requerimientos no funcionales del proyecto, teniendo en cuenta las restricciones asociadas al diseño como los objetivos de diseño.

### ***3.2. REQUERIMIENTOS FUNCIONALES.***

Los requerimientos funcionales hacen referencia al comportamiento de la aplicación, con todas aquellas funcionalidades que la integran. Estos requerimientos funcionales están representados en los diagramas de uso para cada tipo de usuario presente en el proyecto. Además, se presentan los diagramas de secuencia de cada uno de estos requisitos funcionales.

### **3.2.1. Visión general.**

El proyecto que se presenta en este documento es la implementación de un portal WEB para una empresa que se dedica al desarrollo de software relacionado con el diseño y control de sistemas de iluminación. De forma complementaria, la empresa se dedica al diseño de modelos de funcionamiento, basados en el protocolo *DMX512*, de los dispositivos físicos de dichos sistemas de iluminación.

El WEB que se obtendrá como resultado de este proyecto tiene como objeto principal el facilitar el acceso a las actualizaciones del software que la empresa desarrolle y de los modelos de funcionamiento. Las actualizaciones y los modelos de funcionamiento únicamente estarán disponibles para los clientes de la empresa que dispongan de la correspondiente licencia.

El portal necesitará de un usuario gestor de contenidos, que se encargue de actualizar la base de datos del WEB. Esta gestión consistirá básicamente en actualizar los catálogos de productos de la empresa, de las actualizaciones disponibles y de los modelos de funcionamiento.

Por otro lado, el portal servirá para proporcionar información a los clientes y futuros clientes sobre la empresa. Esta información estará dividida en dos partes: por un lado la información estática (ubicación de la empresa, contacto, personal, etc.), y por otro lado la información dinámica, referente a noticias relacionadas con la empresa, con el sector al que se dedica la empresa o que se crean de interés. Estas noticias se agregarán a la BBDD del WEB y por tanto se precisará de un usuario gestor de noticias para administrar estos contenidos.

Finalmente existirá la figura del usuario administrador, que se encargará de la gestión de perfiles y derechos de usuarios. Esta gestión permitirá cambiar las funcionalidades asignadas a cada perfil de forma que a través de la BBDD podremos modificar las capacidades de los diferentes perfiles dinámicamente, y de asociar las diferentes funcionalidades con los perfiles de usuario.

### **3.2.2. Requisitos funcionales de usuarios.**

Como se explica en el apartado anterior, la aplicación dispondrá inicialmente de 5 perfiles de usuario distinto (usuario no registrado, usuario registrado, usuario gestor de noticias, usuario gestor de contenidos y administrador), cada uno con requisitos funcionales específicos.

Así mismo, existirá un nivel de privilegios: el primer nivel formado por el usuario no registrado, un segundo nivel en el que estarán incluidos el usuario registrado, el usuario gestor de noticias y el usuario gestor de contenidos y finalmente el tercer nivel formado por el administrador de la aplicación. Los usuarios de cada nivel tendrán asignadas sus funcionalidades pero además heredarán las de los niveles inferiores. De este modo, los usuarios del tercer nivel podrán acceder a sus funcionalidades y además a las funcionalidades de los perfiles del segundo y primer nivel de privilegios. Cabe destacar que en el segundo nivel se encuentran tres perfiles de usuario diferentes, y por tanto estos únicamente podrán acceder a sus funcionalidades específicas más las funcionalidades del primer nivel de privilegio. En la figura 3.1 se muestra la relación entre niveles.

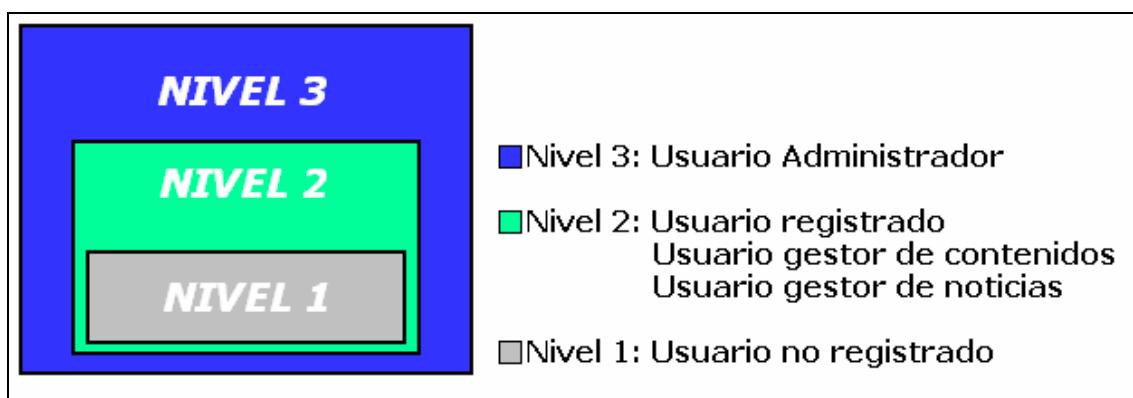


Fig. 3.1: Relación entre niveles de privilegios

A continuación se presentan los requisitos de funcionamiento específicos de cada perfil de usuario definido en el proyecto.

### 3.2.2.1. Usuario no registrado.

El usuario no registrado o visitante ocasional, dispondrá de los siguientes requisitos funcionales:

- **Acceso al contenido público.**

Los usuarios no registrados podrán entrar al portal WEB para acceder a todos aquellos módulos que no necesiten autenticación a fin de consultar información referente a la empresa, ponerse en contacto con el servicio de atención al cliente vía correo electrónico, etc.

- **Consultar Catálogo de productos.**

La aplicación WEB debe permitir a los usuarios no registrados acceder al catálogo de productos software que oferta la empresa.

- **Consultar noticias.**

El portal WEB permitirá a los usuarios no registrados acceder al modulo de consulta de noticias relacionadas con el ámbito en el que trabaja la empresa que el gestor de noticias crea conveniente añadir a la base de datos de la aplicación.

- **Registrarse.**

Los usuarios no registrados podrán registrarse en la WEB, mediante el modulo correspondiente. Para ello deberán llenar un formulario con los datos necesarios y aceptar los términos legales sobre difusión y gestión de datos.

- **Autentificarse.**

Todos aquellos usuarios que hayan realizado su registro en la base de datos del portal, deberán hacer una autenticación (o log-in), para poder acceder al contenido protegido, y a las funcionalidades específicas al perfil de usuario asignado a su registro.

### **3.2.2.2. Usuario registrado.**

El usuario registrado heredará las funcionalidades del usuario no registrado, y además dispondrá de los siguientes requisitos funcionales:

- **Comprobar versiones de software.**

La aplicación tendrá un modulo para que los usuarios puedan comprobar si existe una versión más reciente de aquel software que tengan instalado en su ordenador personal.

- **Descargar actualizaciones de software.**

Los usuarios autenticados podrán descargar las actualizaciones de software del que dispongan las licencias, mediante el módulo correspondiente.

- **Comprobar modelos de funcionamiento disponibles.**

La aplicación tendrá una funcionalidad para que los usuarios puedan comprobar la lista de modelos de funcionamientos de que dispone la empresa.

- **Actualizar modelos de funcionamiento.**

Una vez comprobado los modelos de funcionamiento de que dispone la empresa el usuario podrá actualizar su base de datos con aquellos modelos que no disponga, permitiéndole que haga una selección.

### **3.2.2.3. Usuario gestor de noticias.**

El usuario gestor de noticias heredará las funcionalidades del usuario no registrado, y además dispondrá de los siguientes requisitos funcionales:

- **Agregar noticias.**

Los usuarios gestores de noticias, una vez autentificados, podrán agregar noticias, mediante el correspondiente módulo, para que los demás usuarios puedan consultarlas.

- **Editar Noticias**

Los usuarios gestores de noticias, podrán editar las noticias ya añadidas a la base de datos. Esta edición consistirá en cambiar el contenido de la noticia en el caso que sea necesario rectificar parte de la misma o añadir información. No se podrá editar el autor ni la fecha de publicación de la noticia.

- **Eliminar noticias.**

Todas aquellas noticias que el gestor de noticias crea conveniente eliminar podrán ser eliminadas por é.

### **3.2.2.4. Usuario gestor de contenidos.**

El usuario gestor de contenidos heredará las funcionalidades del usuario no registrado, y además dispondrá de los siguientes requisitos funcionales:

- **Gestión de catálogo.**

Una vez autenticado el usuario gestor de contenidos podrá acceder a tres funcionalidades relacionadas con la gestión del catálogo:

- **Alta de producto:** cuando la empresa desarrolle un nuevo software, el usuario gestor de contenidos podrá agregarlo al catálogo de productos software. Esta alta de producto aparecerá información que la empresa crea necesaria.
- **Baja de producto:** el usuario gestor de contenidos podrá eliminar de la base de datos aquellos productos que la empresa crea necesario.
- **Modificar producto:** En el caso que sea necesario, el usuario gestor de contenidos, podrá modificar la información referente a cualquier producto añadiendo o eliminando los contenidos que la empresa crea conveniente.

- **Gestión de Modelos de Funcionamiento.**

Una vez autenticado el usuario gestor de contenidos podrá acceder a tres funcionalidades relacionadas con la gestión de los modelos de funcionamiento:

- **Alta de modelo de funcionamiento:** el usuario gestor de contenidos será el encargado de agregar a la base de datos los modelos de funcionamiento de los que dispongan, así como la información relacionada con el mismo.
- **Baja de modelo de funcionamiento:** si es necesario, el gestor de contenidos podrá eliminar de la base de datos aquellos modelos de funcionamiento que se crean innecesarios.
- **Modificar modelos de funcionamiento:** el gestor de contenidos podrá modificar aquellos modelos de funcionamiento que deban ser modificados.

- **Gestión de actualizaciones de software.**

El usuario gestor de contenidos podrá acceder a las funcionalidades referentes a la gestión de las actualizaciones de software:

- **Alta de actualización:** el usuario gestor de contenidos será el encargado de agregar a la base de datos todas las actualizaciones de software que se desarrollen, siendo el usuario Administrador el encargado de añadir el archivo correspondiente a la actualización en el servidor.
- **Baja de actualización:** en el caso que sea necesario el usuario gestor de contenidos podrá eliminar de la base de datos aquellas actualizaciones que se crean innecesarias. El encargado de retirar el archivo, correspondiente a la actualización, del servidor será el usuario Administrador.
- **Modificar actualización:** la modificación de la información referente a cualquier actualización de software la podrá realizar el usuario gestor de contenidos. . El encargado de modificar el archivo, correspondiente a la actualización de software, del servidor será el usuario Administrador.

### **3.2.2.5. Usuario administrador.**

El usuario administrador heredará las funcionalidades disponibles para los niveles inferiores de usuario de la aplicación, y además dispondrá de los siguientes requisitos funcionales:

- **Clasificación de usuarios.**

El usuario administrador será el encargado de determinar a qué perfil de usuario pertenece cada usuario. Por defecto los usuarios serán añadidos al la base de datos como usuarios registrados.

- **Gestión de perfiles.**

A cada perfil de usuario, como ya se ha explicado, le corresponden unas determinadas funcionalidades. El administrador, en el caso que sea necesario, podrá cambiar estas asociaciones o añadir nuevas, de manera que se le puedan añadir funcionalidades a cada perfil de usuario.

- **Subir archivos/imágenes al servidor.**

Cuando un usuario gestor de contenidos añade una versión del software a la base de datos, esta versión, debe estar asociada a un archivo existente en el servidor. El administrador, es el encargado de subir el a servidor el archivo asociado a dicha versión.

Además, si un usuario gestor de noticias o contenidos, desea mostrar en una noticia o producto una imagen hospedada en el servidor, debe ser el administrador quien suba la imagen al servidor.

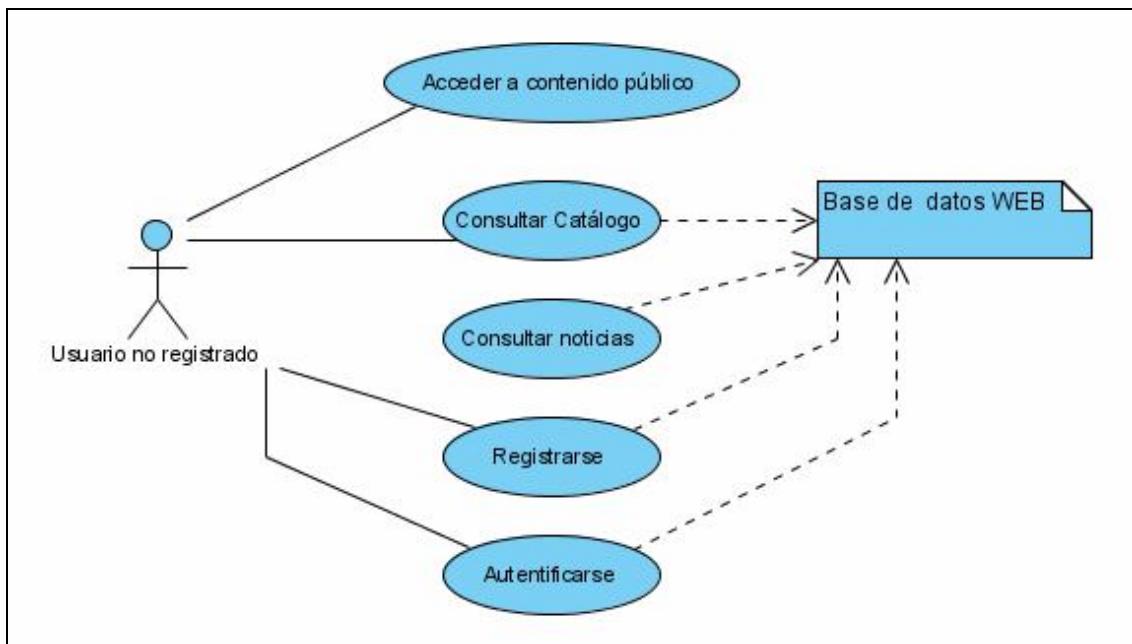
- **Gestionar BBDD.**

El administrador de la aplicación tendrá libre acceso a la base de datos. Esto le permitirá gestionar posibles incidencias que ocurran durante el funcionamiento del portal (dar de baja usuarios, eliminar contenidos, etc.).

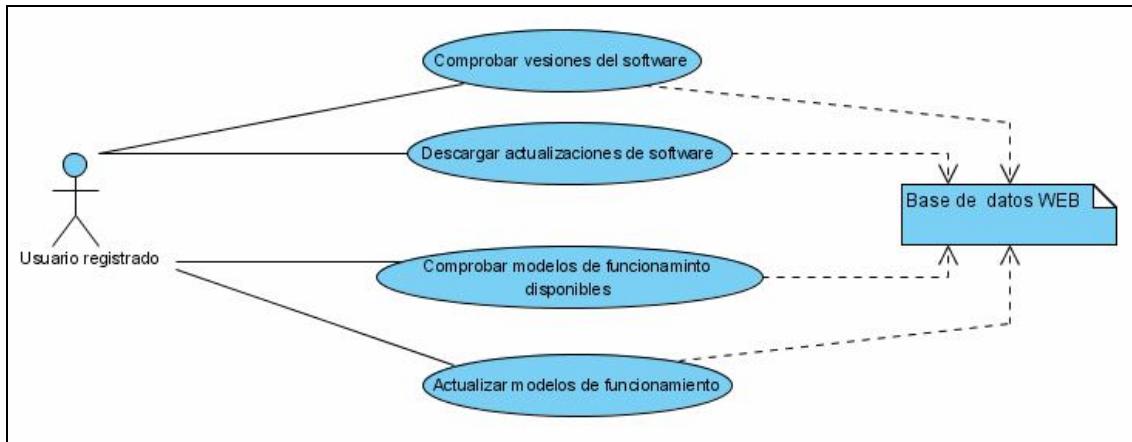
### 3.2.3. Diagramas de casos de usos.

A continuación se muestran los casos de usos de cada perfil de usuario según se han especificado en el apartado **3.1.2 Requisitos funcionales de usuarios**.

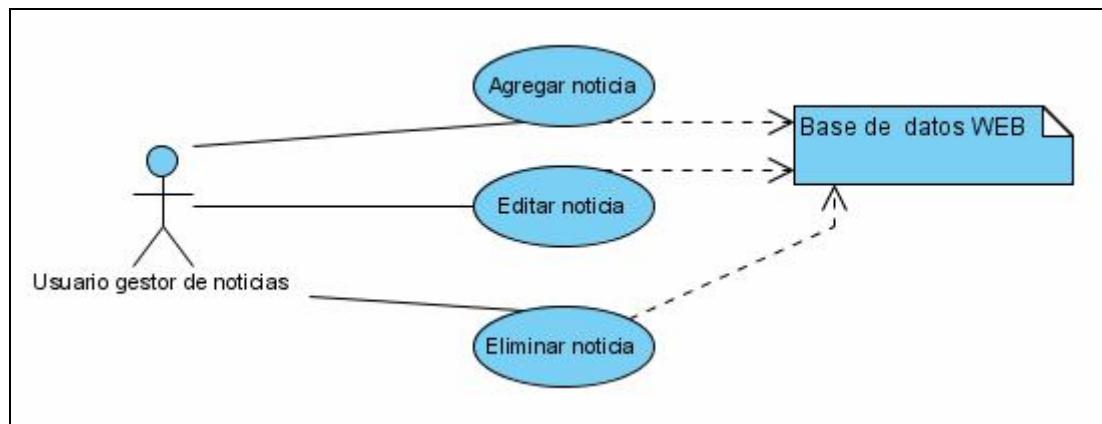
#### 3.2.3.1. Diagrama de casos de uso para usuario no registrado.



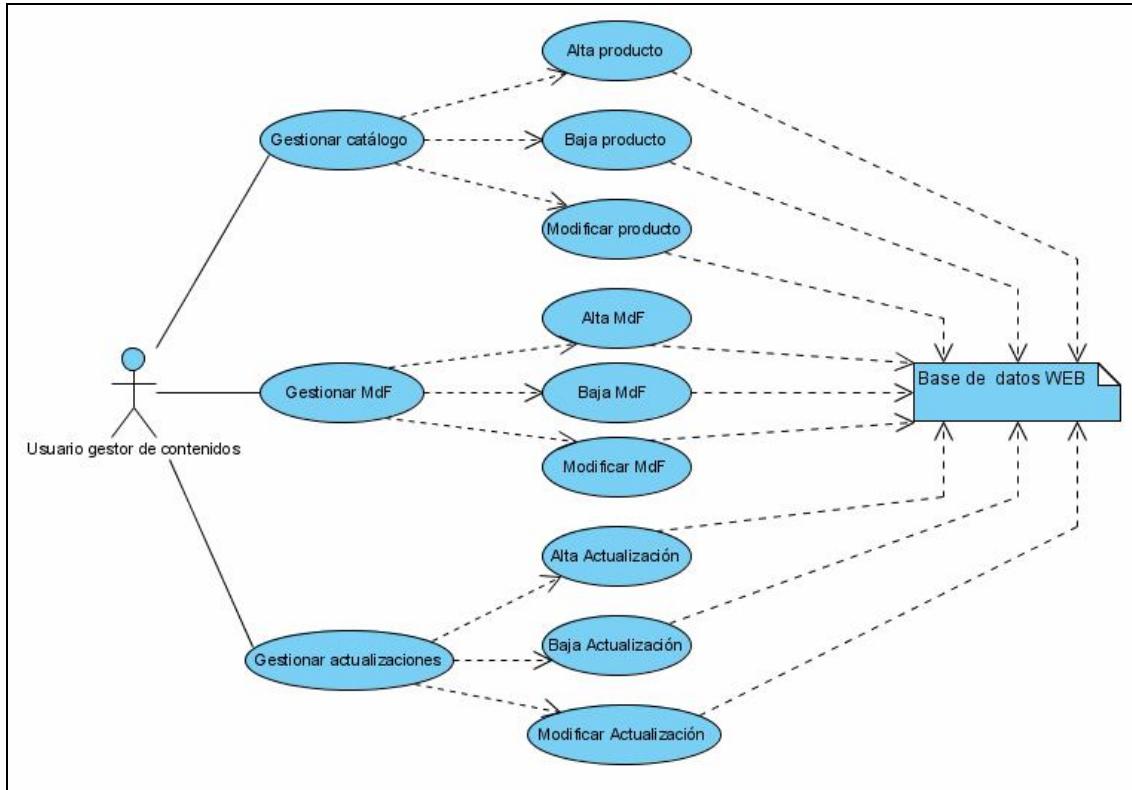
### 3.2.3.2. Diagrama de casos de uso para usuario registrado.



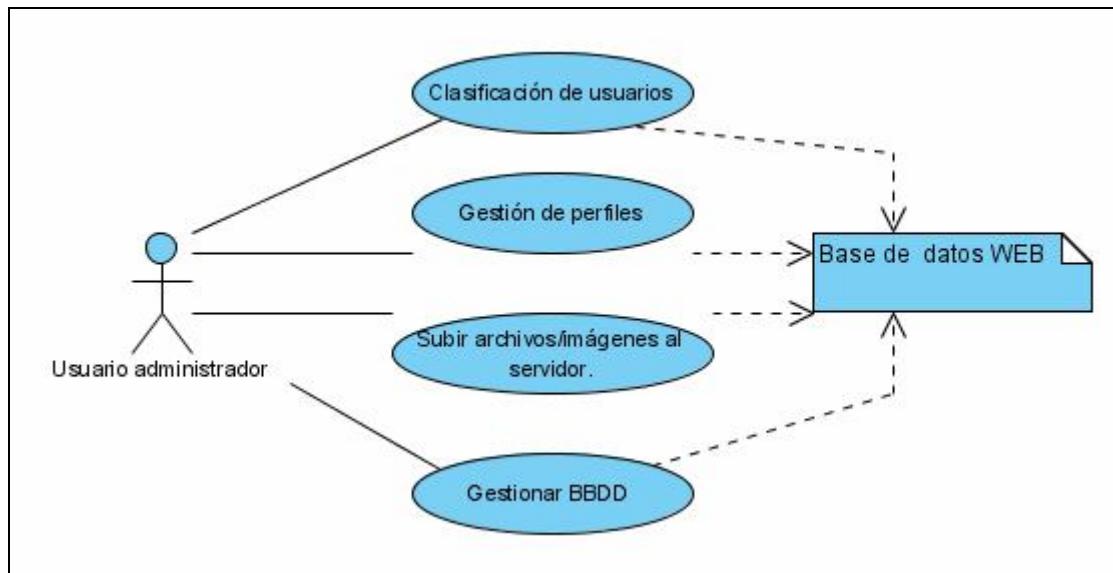
### 3.2.3.3. Diagrama de casos de uso para usuario gestor de noticias.



### 3.2.3.4. Diagrama de casos de uso para usuario gestor de contenidos.



### 3.2.3.5. Diagrama de casos de uso para usuario administrador.



### 3.2.4. Diagramas de secuencias.

A continuación se muestran los diagramas de secuencias para los requisitos funcionales más significativos de cada perfil de usuario según se han especificado en el apartado **3.1.2 Requisitos funcionales de usuarios** y en el apartado **3.1.3 Diagramas de casos de usos**.

#### 3.2.4.1. Diagramas de secuencias para usuario no registrado.

En las figuras 3.2, 3.3, 3.4 y 3.5 se muestran los diagramas de secuencias correspondientes a las funcionalidades del usuario no registrado.

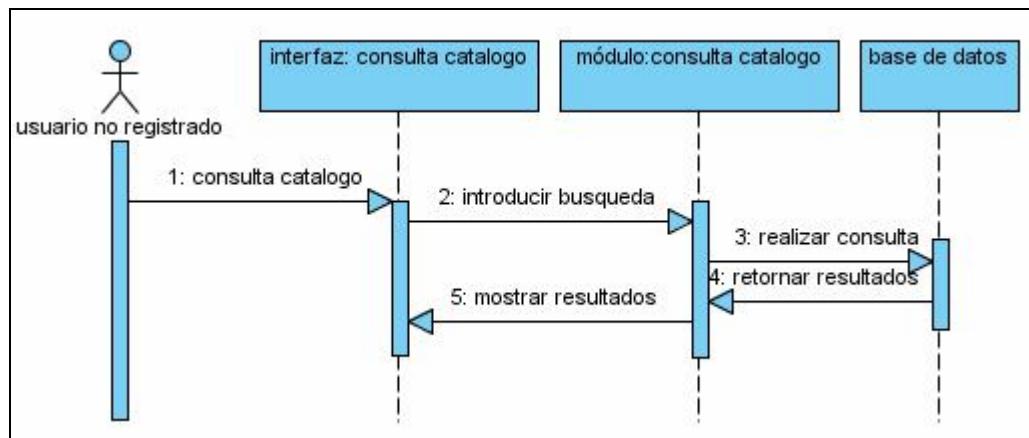


Fig. 3.2: Consultar Catálogo de productos.

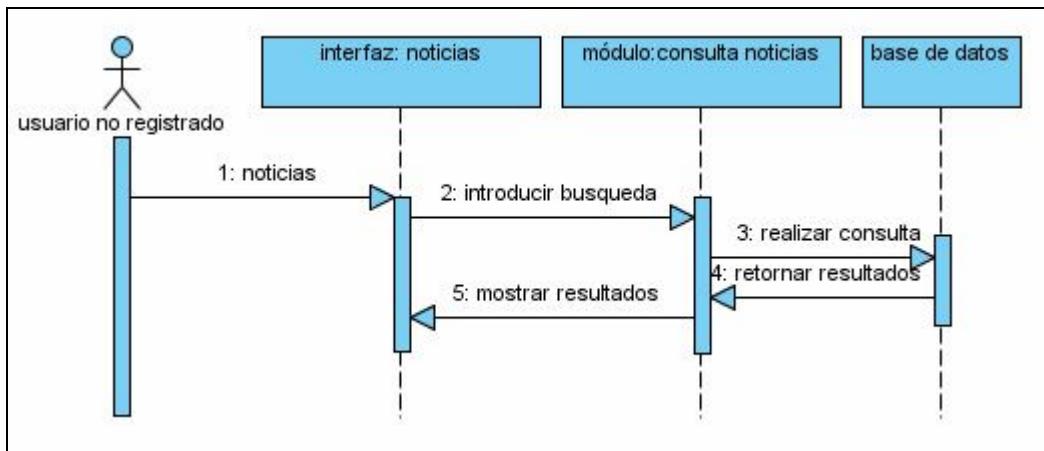


Fig.3.2: Consultar Noticias.

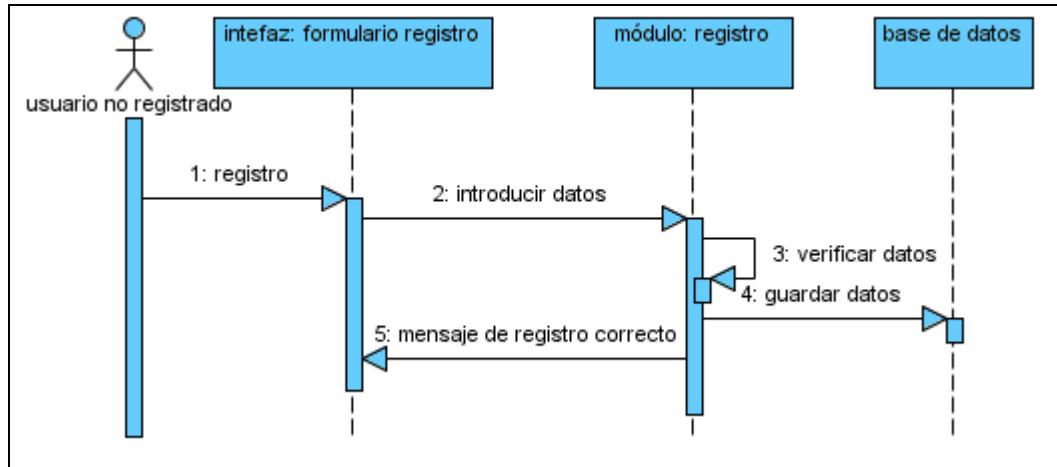


Fig.3.3: Registrarse

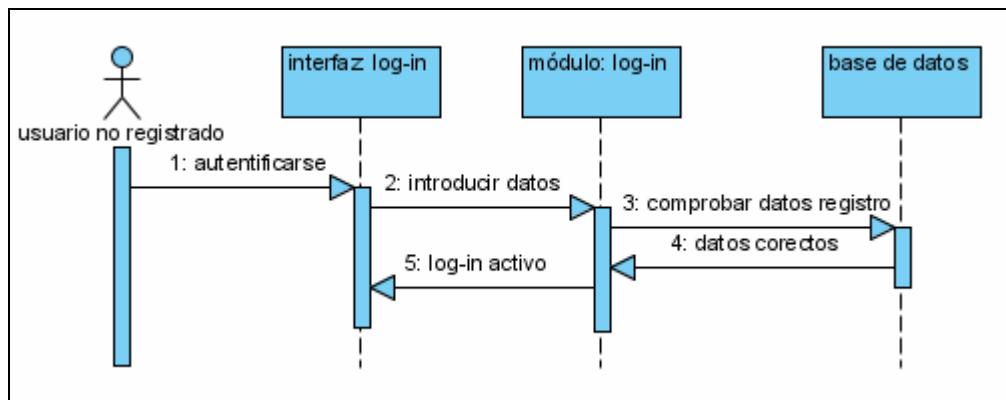


Fig. 3.4: Autentificarse

### 3.2.4.2. Diagramas de secuencias para usuario registrado.

De la figura 3.6 a la figura 3.9 se muestran los diagramas de secuencias correspondientes a las funcionalidades del usuario registrado.

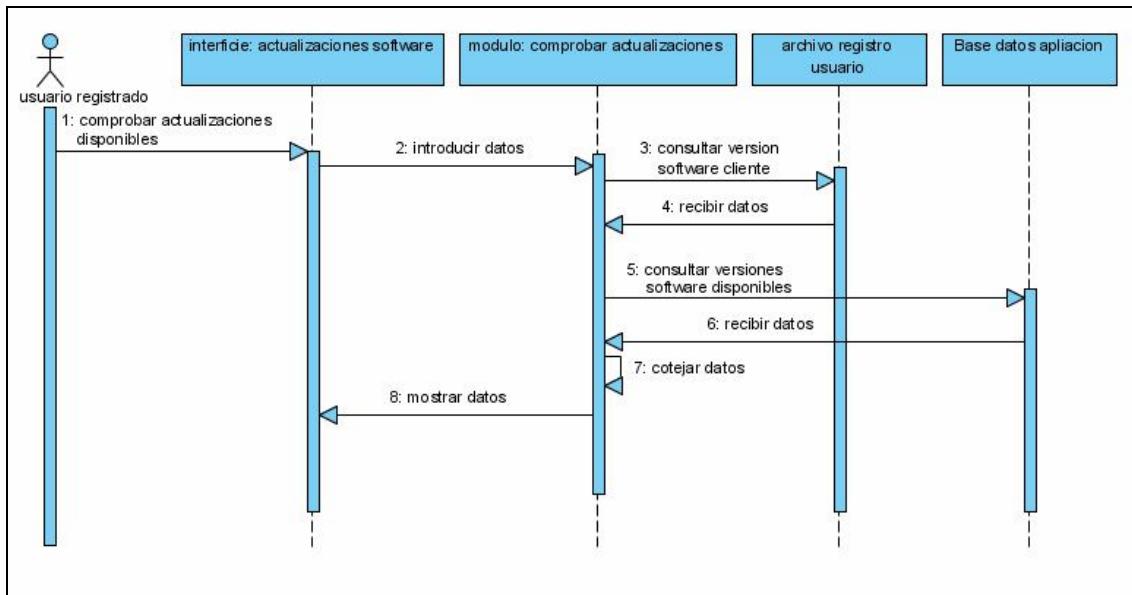
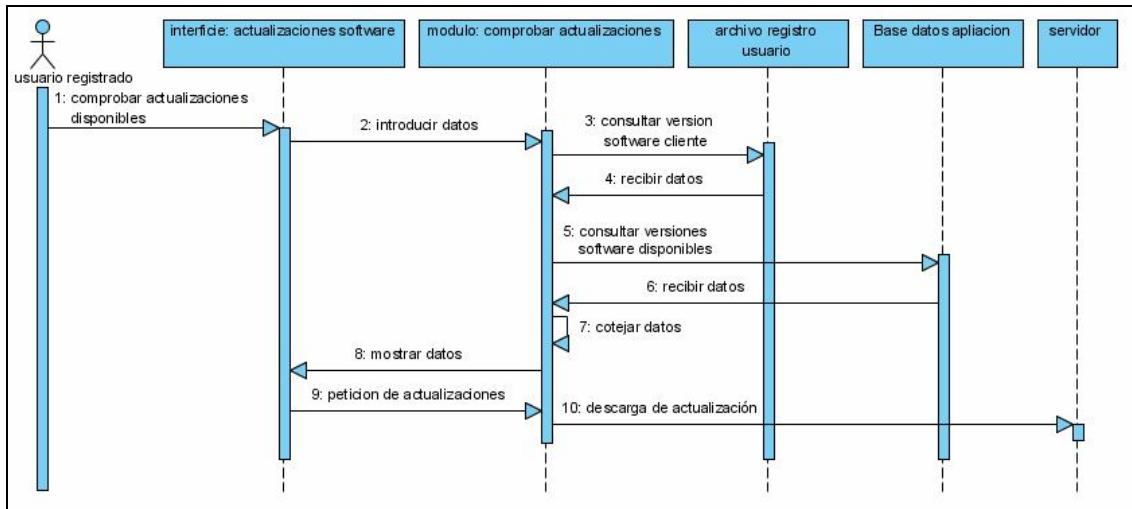
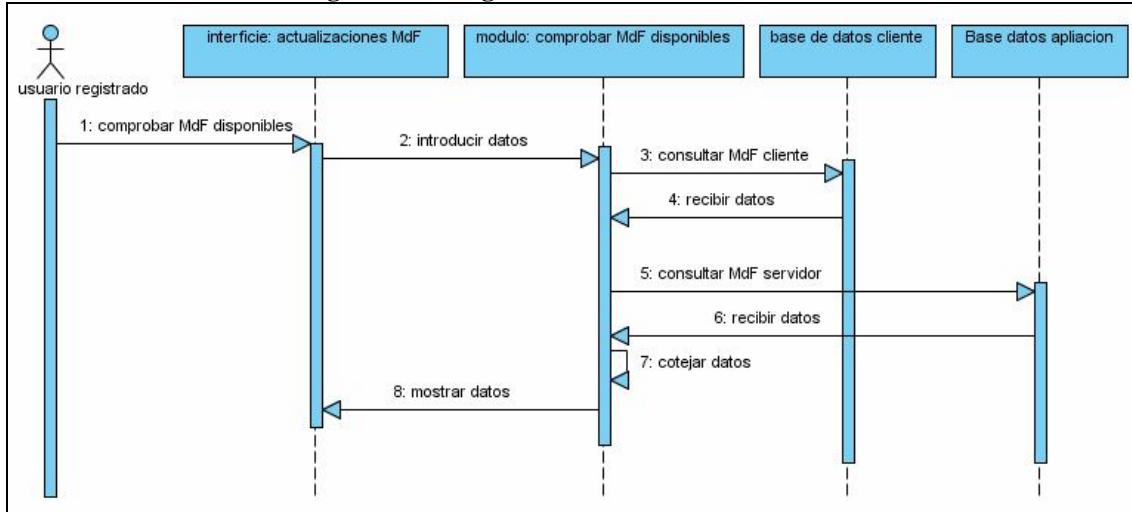
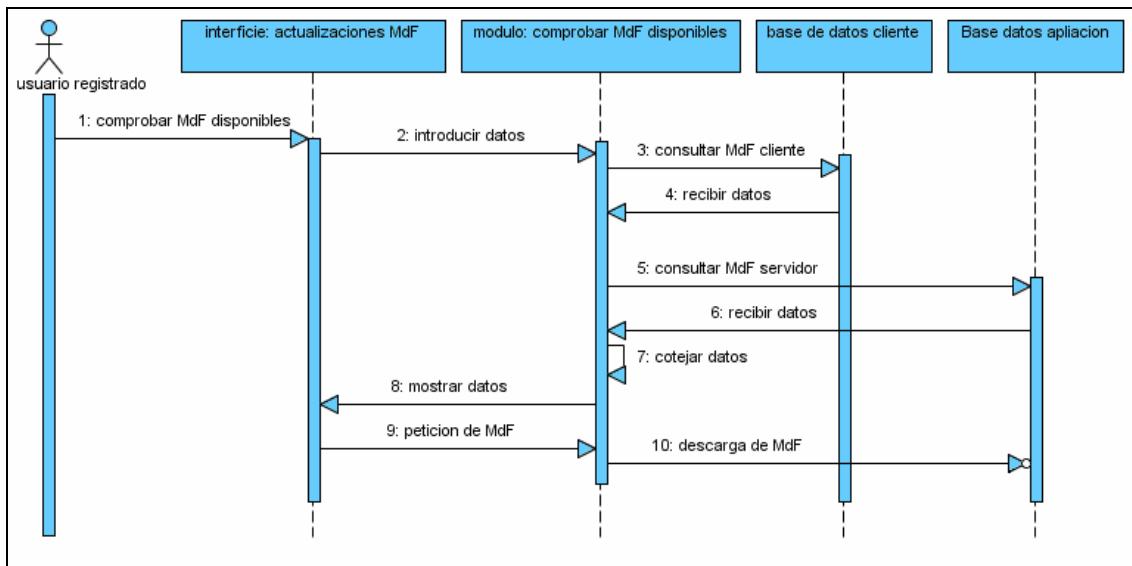


Fig. 3.6: Comprobar versiones de software.

**Fig. 3.7: Descargar actualizaciones de software.****Fig. 3.8: Comprobar modelos de funcionamiento disponibles.****Fig. 3.9: Actualizar modelos de funcionamiento.**

### 3.2.4.3. Diagramas de secuencias para usuario gestor de noticias.

De la figura 3.10 a la figura 3.12 se muestran los diagramas de secuencias correspondientes a las funcionalidades del usuario gestor de noticias.

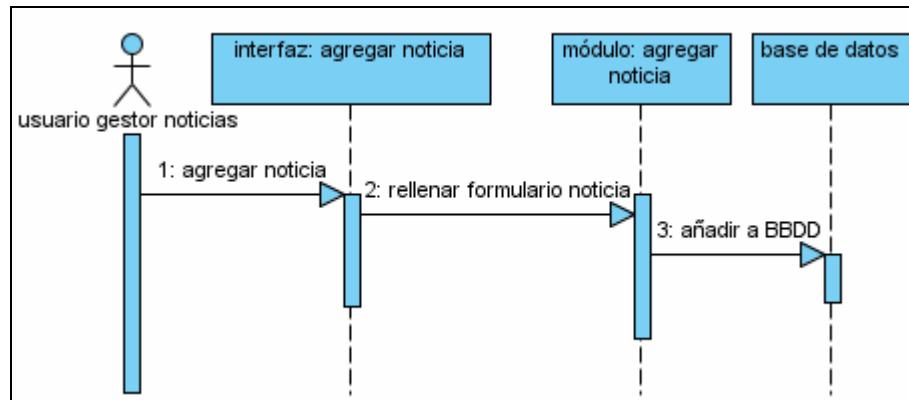


Fig. 3.10: Agregar noticia.

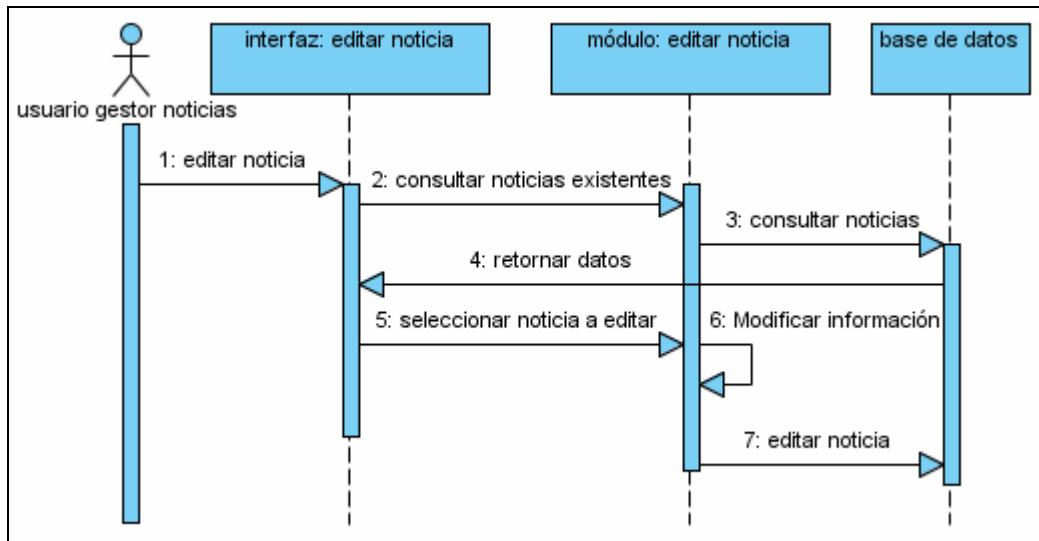


Fig. 3.11: Editar noticia.

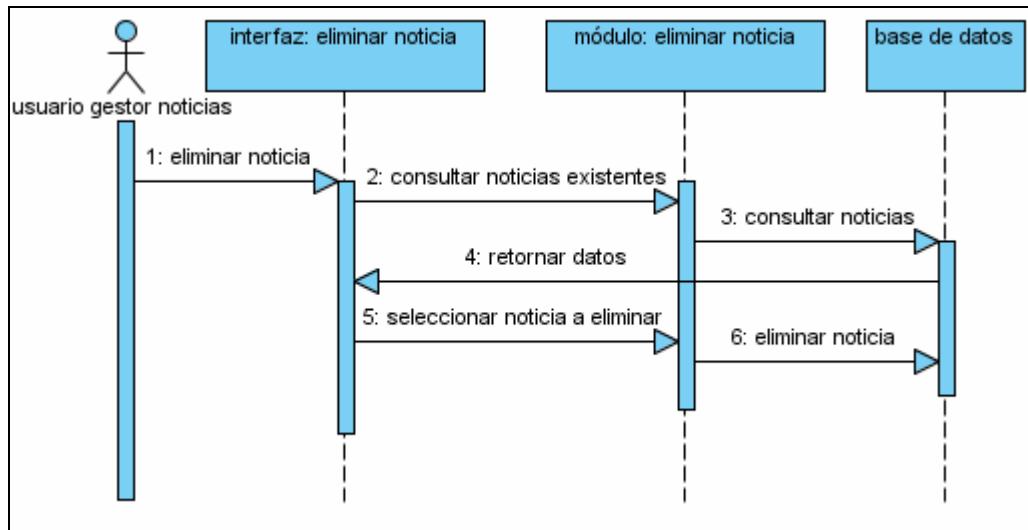


Fig. 3.12: Eliminar noticia.

### 3.2.4.4. Diagramas de secuencias para usuario gestor de contenidos.

De la figura 13 a la figura 21 se muestran los diagramas de secuencias correspondientes a las funcionalidades del usuario gestor de contenidos.

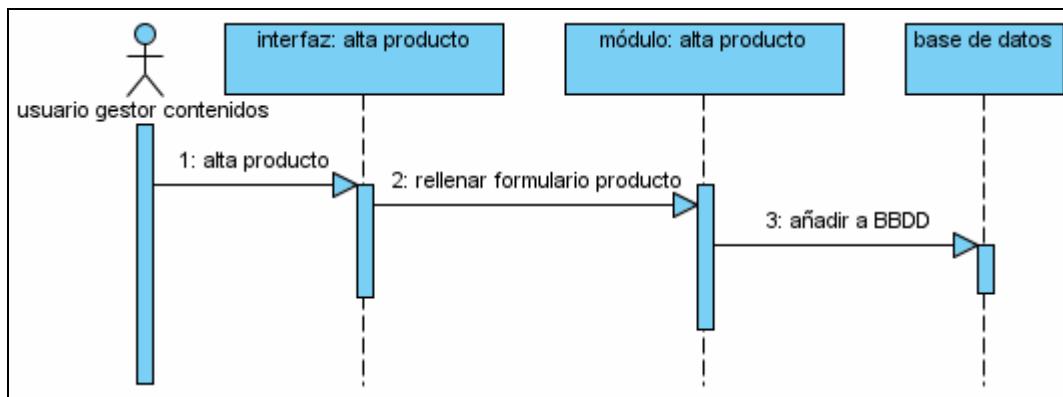


Fig. 3.13: Alta producto.

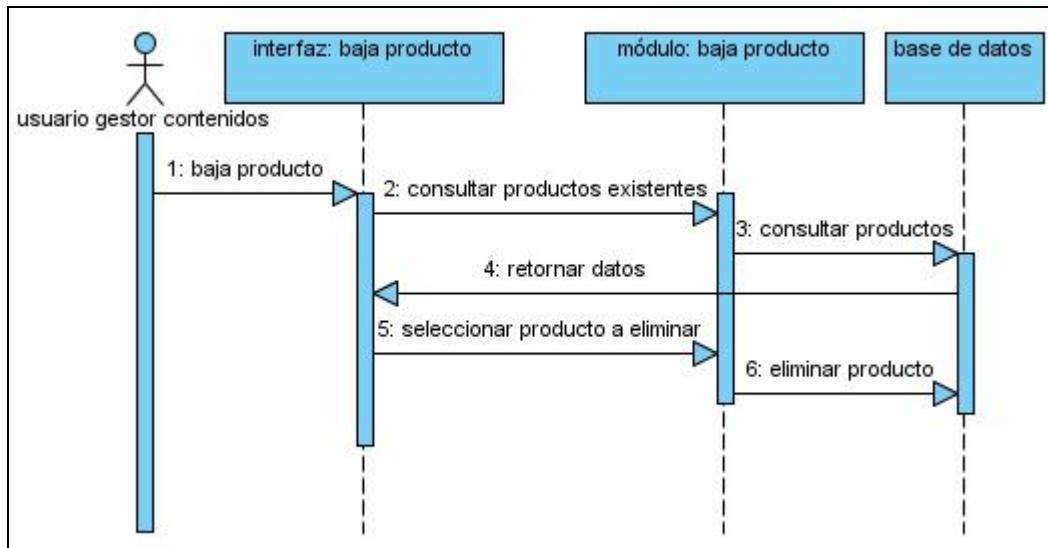


Fig. 3.14: Baja producto.

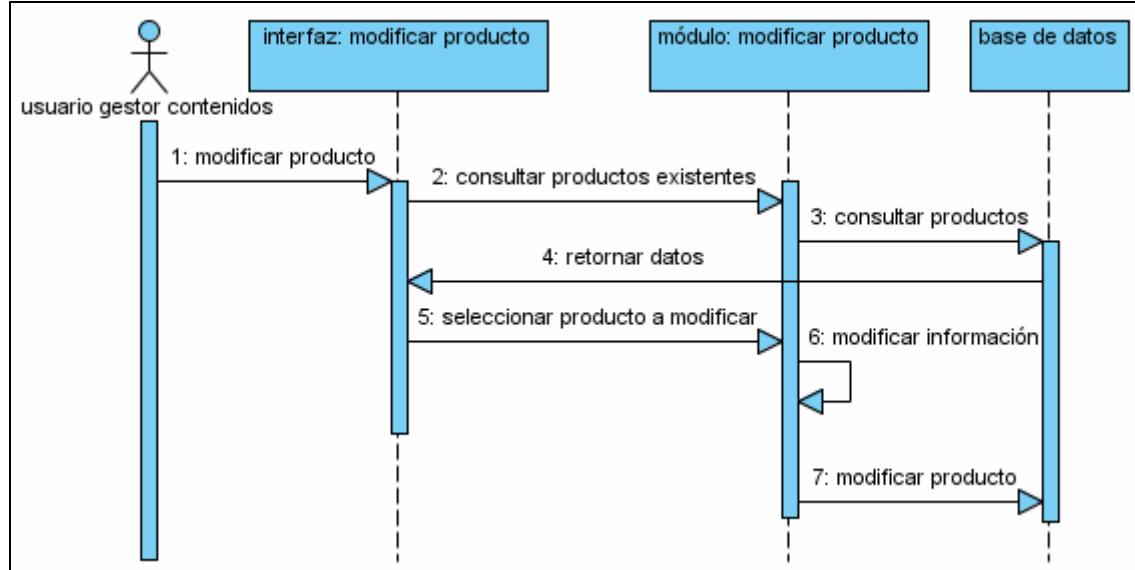


Fig. 3.15: Modificar producto.

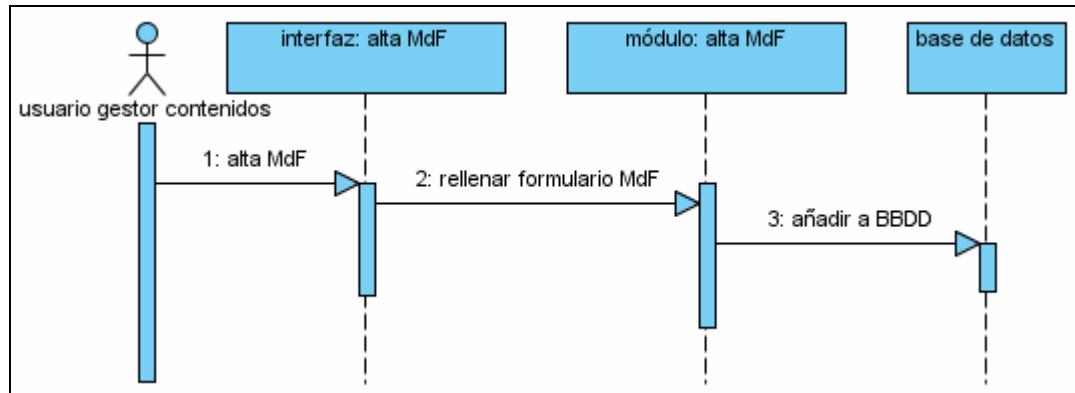
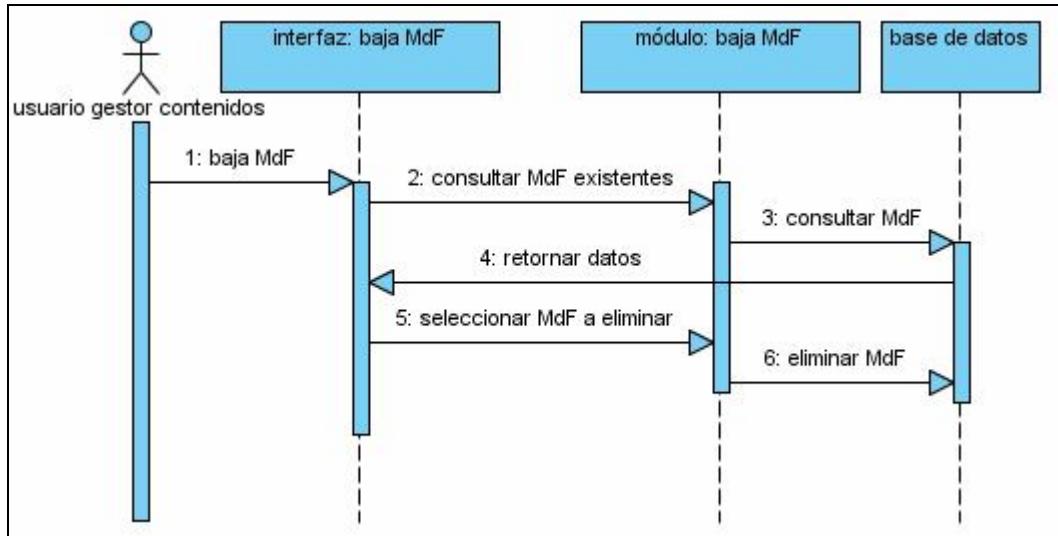
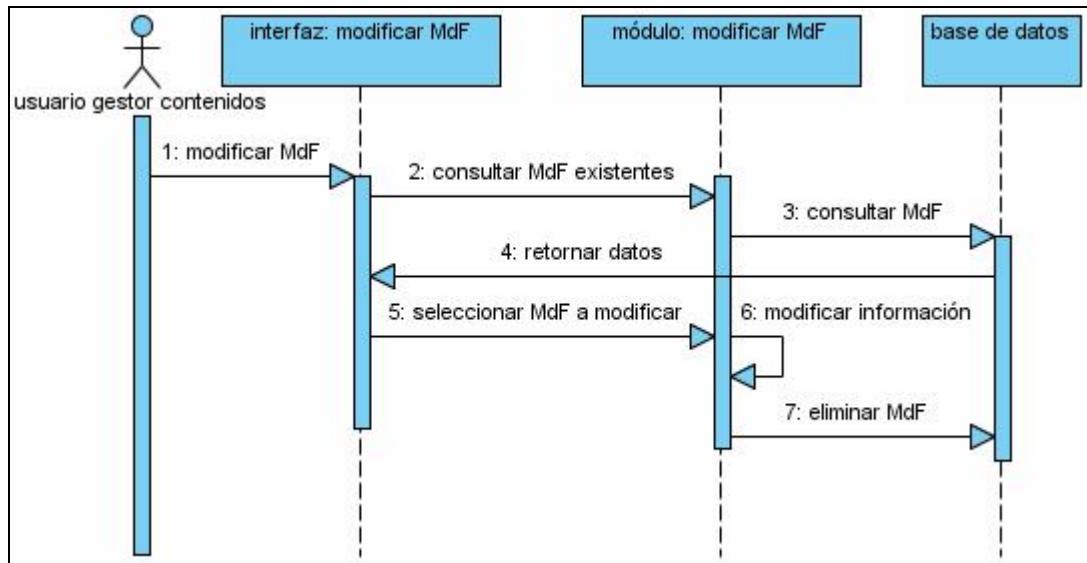
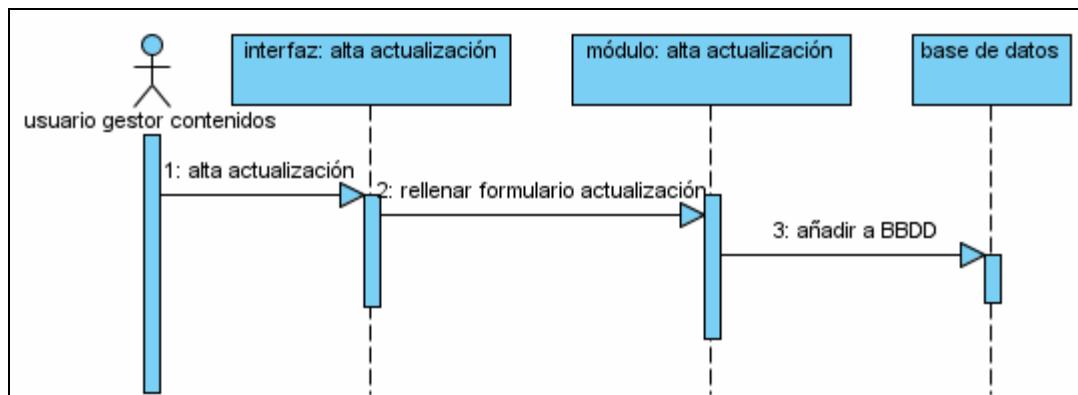


Fig. 3.16: Alta modelos de funcionamiento.

**Fig. 17: Baja modelo de funcionamiento.****Fig. 3.18: Modificar modelos de funcionamiento.****Fig. 3.19: Alta de actualización.**

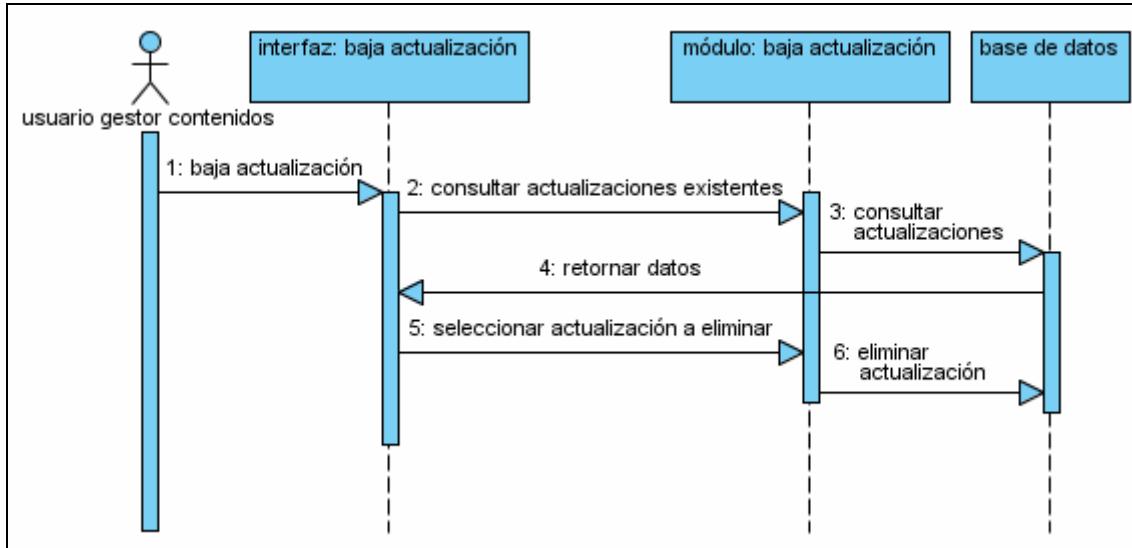


Fig. 3.20: Baja de actualización.

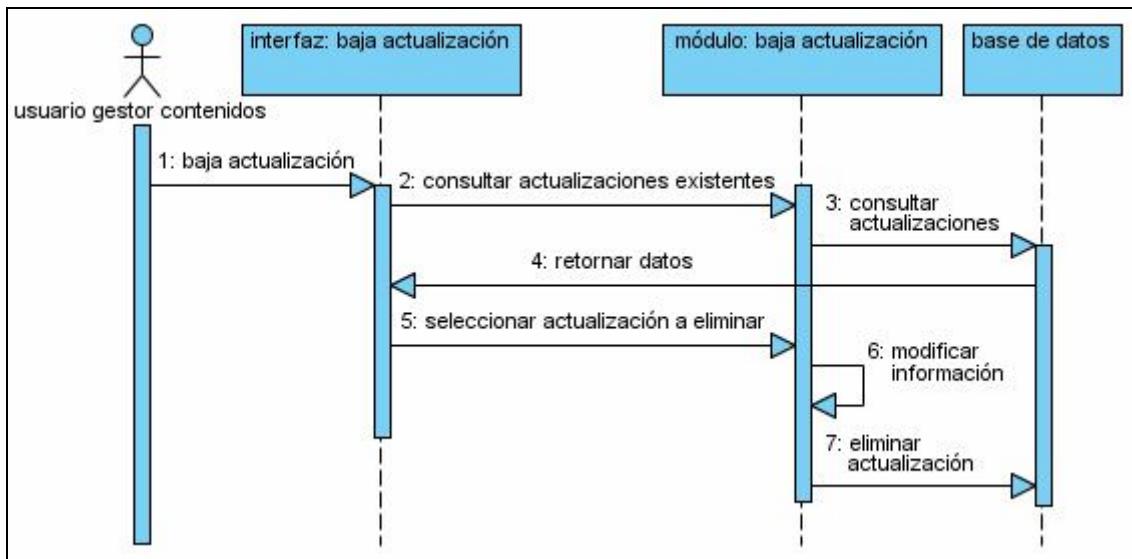


Fig. 3.21: Modificar actualización.

### 3.2.4.5. Diagramas de secuencias para usuario administrador.

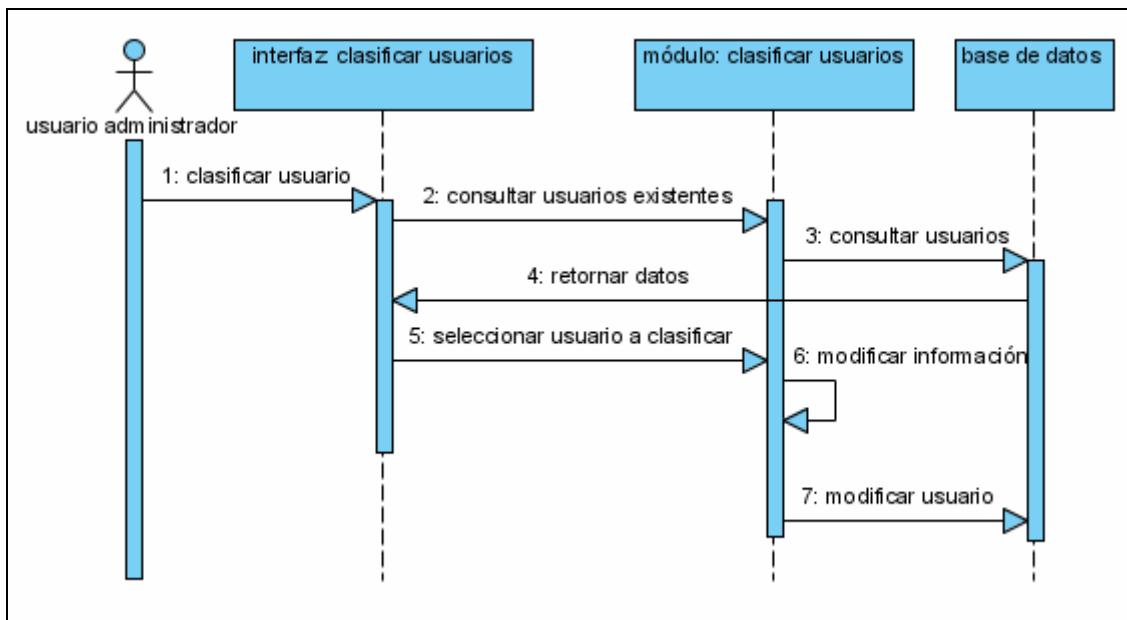


Fig. 3.22: Clasificar usuarios.

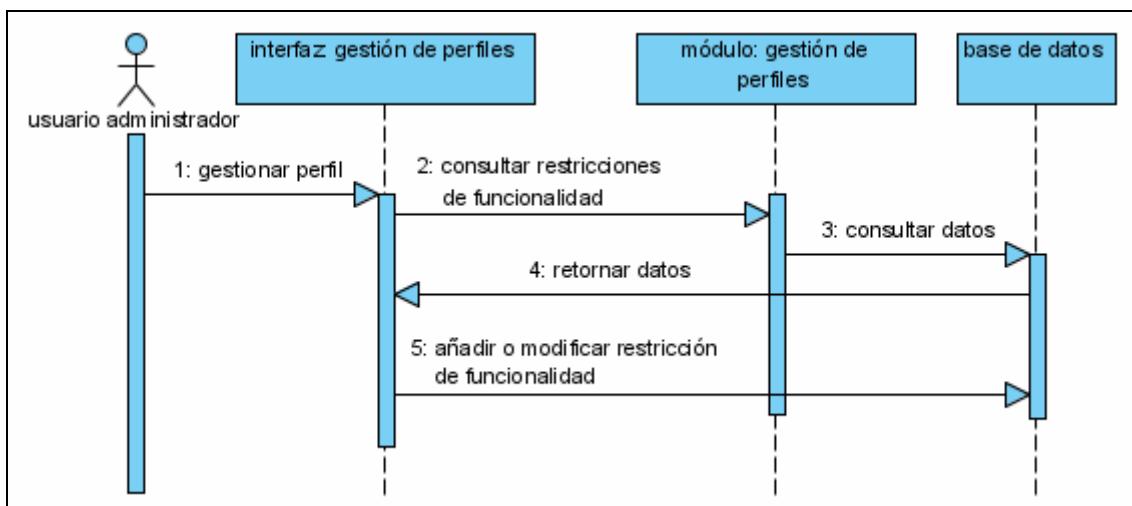


Fig. 3.23: Gestión de perfiles.

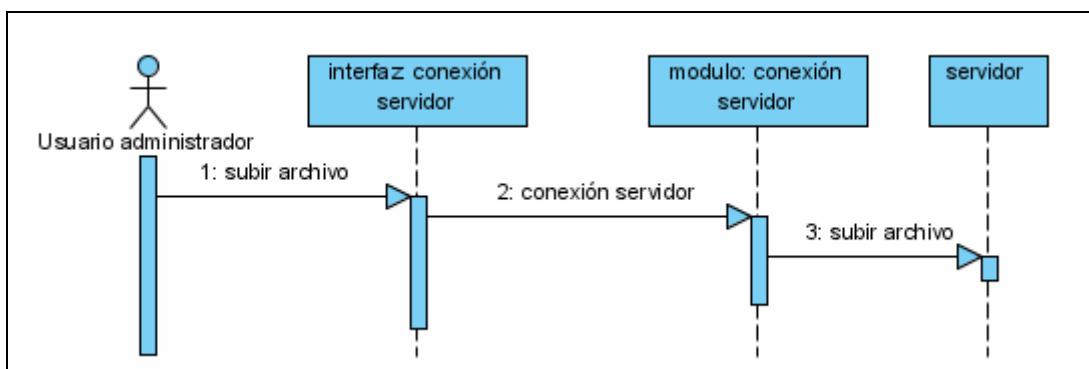


Fig. 3.24: Subir archivo al servidor.

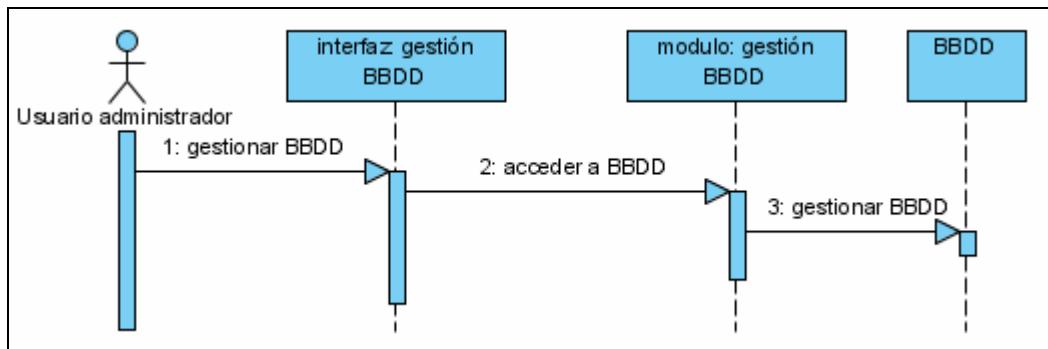


Fig. 3.25: Gestión de la BBDD.

### 3.2.5. Menús de navegación.

Al portal accederán diferentes perfiles de usuario, cada uno con funcionalidades específicas. Por tanto es necesario aplicar un sistema de menús de navegación. Este sistema se encargará de permitir a cada usuario únicamente el acceso a las funcionalidades que le sean permitidas en función del perfil de usuario al que pertenece.

Mediante un menú estático se accederá a aquellas funcionalidades que sean comunes para todo tipo de usuario que acceda al portal.

## 3.3. REQUERIMIENTOS NO FUNCIONALES.

Los requerimientos no funcionales hacen referencia a restricciones impuestas por el cliente o el propio desarrollador y restricciones impuestas por la dependencia de la aplicación con otras aplicaciones o software ya desarrollado o diseñado.

A continuación se detallan las restricciones no funcionales que se presentan en el proyecto.

### 3.3.1. RESTRICCIONES DE DISEÑO.

#### Dependencia de otras funcionalidades.

En el presente proyecto, se pretende realizar una funcionalidad que permita al usuario actualizar la base de datos del software de diseño y control de sistemas de iluminación con la base de datos de la aplicación. La aplicación por tanto, deberá poder realizar una conexión con la base de datos del cliente, que en este caso se trata de un sistema de datos *Microsoft Access*. Esta base de datos del cliente, incluye la tabla *mdf*, donde se almacenan la información de los modelos de funcionamiento, a la que se

accederá para consultar los modelos que el usuario posee y se incluirán aquellos que el usuario requiera.

Para la funcionalidad que permite al usuario comprobar y descargar actualizaciones de software, la aplicación deberá acceder a un archivo del usuario donde se especifica el software y la versión que tiene instalado el cliente. Este archivo se trata de un archivo de texto plano.

### **Seguridad.**

El portal WEB esta pensado para que sea utilizado por diferentes perfiles de usuarios, cada uno con acceso a diferentes funcionalidades. Por tanto, es necesario implementar un sistema de control de accesos.

Este sistema procurará básicamente permitir el acceso a las diferentes funcionalidades únicamente a los usuarios que estén autorizados. Por lo tanto, cada usuario tendrá una contraseña, que sólo conocerá él, y que utilizará para autenticarse.

Una vez hecha esta autenticación, en el menú de navegación se mostraran todas aquellas funcionalidades a las que el usuario puede acceder debido a sus características. Si un usuario intenta acceder a una funcionalidad a la que no esta autorizado, se le impedirá el acceso y se le notificará la incidencia.

El encargado de definir a que perfil pertenece cada uno de los usuarios que se registren es el Administrador de la aplicación WEB.

### **Prevención de errores en los formularios.**

Diversas funcionalidades con las que contará la aplicación WEB, requieren llenar formularios para la introducción de datos en la BBDD de la WEB. Aunque la mayoría de estas funcionalidades estarán disponibles para perfiles de usuario con conocimientos específicos de su área de trabajo, es necesario comprobar que se rellenan todos los campos obligatorios de los formularios y en el caso que sea necesario, comprobar que estos datos sean coherentes, para no comprometer la integridad de la BBDD.

### **3.3.2. OBJETIVOS DE DISEÑO.**

#### **Programación estructurada.**

El portal WEB que se obtendrá como resultado de la implementación del proyecto, es susceptible a sufrir mejoras o ampliaciones de las funcionalidades. Por ello, es conveniente que toda la programación se realice de manera modular y estructurada, además de utilizar todos los recursos de programación que permitan la comprensión del código fuente.

Por otro lado, la utilización de CSS (*Cascading Style Sheets*) puede facilitar los cambios futuros en el diseño del portal, por lo que su aplicación es altamente recomendable.

#### **Claridad.**

Es importante que toda la información que se presente en el portal se muestre de forma clara y coherente, es decir, utilizar un diseño que no dificulte la comprensión de los datos. Además el diseño debe ser formal ya que se trata de una WEB de una empresa.

#### **Accesibilidad y fácil manejo.**

En el portal existirán múltiples funcionalidades, y cada perfil de usuario podrá acceder a varias de estas funcionalidades. Es por esto que se debe facilitar al usuario el acceso a estas. Esto se conseguirá con un sistema de menús, con un menú específico a cada perfil de usuario, que permita el acceso a las funcionalidades de cada perfil de usuario.

## 4. DISEÑO.

---

### 4.1. INTRODUCCIÓN.

En este capítulo se analizarán la solución propuesta a los requerimientos que se enumeran en el capítulo **3. Análisis del proyecto**. Por tanto, todas las decisiones que se han tomado, y que están presentes en este capítulo, condicionan la resolución final del proyecto, e inclusive, las posibles ampliaciones o mejoras de éste.

En primer lugar se analizan los entornos de desarrollo que se emplearán en implementación de los requisitos funcionales del proyecto, describiendo los motivos por los cuales se han seleccionado.

A continuación se describe la configuración de la plataforma donde se dará servicio a la aplicación.

En el siguiente apartado se enumeran y describen las diferentes capas de la aplicación (entorno de usuario, motor de la aplicación y capa de datos).

En este mismo capítulo se describe la estructura de la base de datos de la aplicación, con una descripción de las tablas que la componen y de las relaciones entre las éstas.

A continuación se explicará la arquitectura de la aplicación, describiéndose la relación entre las diferentes capas de la aplicación.

En este capítulo también se incluye una descripción del diseño de las diferentes interfaces de usuario de la aplicación.

Por ultimo se describe la estructura de archivos y la política de back-up's a implementar en el proyecto.

#### **4.1.1. Selección del entorno de desarrollo.**

Para la implementación de la aplicación WEB se utilizarán diversos entornos de desarrollo, en función de las necesidades que implica el proyecto. Estos entornos de desarrollo pueden ser aplicados de forma simultánea y por tanto deben ser compatibles entre ellos.

En primer lugar dispondremos del lenguaje HTML, basado en el uso de etiquetas. Este lenguaje nos permitirá mostrar información al usuario de manera sencilla y de una forma atractiva. Además permite la interactividad del usuario mediante formularios con los que obtener información. Por otro lado, su uso es sencillo y permite al desarrollador explotar todas sus posibilidades con una curva de aprendizaje corta. A todo esto, cabe añadir que su utilización es, a día de hoy, la más extendida, y por tanto la compatibilidad del lenguaje con los diferentes navegadores Webs que se utilizarán para el acceso a la aplicación es total.

Como complemento al lenguaje HTML utilizaremos el entorno de desarrollo CSS. Crearemos una hoja de estilo que defina diversas etiquetas, que puedan ser utilizadas por el lenguaje HTML, y que nos definan el diseño del portal. Este entorno nos permite crear un diseño para la aplicación WEB que cumpla los requisitos de accesibilidad impuestos en el proyecto. Además permite al desarrollador crear un código claro y estructurado, de manera que el proyecto pueda ser modificado o mejorado de forma sencilla .Por otro lado, su uso nos permite crear hojas de estilo diferentes para modificar el diseño del portal en el futuro.

Otro entorno de desarrollo elegido para la implementación del proyecto es el lenguaje JavaScript. Éste nos permitirá crear funciones que se requieran en el proyecto. Su uso, dentro del desarrollo de este proyecto, se basará principalmente en la comprobación de datos introducidos por el usuario, o funciones de validación.

Por último utilizaremos el lenguaje de programación PHP. Este entorno nos permitirá gestionar la parte dinámica de la aplicación, es decir, será en enlace entre la capa de datos y el entorno de usuario. La interacción con la BBDD se realizará mediante

funcionalidades específicas incluidas en las librerías de PHP, lo que facilita esta interacción. Pero además, nos permite realizar conexiones a base de datos Microsoft Access mediante funciones incluidas en sus librerías. Además su uso es sencillo y requiere un tiempo de aprendizaje corto por parte del desarrollador. Por otro lado, se trata de software libre, lo que reduce el costo global del proyecto, sin necesidad de comprometer funcionalidades. Cabe destacar que al ser multiplataforma nos permite adaptarnos al sistema operativo *Microsoft Windows*.

#### **4.1.2. Selección de bases de datos.**

El sistema de gestión de base de datos elegido para el desarrollo del proyecto es MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, entre otras características. Este sistema nos permite almacenar y realizar consultas complejas de los datos de manera eficiente y varios usuarios al mismo tiempo. MySQL AB —desde enero de 2008 una subsidiaria de *Sun Microsystems* y esta a su vez de *Oracle Corporation* desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual, lo que reduce considerablemente los gastos del desarrollo e implantación del proyecto.

### **4.2. CONFIGURACIÓN DE LA PLATAFORMA.**

Una vez analizados los entornos de desarrollo que se utilizarán en el proyecto, debemos concretar la configuración de la plataforma donde se dará servicio a la aplicación.

En primer lugar, la aplicación WEB deberá estar hospedada en un servidor HTTP *Apache*, de acceso remoto a través de Internet. Éste deberá funcionar en un sistema operativo basado en Windows, como por ejemplo *Windows Server 2003*. La elección de este sistema operativo es debida en gran parte a que la aplicación deberá realizar conexiones a BBDD *Microsoft Access*.

Además el servidor deberá dar soporte al lenguaje PHP y deberá tener configuradas y activadas las funciones ODBC para conexiones a BBDD de *Microsoft Access* y las funciones específicas para el acceso a BBDD de *MySQL*.

## 4.3. CAPAS DE LA APLICACIÓN.

La aplicación a desarrollar estará constituida por tres capas diferenciadas.

### 4.3.1. Entorno de usuario.

El entorno de usuario es donde se nos muestra la información e incluso se le solicita información al usuario. Como se explica en el apartado **4.1.1 Selección del entorno de desarrollo**, esta capa se implementará utilizando lenguaje HTML y el uso de CSS.

En un archivo de estilo (style.css por ejemplo) definiremos diversas etiquetas que luego utilizaremos en los archivos con lenguaje HTML. Principalmente definiremos cuatro elementos: un gestor de autentificación (conocido normalmente como *log-in*), un menú lateral, el contenido y un cuerpo de página. Como complemento definiremos una cabecera y un pie de página. Además definiremos el estilo de cabeceras, enlaces, tipos y tamaños de letras y demás elementos que necesitemos.

Una vez definidos todos estos elementos podremos aplicarlos al diseño del portal. En primer lugar situaremos el cuerpo de la página. Éste, estará centrado en relación al área del navegador y situado sobre el fondo. A este cuerpo de página le sobrepondremos los otros tres elementos principales del diseño (gestor de autentificación, menú y contenido), además de la cabecera y el pie de página.

- **El gestor de autentificación** estará situado en la parte superior izquierda del cuerpo de la página, ocupando una cuarta parte aproximadamente del ancho de éste. Este gestor de autentificación servirá para que los usuarios registrados de la aplicación puedan acceder a las funcionalidades específicas de su perfil de usuario.
- **El menú lateral** de la aplicación estará situado a la izquierda, justo debajo del gestor de autentificación, y ocupará el mismo ancho que éste. Estará compuesto por dos partes: una primera estática, donde mostraremos el acceso a las páginas en las que se muestra información de la empresa, y una parte dinámica en la que se mostrarán las funcionalidades específicas a cada perfil de usuario.

- **El contenido** estará situado a la derecha de los dos elementos anteriores, ocupando el resto del ancho de la página. En éste mostraremos la información y contenidos que creamos necesarios así como los elementos de interacción entre el usuario y la base de datos.
- **La cabecera** estará situada en la parte superior del cuerpo de la página ocupando todo su ancho. En esta cabecera se pretende mostrar un logotipo de la empresa.
- Finalmente situaremos **el pie de página** en la parte inferior del cuerpo. En éste mostraremos aquella información adicional que creamos conveniente, como por ejemplo enlaces a las páginas oficiales de los entornos de desarrollo utilizados en el proyecto.

En la figura 4.01, que se muestra a continuación, podemos ver como quedaría este diseño, en el que se incluye la cabecera y el pie de página.

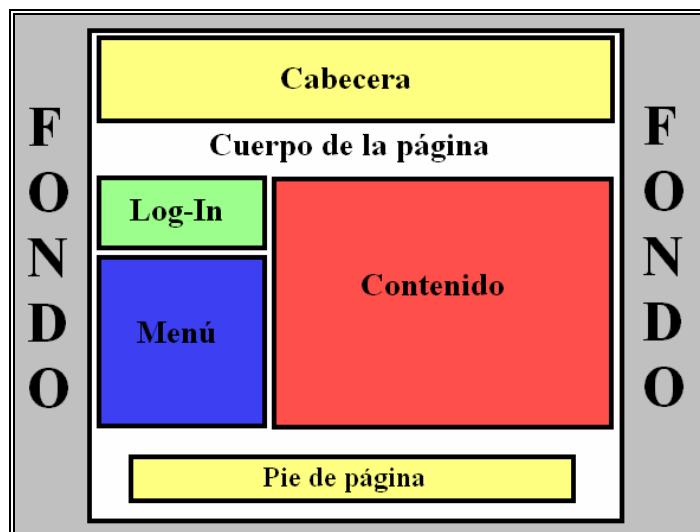


Figura 4.01: Diseño del entorno de usuario.

Para aplicar este diseño implementaremos en HTML el siguiente pseudo-código:

```
<inicio elemento cuerpo>
    <inicio elemento log-in>
        Código para la gestión de
        autentificaciones.
    <fin elemento log-in>

    <inicio elemento menú>
        Código para la gestión del
        menú.
    <fin elemento menú>

    <inicio elemento contenido>
        Código para mostrar
        información.
    <fin elemento contenido>
<fin elemento cuerpo>
```

#### 4.3.2. Motor de aplicación.

El motor de la aplicación será el encargado de relacionar el entorno de usuario con la capa de datos. Esta capa elegirá y generará los contenidos a mostrar por el entorno de usuario que, en la mayoría de casos, estarán almacenados en la base de datos de la aplicación. Así mismo, el motor de la aplicación, también recibirá información del entorno de usuario para que sea tratada por la capa de datos.

Para implementar esta interacción entre las tres capas se recurrirá al patrón de arquitectura de software Modelo-Vista-Controlador (MVC), adaptado a las necesidades del proyecto. La información referente a este patrón se encuentra en el apartado **4.5 Arquitectura de la aplicación**, de este capítulo.

Las diferentes aplicaciones estarán implementadas en archivos \*.php independientes. De esta manera facilitamos futuras ampliaciones o modificaciones de código. Los diferentes archivos tendrán una estructura similar, ya que su flujo de actuación es muy parecido.

El lenguaje escogido para implementar toda esta interacción es el lenguaje PHP. Su compatibilidad con el lenguaje HTML es total y permite recibir información del usuario, mediante formularios, de manera eficiente y sencilla y además posee funciones específicas para el acceso a bases de datos MySQL.

## **Funcionalidades especiales.**

En el presente proyecto, destacan algunas funcionalidades que otorgan al mismo un carácter especial. En primer lugar, cabe resaltar la funcionalidad que permitirá a los usuarios actualizar sus propias bases de datos de modelos de funcionamiento, a las que dan uso mediante el software que distribuye la propia empresa. Por otro lado, también destaca la funcionalidad que ayudará a los usuarios a descargar los archivos de actualización para el software que el cliente tenga adquirida la licencia.

Para la función de actualización de la base de datos, el motor de la aplicación accederá a la base de datos del cliente, y extraerá información de los modelos de funcionamiento de que dispone. Una vez realizada esta operación cotejará estos resultados con la base de datos del portal, con el fin de buscar los modelos que el cliente no tenga instalados en su base de datos. Estos serán mostrados al usuario, el cual podrá escoger los que desee añadir a su base de datos. Todos aquellos modelos de funcionamiento que hayan sido seleccionados se añadirán a la base de datos del cliente. Para realizar la conexión con la base de datos del cliente, el motor de la aplicación, deberá subir la BBDD al servidor, y realizar el acceso a través de funcionalidades específicas de PHP. Posteriormente se realizarán las modificaciones sobre esta BBDD, y se le retornará al usuario de manera que éste reemplace su anterior BBDD por ésta.

Para la funcionalidad de descarga de actualizaciones, el motor de la aplicación, solicitará el acceso a un archivo de registro, situado en la ruta de instalación del software, donde se especifica el software que el cliente tiene instalado en su ordenador, y sus versiones. A continuación, el motor de la aplicación buscará las versiones del software que sean más recientes en la base de datos del portal y se las mostrará al usuario. Para aquellas actualizaciones que el usuario disponga de licencia, se mostrará el acceso a la descarga de los archivos de actualización del software. Para las actualizaciones que el usuario no disponga de licencia simplemente se le informará de que existe una versión más reciente del software.

## **Uso de herramientas de apoyo.**

Para la implementación del proyecto se ha decidido utilizar varias herramientas de apoyo que ayuden a reducir el tiempo de desarrollo del mismo y la consecución de objetivos. Estas herramientas son *TinyMCE*, que facilitará la edición de textos para su

correcta visualización, y *phpMyAdmin* y *WinSCP*, que se utilizará para las funcionalidades del usuario administrador del portal WEB.

*TinyMCE* es una aplicación *JavaScript* diseñada para la edición de textos. Una vez insertada y activada en la aplicación, añade una serie de funcionalidades que permiten al usuario crear contenidos, a los que se le añaden etiquetas HTML, de una forma sencilla y práctica. Además, es compatible con todos los navegadores WEBs con los que será compatible la aplicación, es de fácil integración en el proyecto y es de distribución gratuita. Por estos motivos, esta herramienta se añadirá a las funcionalidades de inserción y edición de noticias, productos, versiones software y modelos de funcionamiento.

*phpMyAdmin* es una aplicación que se utiliza para administrar bases de datos en MySQL desde un entorno WEB. Es, además, un programa de distribución libre y su uso está muy extendido en los servidores WEBs con sistemas de base de datos MySQL. Su uso es sencillo e intuitivo. Por todo esto, se ha decidido su aplicación en las funcionalidades del usuario administrador, ya que estas se resumen en la administración de los contenidos de la BBDD del portal.

Por último, el administrador deberá subir al servidor diferentes archivos, para el funcionamiento de la aplicación (archivos de actualización, imágenes, etc.). En el caso que el administrador no pueda tener acceso directo al servidor, utilizará la aplicación *WinSCP*. *WinSCP* es un cliente SFTP (*Secure File Transfer Protocol*) para entorno Windows que emplea SSH (*Secure SHell*). La aplicación también puede trabajar bajo el protocolo SCP (*Secure Copy*) y FTP (*File Transfer Protocol*), menos usados a día de hoy. La herramienta es un software libre, por lo que su aplicación en el proyecto no supone un incremento en el coste del mismo.

#### **4.3.3. Capa de datos.**

En la capa de datos guardaremos toda la información que necesitemos (información de los usuarios, noticias, productos, etc.). Esta información la almacenaremos en una base de datos, diseñada para la aplicación. El sistema de gestión para la base de datos de la aplicación es MySQL.

En esta capa, recibiremos información para la inserción, modificación, eliminación o selección de datos de la base de datos, del motor de la aplicación. Toda

esta interacción se realizará mediante el uso de consultas del Lenguaje de Consultas Estructurado (Structured Query Language, SQL).

En el siguiente apartado de esta capítulo se muestra la estructura de la base de datos.

#### **4.4. ESTRUCTURA DE LA BASE DE DATOS.**

La base de datos diseñada para el presente proyecto estará estructurada en tablas diseñadas en función de las necesidades del proyecto. Cada tabla contendrá campos, donde almacenaremos la información deseada. Además, existirán relaciones entre las diferentes tablas, para facilitar el uso de la BBDD y para crear consultas complejas.

En la figura 4.02, observamos la estructura de esta base de datos, y a continuación una explicación de las diferentes tablas y campos más importantes que la componen.

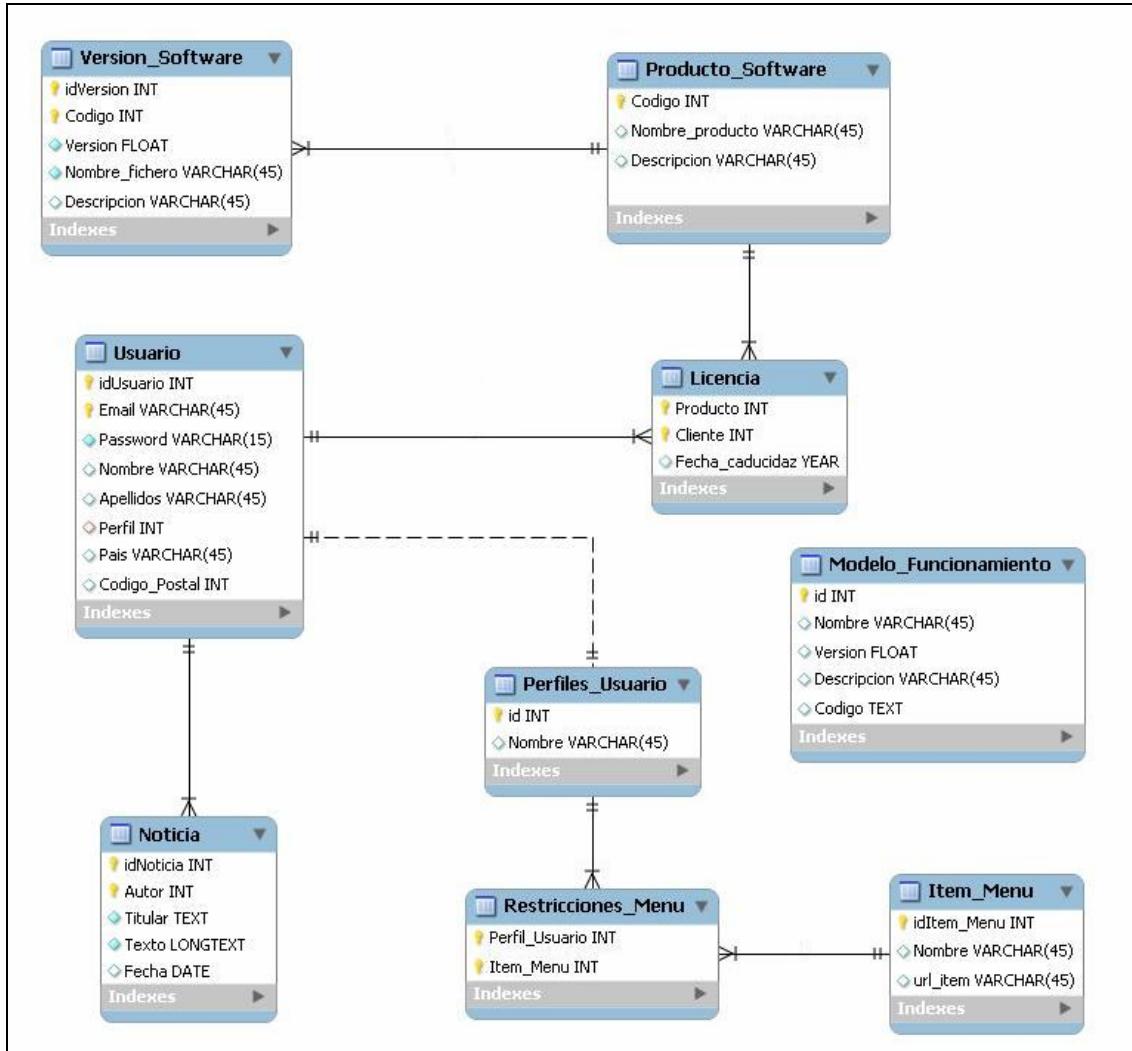


Figura 4.02: Estructura de la BBDD.

## Tabla Usuario

Tabla donde almacenaremos la información referente a los usuarios. Cuenta con los siguientes campos:

- idUsuario: identificador único, que nos asegura la integridad de los datos. De tipo entero y auto-incremental.
- Email: dirección de correo electrónico del usuario. El usuario introducirá su email para ingresar en la aplicación. De tipo cadena de caracteres.
- Password: contraseña que utilizará el usuario para acceder a la aplicación. De tipo cadena de caracteres. El contenido de este campo estará protegido mediante un algoritmo de cifrado de información.

- Perfil: perfil de usuario al que pertenece el usuario. De tipo entero. Por defecto se completará el campo con un 1 (para los usuarios normales).

Además se le añadirán los campos referentes a información personal del usuario que creamos convenientes (nombre y apellidos, país de origen, etc.).

### **Tabla Noticia**

Tabla donde almacenaremos las noticias de la WEB. Esta relacionada con la tabla de usuarios. Cuenta con los siguientes campos:

- idNoticia: identificador único, que nos asegura la integridad de los datos. De tipo entero y auto-incremental.
- Autor: nos especifica el autor de la noticia. Esta relacionado con el campo idUsuario de la tabla Usuario. Es de tipo entero.
- Titular: titular de la noticia. De tipo cadena de caracteres.
- Texto: cuerpo de la noticia. De tipo texto largo.
- Fecha: fecha de inserción de la noticia. De tipo fecha.

### **Tabla Producto\_Software**

En esta tabla almacenaremos la información de los diferentes productos software que la empresa desarrolla, y que se desea mostrar en el portal. Contiene los siguientes campos:

- Código: identificador único, que nos asegura la integridad de los datos. De tipo entero y auto-incremental.
- Nombre\_Producto: nombre del producto. De tipo cadena de caracteres.
- Fabricante: fabricante del producto. De tipo cadena de caracteres.
- Descripción: información a mostrar sobre el producto. De tipo texto.

### **Tabla Version\_Software**

Tabla donde guardaremos la información sobre las diferentes versiones del software que distribuye la empresa. Estará, por tanto, relacionada con la tabla Producto\_Software. Contiene los siguientes campos:

- idVersion: identificador único, que nos asegura la integridad de los datos. De tipo entero y auto-incremental.

- Código: especifica la versión de software con el software correspondiente. Esta relacionado con el campo Código de la tabla Producto\_Software. De tipo entero.
- Version: nos indica a que número de versión pertenece. De tipo real.
- Nombre\_fichero: especifica el fichero de actualización con el que va asociada la versión del software.
- Descripción: información a mostrar sobre las mejoras que aporta la versión del software. De tipo texto.

### **Tabla Version\_Software**

En esta tabla almacenaremos las licencias de software de los diferentes usuarios. Por tanto, tiene una relación directa con la tabla Usuario y la tabla Producto\_Software. Contiene los siguientes campos:

- Producto: especifica a que producto esta asociada la licencia. Esta relacionado con el campo Código de la tabla Producto\_Software. De tipo entero.
- Cliente: especifica el usuario propietario de la licencia. Esta relacionado con el campo idUsuario de la tabla Usuario. De tipo entero.
- Fecha\_caducidad: informa del año de vigencia de la licencia. De tipo YEAR (tipo de variable específica de PHP).

### **Tabla Perfiles\_Usuario**

En esta tabla guardaremos los distintos perfiles de usuario que soporta la aplicación. Contiene los siguientes campos:

- id: identificador único, que nos asegura la integridad de los datos. De tipo entero.
- Nombre: nombre del perfil de usuario. (Normal, Gestor de Contenidos, Gestor de Noticias y Administrador). De tipo cadena de caracteres.

### **Tabla Item\_Menu**

En esta tabla almacenaremos todos los posibles ítems de menú asociados al acceso a las distintas funcionalidades del portal. Contiene los siguientes campos:

- idItem\_Menu: identificador único, que nos asegura la integridad de los datos. De tipo entero.
- Nombre: nombre del ítem. De tipo Cadena de caracteres.
- url\_item: localizador de la funcionalidad asociada al ítem de menú. De tipo cadena de caracteres.

### **Tabla Restricciones\_Menu**

Tabla para guardar los ítems de menú a los que puede acceder cada perfil de usuario. Por tanto esta relacionada tanto con la tabla Perfiles\_Usuario como con la tabla Item\_Menu. Contiene los siguientes campos:

- Perfil\_usuario: relacionado con el campo id de la tabla Perfiles\_Usuario. De tipo entero.
- Item\_Menu: relacionado con el campo idItem\_Menu de la tabla Item\_Menu. De tipo entero.

### **Tabla Medelo\_funcionamiento**

En esta tabla almacenaremos la información de los diferentes modelos de funcionamiento que la empresa desarrolle. Contiene los siguientes campos:

- Id: identificador único, que nos asegura la integridad de los datos. De tipo entero.
- Nombre: nombre del modelo de funcionamiento. De tipo cadena de caracteres.
- Versión: número de versión del modelo de funcionamiento. De tipo real.
- Descripción: información referente al modelo de funcionamiento que se deseé mostrar. De tipo texto.
- Código: código DMX asociado al modelo de funcionamiento. De tipo texto.

## 4.5. ARQUITECTURA DE LA APLICACIÓN.

Como ya se ha explicado en el apartado **4.3 Capas de la aplicación**, disponemos de tres capas diferenciadas: el entorno de usuario, el motor de la aplicación y la capa de datos. Para gestionar la interacción entre estas tres capas utilizaremos el patrón de desarrollo de software Modelo-Vista-Controlador, con algunas modificaciones para adecuarlo a las necesidades del proyecto. Este patrón de desarrollo de software está definido por un flujo similar al siguiente donde la *vista* es nuestro entorno de usuario, el *controlador* es el motor de la aplicación y el *modelo* es la capa de datos:

1. El usuario interactúa con el entorno de usuario mediante algún elemento interactivo (botones, enlaces, etc.).
2. El controlador recibe la notificación de acción solicitada por el usuario así como la información que se envíe con la acción. Si es necesario el controlador valida y/o procesa la información.
3. El controlador accede al modelo, realizando la acción que se solicite sobre éste.
4. El controlador envía información a la vista para que sea mostrada. Esta información puede venir del modelo o ser únicamente información de validación de la acción.
5. El entorno de usuario espera nuevas acciones, comenzando el ciclo nuevamente.

En la figura 4.03 observamos la interacción entre estas tres capas.

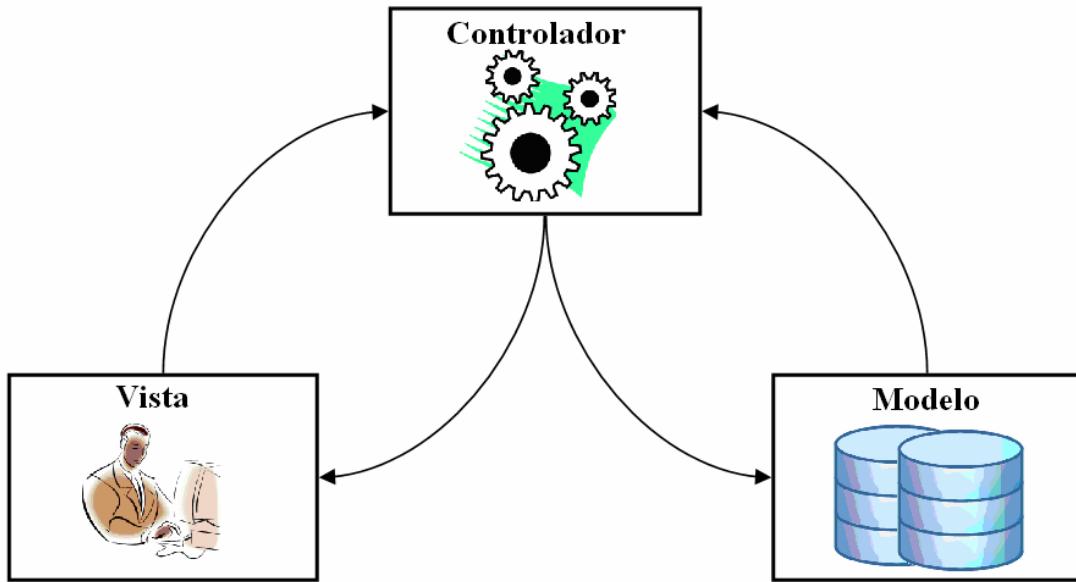
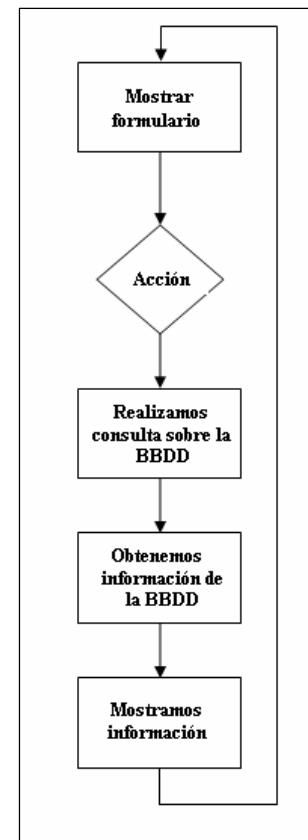


Figura 4.03: Esquema del patrón Modelo-Vista-Controlador

En el presente proyecto, se pueden diferenciar básicamente dos tipos de funcionalidades según la interacción con el usuario: aquellas en las que simplemente se muestran información de la BBDD (p.e. ver noticias), y aquellas en las que el usuario introduce información para actuar sobre la BBDD (p.e. introducir noticias). Por tanto definiremos dos flujos de acción para estas funcionalidades.

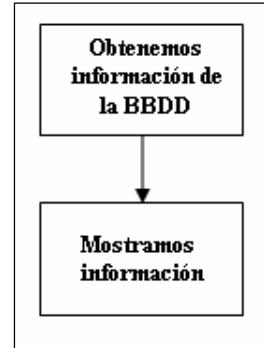
A continuación vemos el flujo para las funcionalidades en las que se introduce información:

1. Mostramos formulario para introducir información.
2. Informamos al controlador sobre la acción a realizar. Recogemos información, y la validamos y/o procesamos.
3. Realizamos la acción sobre la BBDD.
4. Obtenemos información de la BBDD, si es necesario, y mostramos esta información o un mensaje de ‘operación correcta’.
5. Iniciamos en el punto 1.



A continuación vemos el flujo para las funcionalidades en las que simplemente se muestra información de la BBDD:

1. Obtenemos información de la BBDD.
2. Mostramos información al usuario.



## 4.6. INTERFASE.

Para el presente proyecto se ha creado un diseño que será aplicado, de forma general, a todas las interfaces de la aplicación, explicado en el apartado **4.3.1 Entorno de usuario**. En este diseño aparecen dos elementos que variarán en función del usuario que acceda a la aplicación y de la funcionalidad en la que se encuentre el usuario: el menú dinámico y el contenido, respectivamente.

### 4.6.1. Definición de las Interfaces para diferentes perfiles de usuario.

Para el acceso a las diferentes funcionalidades del portal WEB, se ha diseñado un menú dinámico. Éste, variara en función del perfil de usuario que acceda a la aplicación de manera que únicamente muestra las funciones a las que tenga permitido el acceso el usuario activo. Para que esto sea posible, en la base de datos se almacenarán todas las funcionalidades disponibles en el portal, con su correspondiente ruta de acceso. Estas estarán asociadas a los diferentes perfiles de usuario, de manera que estas relaciones puedan ser modificadas según se necesite. En la figura 4.04 se muestran los diferentes menús de usuario.

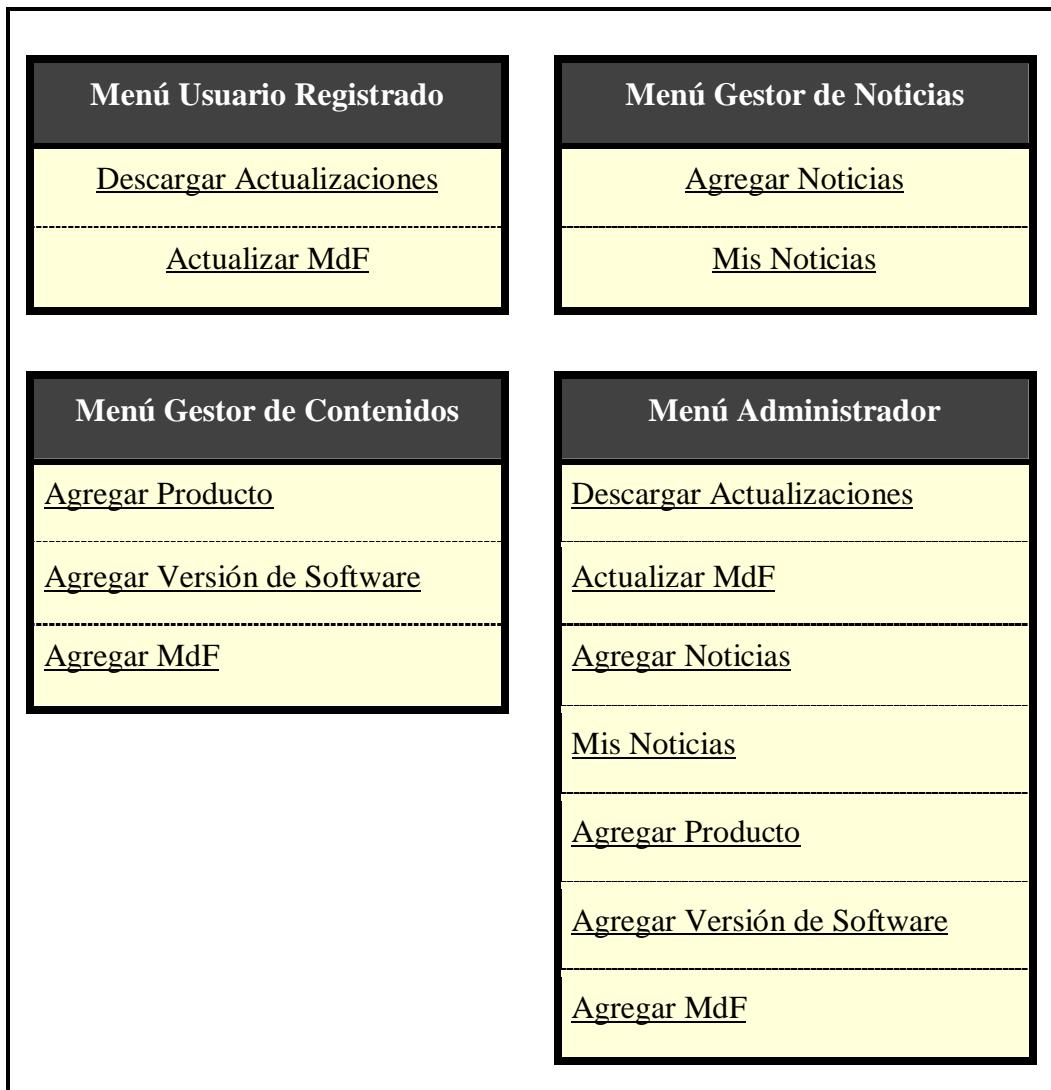


Fig. 4.04: Menús de usuario

Este menú, permitirá acceder a las diferentes funcionalidades. Cada una de estas tendrá una interfase diferenciada, que serán mostradas en el contenido del diseño del portal. A continuación se describen la interfase para las funcionalidades más importantes de cada perfil de usuario.

### Usuario registrado

Para la funcionalidad de actualización de la Base de datos de modelos de funcionamiento, se ha diseñado una interfase en la que al usuario se le mostrarán los modelos de funcionamiento disponibles que no posee en su BBDD. El usuario podrá seleccionar mediante *ticks* (checkboxs de HTML) que modelos desea agregar a su BBDD.

**Seleccione los MdF que desea agregar a su BBDD**

Modelo de funcionamiento 1

Modelo de funcionamiento 2

Modelo de funcionamiento 4

Modelo de funcionamiento 10

**Figura 4.05:** Actualización de MdF.

En la interfase para la funcionalidad de descarga de actualizaciones, se le mostrara al usuario el software que tiene instalado, la versión del mismo, y la opción de descarga si posee licencia.

Software:	Version Actual	Version Disponible	Licencia
Diseño de equipos de iluminación	1.02	1.03	<a href="#">Descargar Versión</a>
Diseño de MdF	1.01	1.01	-
Diseño de sistemas	1.01	1.04	No dispones de la licencia

**Figura 4.06:** Descarga de actualizaciones.

### Usuario gestor de noticias

La funcionalidad principal del gestor de noticias es la de agregar y modificar o eliminar noticias a la base de datos. Para estas funcionalidades se ha diseñado una interfase basada en el uso de formularios de HTML. En estos formularios aparecerán todos los campos relacionados con la tabla Noticia de la BBDD de la aplicación (excepto los campos añadidos para la gestión de la BBDD).



Formulario para la creación de noticias. Contiene un campo para el título y un campo de texto grande con scroll vertical.

**Figura 4.07: formulario noticias.**

Cabe destacar que en el texto de la noticia, el usuario gestor de noticias, dispondrá de una pequeña aplicación JavaScript (TinyMCE). Esta aplicación consiste en un procesador de textos, con la particularidad de que al texto obtenido se le añaden etiquetas HTML para su visualización en portales WEBS. De este modo, la funcionalidad encargada de visualizar las noticias, interpretará las etiquetas HTML y mostrará la noticia con el estilo que el gestor elija.

### **Usuario gestor de contenidos**

El usuario gestor de contenidos es el encargado de agregar, modificar y eliminar tanto los productos software del catálogo, como las versiones asociadas a estos productos, como los modelos de funcionamiento que desarrolla la empresa. Para estas funcionalidades se han diseñado unos formularios con los campos necesarios.



Formulario para la creación de productos. Contiene un campo para el nombre del producto y un campo para la descripción, ambos con scroll vertical. A continuación, hay un botón para insertar.

**Figura 4.08: Formulario producto.**

Producto Software al que esta asociada la actualización:

Producto Asociado

Descripción:

Version:

Formato: X.xx

Nombre del fichero asociado a la actualización:

**Figura 4.09: Formulario versión.**

Nombre:

Version:

Formato: X.xx

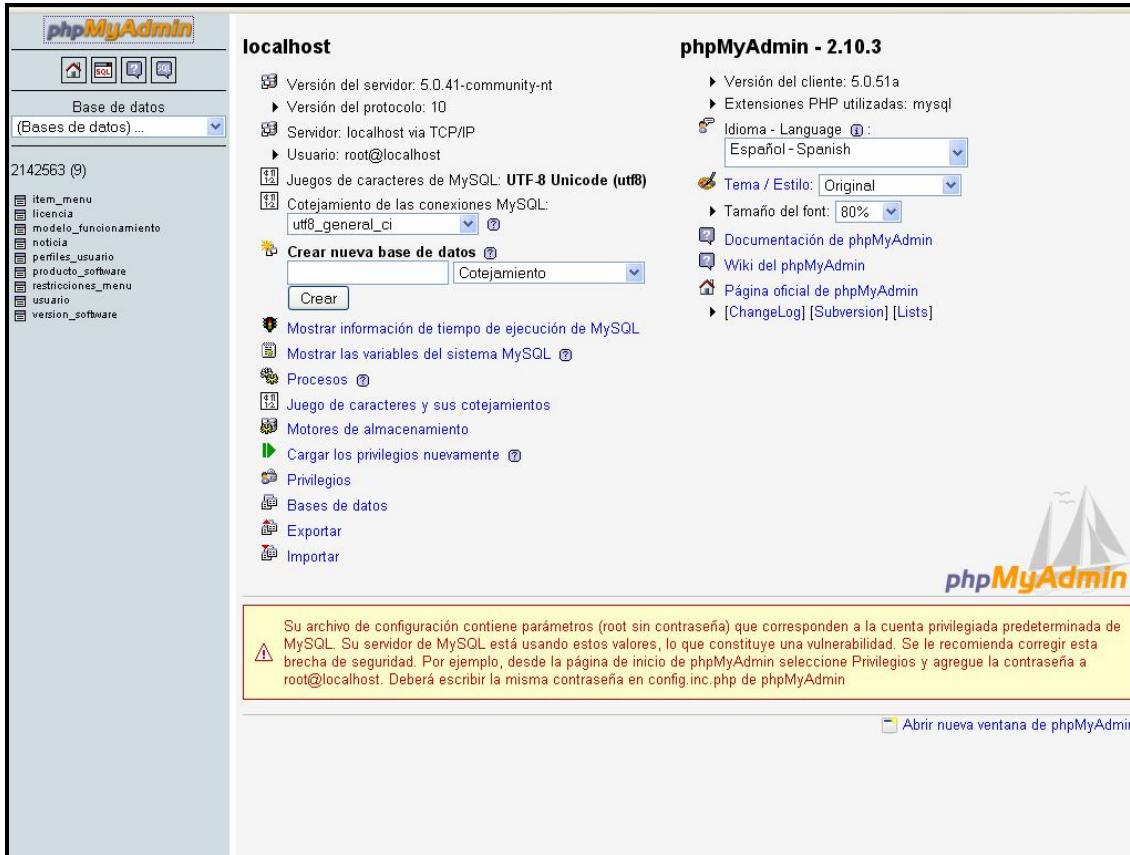
Descripción:

Código DMX512:

**Figura 4.10: Formulario MdF**

## Usuario administrador

Como se explica en el apartado **4.3.2 Motor de la aplicación**, para las funcionalidades de control y gestión de la BBDD, del usuario administrador, se ha decidido utilizar la herramienta *phpMyAdmin*. Esta herramienta permitirá al administrador realizar todas sus funcionalidades bajo el mismo entorno.



**Fig. 4.11: Entorno de phpMyAdmin**

Para las funcionalidades que requieren subir archivos al servidor, el administrador hará uso de la aplicación WinSCP.

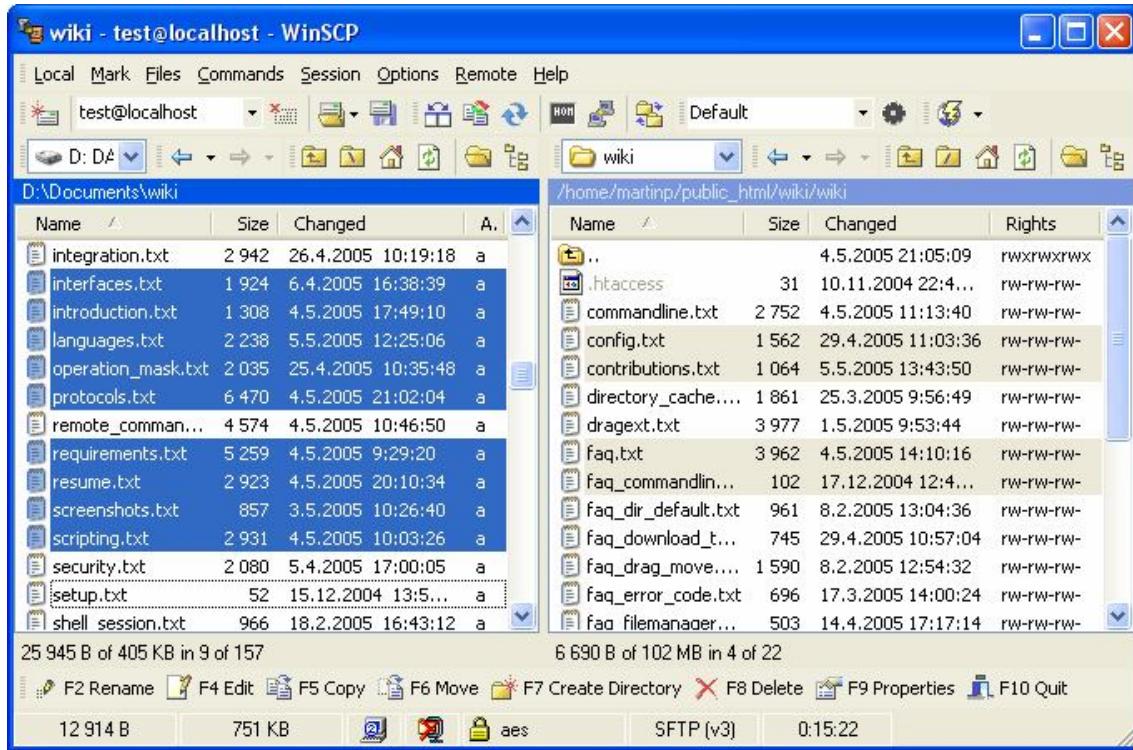


Fig. 4.12: Entorno de WinSCP

## 4.7. ESTRUCTURA DE ARCHIVOS Y POLÍTICA DE BACKUP'S.

Para una mejor organización crearemos una estructura de archivos donde almacenar todos aquellos ficheros que serán necesarios para el funcionamiento de la aplicación.

En el directorio *raíz* (ilusoft), almacenaremos todos los archivos de las funcionalidades de la aplicación. En éste, además, encontramos directorios para almacenar los demás archivos que necesitemos:

- **Docs:** en este directorio almacenaremos los documentos que definen la política de privacidad y términos de uso. Además almacenaremos los documentos relacionados con la aplicación como por ejemplo *tutoriales*.

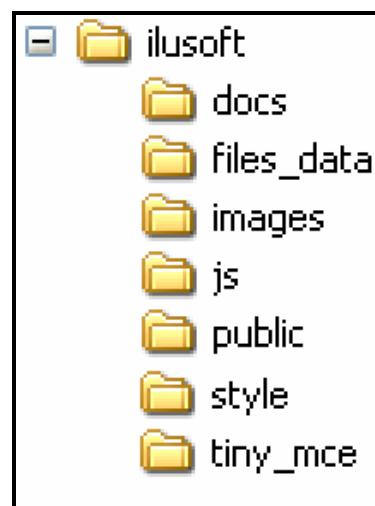


Fig. 4.13: Estructura de archivos

- ❑ **Files\_data:** en este directorio se almacenarán los archivos de actualización de software.
- ❑ **Images:** en este directorio se almacenarán las imágenes que se utilizan en la aplicación, tanto las relacionadas con las interfaces como las imágenes que aparezcan en las noticias o productos.
- ❑ **Js:** en este directorio se almacenarán los archivos *JavaScritp* de la aplicación.
- ❑ **Public:** este es un directorio destinado a almacenar los archivos temporales que se generen en las funcionalidades que lo requieran.
- ❑ **Style:** en este directorio se almacenarán las hojas de estilo (archivos .css) necesarios para la visualización de las interfaces.
- ❑ **Tiny\_mce:** este directorio es el correspondiente a la aplicación *TinyMCE* de edición de textos. Contiene varios subdirectorios y archivos JavaScript.

Por otro lado, es importante que se realicen back-up's para garantizar la seguridad de la aplicación. Una gran parte de las empresas de alquiler de servidores disponen de servicio de back-up's para bases de datos, aunque suele ser necesario configurar el sistema. En el caso que no sea posible realizar back-up's automáticos, el administrador se deberá encargar de realizarlos de manera manual, a través de la aplicación *phpMyAdmin*. Además, el administrador, deberá hacerse cargo de tener copias de seguridad de los archivos de actualización.

## 5. IMPLEMENTACIÓN.

---

### 5.1. INTRODUCCIÓN.

En este capítulo se detallan las soluciones implementadas para las diferentes funcionalidades de la aplicación. Esta implementación es consecuencia directa del capítulo 4: **Diseño**, y se han respetado todas las decisiones que se especifican en éste.

En este capítulo se describen los detalles más importantes de las capas de la aplicación. En primer lugar se describen los detalles relacionados con el entorno de usuario. A continuación se hace referencia a la implementación del motor de la aplicación, haciendo hincapié en las funcionalidades que tienen un mayor peso dentro de la aplicación. Luego, se describe los aspectos más relevantes de la implementación de la capa de datos.

Por último se especifica la manera en la que se han integrado las diferentes herramientas o aplicaciones que darán apoyo a la aplicación.

## 5.2. CODIFICACION DE LAS DIFERENTES CAPAS DE LA APLICACIÓN.

### 5.2.1. Entorno de usuario.

La capa de entorno de usuario, es la encargada de realizar la comunicación con el usuario de la aplicación. Tal y como se especifica en el capítulo 4: Diseño, las diferentes interfaces se han codificado utilizando el lenguaje de etiquetas HTML, combinado con el uso de CSS, que nos permite estructurar la información mediante el uso de capas.

Para el diseño de la aplicación se ha recurrido a una plantilla CSS obtenida del portal WEB *Free CSS Templates*<sup>5</sup> ([www.freecsstemplates.org](http://www.freecsstemplates.org)). Esta plantilla ha sido modificada a fin de adaptarla a las condiciones de diseño y visualización del proyecto. Este diseño, a sido implementado en el archivo *style.css*, y se enlaza a él en cada uno de los archivos de la aplicación, a fin de no incluir el código de estilo en cada uno de estos archivos.

Como se especifica en el apartado **4.3.1. Entorno de usuario**, en este archivo *style.css*, se especifica el formato de los diferentes elementos que aparecen en la aplicación (tipo de letra del texto, formato de los enlaces, títulos, etc.). Además se incluye la posición en la que aparecerán los diferentes elementos de la interfase (cabecera, menú, contenido, etc.), definiendo así la estructura de la aplicación, tal y como se especifica en el capítulo de diseño.

Cabe mencionar que, en la aplicación se incluyen elementos de uso específico (tablas para mostrar una información determinada, o separadores de información) que han sido incluidos en la interfase mediante el uso de etiquetas HTML de forma directa. Ejemplo de esto sería la tabla donde se muestra la información referente a las actualizaciones de software, que posee un estilo no incluido en el archivo de estilo.

---

<sup>5</sup> Free Css Templates permite utilizar y modificar todas sus plantillas, con la condición que se incluya en la aplicación un enlace a su portal. Esta referencia se encuentra en el pie de página de todas las páginas de la aplicación.

### 5.2.2. Motor de la aplicación.

EL motor de la aplicación realiza la función de enlace con entre el entorno de usuario y la capa de datos, verificando, tratando y adaptando la información que se envíe entre ambas. Esta manipulación de la información se hace mediante el uso del lenguaje de programación *PHP*, que nos permite tanto el uso del lenguaje de etiquetas *HTML*, como la interacción con el sistema de base de datos *MySql*.

Las diferentes funcionalidades que integra la aplicación se han dividido en archivos distintos. Cada uno de estos archivos contiene la estructura principal de la aplicación a la que se le añade el código necesario para la funcionalidad correspondiente. Además, en el proyecto, hay una serie de archivos que contienen código que se aplica varias veces en la aplicación o que pueden ser susceptibles a cambios futuros (gestión de menús, conexión con la base de datos, etc.). El código de estos archivos se incluye en la aplicación cuando es necesario mediante el uso de la función *include* de *PHP*.

En la aplicación encontramos una serie de funcionalidades que consisten en la visualización, inserción o modificación de la información que contiene la base de datos. En las funcionalidades de inserción o modificación, se obtiene la información a través de formularios. Cuando el usuario lo desea, la información es validada por la aplicación y ésta genera una consulta de inserción/actualización que pueda ser interpretada por *MySql*. Cuando la funcionalidad consiste en visualizar información de la BBDD, la aplicación realiza una consulta de selección de información a la base de datos, y prepara dicha información para que sea interpretada por el entorno de usuario. Estas consultas se realizan a través de funcionalidades específicas de *PHP*<sup>6</sup>.

Por otro lado en el proyecto, se incluyen funcionalidades, que aunque interactúan también con el entorno de usuario y la capa de datos, poseen características especiales. Entre éstas, encontramos funcionalidades que cumplen requerimientos no funcionales (seguridad, accesibilidad) y funcionalidades que requieren un tratamiento especial, ya que cumplen con requerimientos funcionales no habituales en portales WEB clásicos de gestión de contenidos,

---

<sup>6</sup> Anexo: Referencia 2. Funciones de PHP para la gestión de BBDD. Pàg 86.

## Autentificarse.

La funcionalidad de autentificarse, permite tener un control de la seguridad de la aplicación. Dicha funcionalidad esta implementada en el archivo *login.php* del proyecto, y se incorpora en la estructura de la aplicación, de manera que es ejecutada en todas las demás funcionalidades.

Esta funcionalidad está implementada con el uso de *cookies*. El usuario, sin estar autenticado, introduce su correo electrónico y la contraseña de su registro en la WEB, mediante el formulario de la funcionalidad (figura 5.1: Formulario autentificación). Esta información es almacenada en las *cookies* del navegador WEB del usuario<sup>7</sup>, encriptando la información mediante el uso de la función SHA-1 de *PHP*. Estas cookies permanecen en el navegador durante una hora (por defecto) o un día. Una vez llenadas las *cookies*, la funcionalidad coteja la información con la base de datos, y en el caso que sea correcta valida la autentificación. Además, cuando la autentificación es correcta, se guarda en una variable el registro que identifica al usuario en la BBDD (*\$user\_actived*), y el perfil de éste (*\$perfil\_user*).

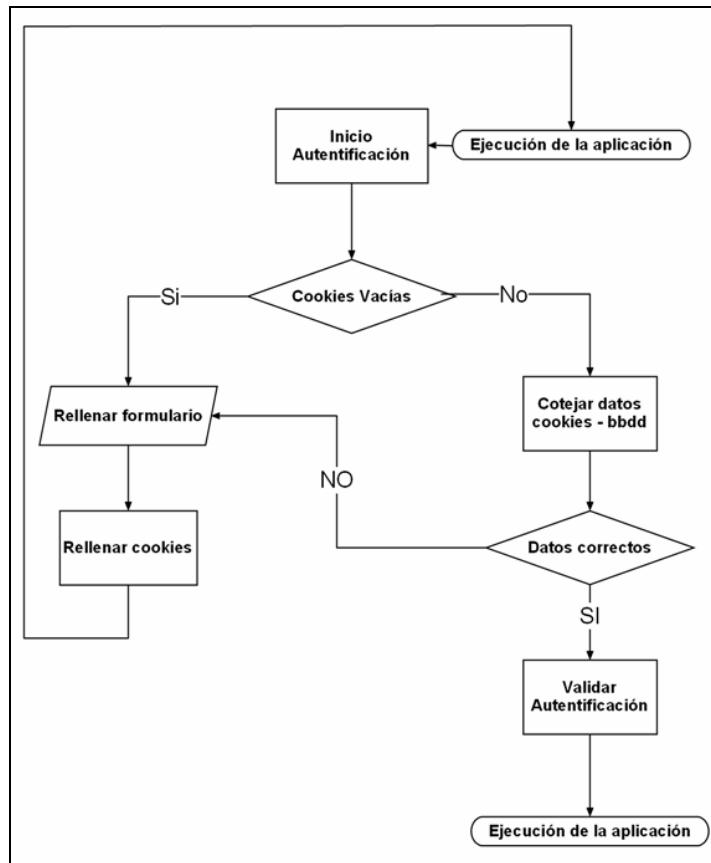
The image shows a screenshot of a web form titled "Mi Cuenta". The form has a dark grey header bar with the title. Below it is a white input field labeled "Email". Underneath the "Email" field is another input field labeled "Contraseña". At the bottom left of the form is a checkbox labeled "Recordar". To the right of the "Recordar" checkbox is a blue rectangular button labeled "Acceder".

Fig. 5.1: Formulario autentificación.

En la figura 5.2, se muestra el diagrama de flujo de la funcionalidad.

---

<sup>7</sup> Anexo: Referencia 3. Código para la gestión de cookies. Pág. 87.

**Fig. 5.2: Diagrama de flujo autentificación.**

### **Mostrar Menú.**

Para cumplir con el requerimiento que especifica que la aplicación debe tener una buena accesibilidad, se ha diseñado un sistema de menús dinámicos. Estos menús cambian su contenido en función del perfil de usuario que accede a la aplicación, mostrando únicamente aquellas funcionalidades a las que tiene acceso. El código para esta funcionalidad se encuentra en el archivo *menu.php*, y éste se integra en la estructura de la aplicación, de manera que es ejecutada en todas las funcionalidades.

El menú, en primera instancia carga los enlaces a las funcionalidades correspondientes al perfil de usuario no registrado, que están almacenados en la base de datos de la aplicación para una mayor flexibilidad. En el caso que el usuario haya

**Fig. 5.3: Menú (usuario registrado)**

realizado la autentificación y sea válida, se obtienen de la BBDD los enlaces a las funcionalidades asociadas al perfil de usuario activado.

La gestión de estos menús, se realiza a través de la base de datos de la aplicación, lo que permite realizar cambios en las funcionalidades asociadas a los distintos perfiles de usuario, tal y como se especifica en el apartado 5.2.3 Capa de datos. En la figura 5.4 se muestra el diagrama de flujo de la funcionalidad.

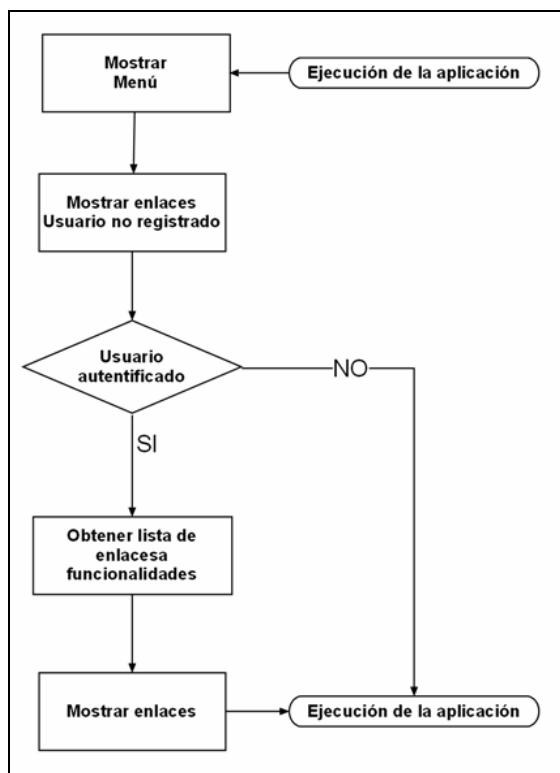


Fig. 5.4: Diagrama de flujo mostrar menú.

### **Comprobar/Descargar actualizaciones de software.**

Debido a que las funcionalidades de comprobar actualizaciones de software y descargar actualizaciones comparten gran parte de sus secuencias, se ha decidido implementar estas funcionalidades en una única funcionalidad. En primer lugar, la aplicación comprueba las versiones de software más recientes a las que posee el usuario, y finalmente se le da la opción a éste de descargar el archivo de actualización, en el caso que tenga la licencia correspondiente. El código de esta funcionalidad, se encuentra el archivo *comprobar\_actualizaciones.php*.

Para comprobar que software, y que versión de éste, tiene instalado el usuario, la aplicación ha de acceder a un archivo almacenado en el PC del usuario donde se especifican estos valores. Para este acceso, el archivo ha de ser subido al servidor,

mediante un formulario donde el usuario especifica la ruta del archivo. Cuando el usuario verifica la ruta, el archivo es subido a un directorio temporal del servidor, y posteriormente se le cambia el nombre (se le añade el identificador de usuario) con el fin que no haya concurrencia de archivos en el servidor. Este archivo es movido a otro directorio dentro de la aplicación donde ésta pueda acceder a él. Del archivo se obtienen la información que se necesita (id del software y versión), y se coteja con los datos de la base de datos de la aplicación con el fin de encontrar versiones de software más actuales. Al usuario se le muestra la información (figura 5.5), y en el caso que disponga de licencia, se le permite la descarga del archivo de actualización. Para asegurar la descarga del archivo, y que usuarios que no disponen de licencia puedan descargar el archivo, la descarga se realiza a través del archivo *descargar\_archivo.php*, que sirve al cliente archivos sin hacer referencia a la ruta del archivo ni al nombre de éste en el servidor<sup>8</sup>.

Software	Versión Actual	Versión Disponible	
DMXSOFT USB-DMX package LEVEL 1	1.01	1.04	<a href="#">Descargar</a>
Diseño Equipos de Iluminación	1.01	1.01	No dispones de licencia

Fig. 5.5: Tabla de información de versiones disponibles.

<sup>8</sup> Anexo: Referencia 4. Código para servir archivos. Pág. 88.

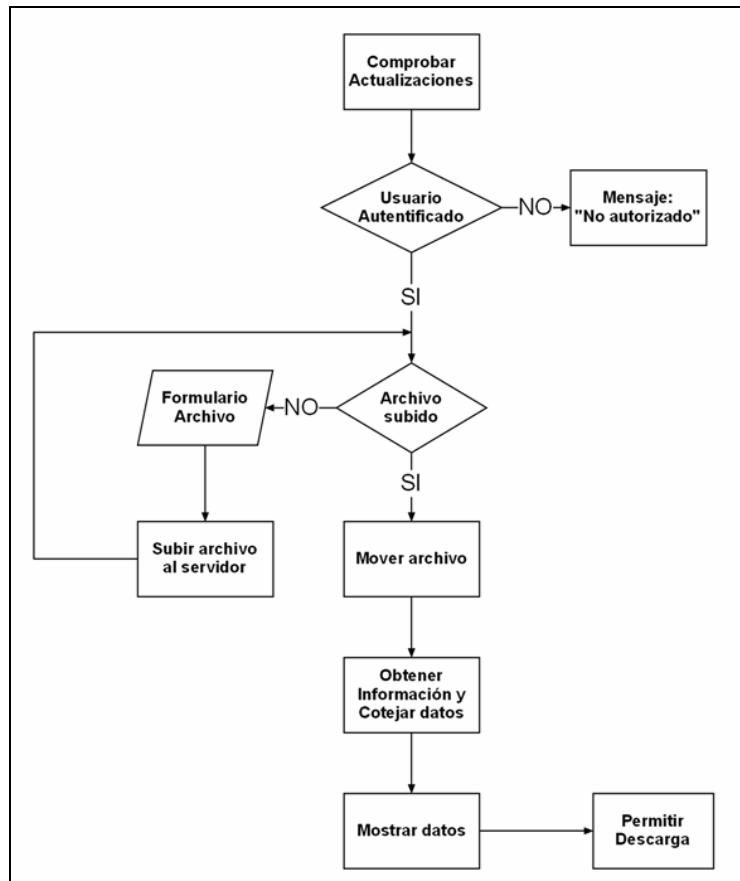


Fig. 5.6: Diagrama de flujo descargar actualizaciones.

### Actualizar lista de Mdf.

Debido a que las funcionalidades de comprobar modelos de funcionamiento disponibles y descargar modelos de funcionamiento comparten gran parte de sus secuencias, se ha decidido implementar estas funcionalidades en una única funcionalidad. En primer lugar, la funcionalidad comprueba los modelos de funcionamiento disponibles para el usuario y luego se agregan a su base de datos aquellos que él elija. El código de la funcionalidad está en el archivo *actualizar\_mdf.php* de la aplicación.

Para comprobar que modelos de funcionamiento tiene instalado el usuario, la aplicación ha de acceder al archivo de base de datos de *Microsoft Access* del usuario. Para realizar la conexión con esta base de datos, el archivo ha de estar subido al servido. Ésta acción se realiza mediante un formulario donde se le pide al usuario que especifique la ruta de acceso al archivo. Cuando el usuario da permiso para subir el archivo, éste es movido de un directorio temporal a un directorio de la aplicación, y se le cambia el nombre (se le añade el identificador de usuario) para que no exista

conurrencia de archivos en el servidor. Una vez en este directorio se accede a la base de datos del cliente para comprobar los modelos que tiene incluidos. Luego se cotejan estos datos y se le muestra al usuario la lista de modelos que dispone la BBDD de la aplicación, y que no están incluidos en la del cliente. Éste puede seleccionar los modelos que desee y se agregan a su base de datos. Finalmente se le sirve al cliente un archivo con su base de datos actualizada, indicándole que deberá sustituir el archivo de la BBDD anterior a la operación por el nuevo archivo.

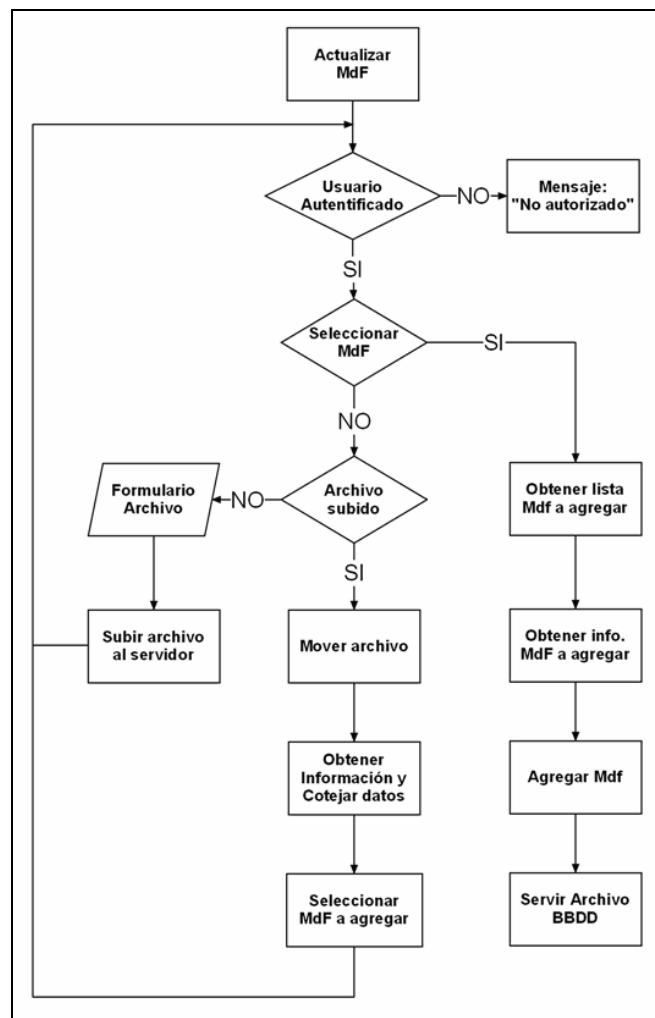


Fig. 6: Diagrama de flujo Actualizar lista Mdf

### 5.2.3. Capa de datos.

En la capa de datos será donde se almacene toda la información de la aplicación: noticias, registro de usuarios, productos software, modelos de funcionamiento etc. Tanto el acceso como la gestión de ésta información se realizará a través de consultas que puedan ser interpretadas por MySQL.

Para almacenar toda esta información se han diseñado una serie de tablas tal y como se describe en el apartado **4.4. ESTRUCTURA DE LA BASE DE DATOS**. Además de estas tablas, existen una serie de tablas que sirven para complementar la funcionalidad que permite mostrar el menú de navegación de la aplicación.

### **Conexión con la base de datos.**

Para que el motor de la aplicación pueda interactuar con la capa de datos, se necesita realizar una conexión con la base de datos de la aplicación. De esta conexión se encarga el archivo *conect.php* de la aplicación. En este archivo se hace uso de funciones que se incluyen en las librerías de *PHP* que permiten la conexión con bases de datos *MySQL*. Cuando la conexión con la BBDD es requerida, la aplicación recurre al código del archivo.

La codificación de esta conexión en un archivo aparte, permite, que modificando unos parámetros, se pueda realizar la conexión con otra base de datos<sup>9</sup>.

### **Consulta y gestión de la información.**

Una vez realizada la conexión, el motor de la aplicación puede acceder a la información de la base de datos. Para realizar este acceso se ha de ejecutar una consulta interpretable por *MySQL*. La ejecución de estas consultas se realiza a través de funciones que se incluyen en las librerías de *PHP*. En el caso que la consulta requiera de un retorno de información, el motor de la aplicación también es capaz de realizarlo mediante funciones específicas de *PHP*<sup>10</sup>.

### **Mostar menú.**

Para poder realizar la funcionalidad que permite mostrar el menú de navegación de la aplicación de forma dinámica, se hace uso de unas tablas de la BBDD diseñadas con este objetivo.

---

<sup>9</sup> Anexo: Referencia 5. Conexión con la base de datos de la aplicación. Pág. 88.

<sup>10</sup> Anexo: Referencia 6: Ejemplos de ejecución de consultas. Pág. 89.

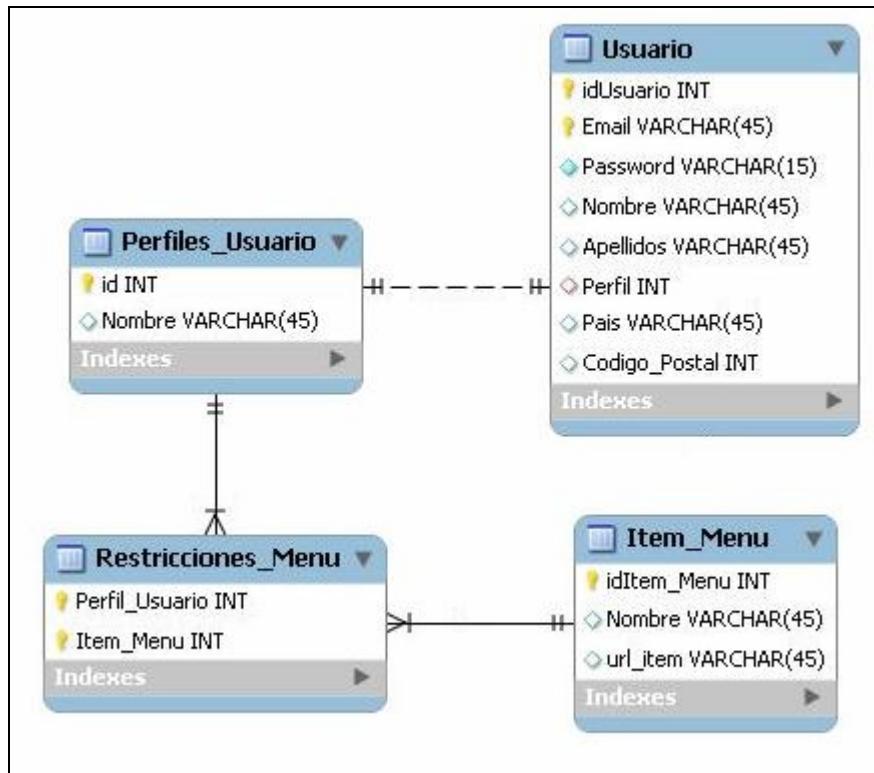


Fig. 5.8: Tablas para mostrar menú.

En la tabla Perfiles\_Usuario almacenaremos todos los perfiles de usuario de la aplicación (asignándole a cada uno un identificador). En la tabla Item\_Menu almacenamos todas las funcionalidades disponibles en el portal. Por último, la tabla Restricciones\_Menu contiene la relación entre los diferentes perfiles de usuario y sus funcionalidades asociadas. Esta estructura, nos permite realizar cambios en las funcionalidades de cada perfil de usuario e incluso añadir funcionalidades a la aplicación o nuevos perfiles de usuario.

Para obtener la lista de funcionalidades del usuario autenticado se realiza la siguiente consulta a la BBDD (donde \$user\_actived es el identificador del usuario)

```

SELECT Item_Menu.Nombre, Item_Menu.url_item
FROM Usuario, Item_Menu, Perfiles_Usuario, Restricciones_Menu
WHERE (Usuario.Perfil = Perfiles_Usuario.id) AND
      (Perfiles_Usuario.id = Restricciones_Menu.Perfil_Usuario) AND
      (Restricciones_Menu.Item_Menu = Item_Menu.idItem_Menu) AND
      (Usuario.idUsuario = $user_actived )
  
```

### 5.3. INTEGRACIÓN DE LAS HERRAMIENTAS DE APOYO.

En el presente proyecto se han integrado una serie de herramientas que permiten cumplir con determinados requerimientos. El uso de estas herramientas ha sido bajo licencias de software libre, lo que permite adaptar la herramienta al proyecto.

*TinyMCE* es un editor de texto basado en *JavaScript* que pretende facilitar, y dar seguridad al texto. Es muy fácil de integrar con los formularios, y completamente personalizable, tanto como en estilos, como en funcionalidades. Es necesario añadir al directorio del proyecto el directorio del editor, que se puede descargar a través de la página WEB de la herramienta ([www.tinymce.kebes.es](http://www.tinymce.kebes.es)). En las funcionalidades que se requiere del editor, se integra el código que lo activa<sup>11</sup>. En la figura 5.9 se muestra el aspecto de un formulario de tipo *textarea* con *TinyMCE* (con una configuración de funcionalidades personalizada).



Fig. 5.9: Formulario *textarea* con *TinyMCE*.

*phpMyAdmin* es una herramienta escrita en *PHP* con la intención de manejar la administración de *MySQL* a través de páginas WEB. Su código se puede descargar de su propia página WEB ([www.phpmyadmin.net](http://www.phpmyadmin.net)) y su integración únicamente requiere de la inserción al servidor de los diferentes archivos que componen la aplicación y de una sencilla configuración.

*WinSCP* es un cliente *SFTP* gráfico para *Windows* que emplea *SSH*. Para su correcto funcionamiento únicamente se requiere la instalación del programa. El archivo de instalación se puede conseguir en la pagina WEB ([www.winscp.net](http://www.winscp.net)) y es gratuito.

<sup>11</sup> Anexo: Referencia 7. Código para el uso de TinyMCE. Pág. 90.

## 6. PRUEBAS.

---

### 6.1. INTRODUCCIÓN.

Una vez finalizada la fase de implementación del proyecto se han realizado una serie de pruebas, con el fin de asegurarnos que se cumplen los objetivos impuestos en la fase de análisis del proyecto.

Estas pruebas nos aseguran el correcto funcionamiento de las funcionalidades, así como aspectos como la seguridad de accesos y la compatibilidad de la aplicación con diferentes navegadores WEB, y son descritas en el actual capítulo.

### 6.2. PRUEBAS DE COMPATIBILIDAD.

Puesto que el acceso a la aplicación se debe realizar mediante un navegador WEB, se ha procedido a realizar una prueba de funcionamiento de la aplicación con los diferentes navegadores con los que se dará uso.

El portal WEB ha sido diseñado e implementado para que funcione tanto con *Microsoft Internet Explorer* como con *Mozilla Firefox*. Por tanto, se han realizado las pruebas de compatibilidad con dichos navegadores junto a otros navegadores de uso emergente:

- Microsoft Internet Explorer 6.0.
- Mozilla Firefox 3.6.
- Opera 9.64.
- Google Chrome 2.0.

El funcionamiento de la aplicación es correcto con estos navegadores, y no se aprecian diferencias de rendimiento entre ellos. El diseño general del portal es con todos los navegadores correcto, pese a que se observan algunas diferencias puntuales en el modo de visualizar determinada información. Estas diferencias se deben a la distinta interpretación entre navegadores de etiquetas del lenguaje *HTML*.

### **6.3. PRUEBAS DE SEGURIDAD.**

A la aplicación acceden usuarios con perfiles distintos y cada perfil de usuario tiene permiso para acceder a funcionalidades distintas. Por esto es necesario comprobar que el control de accesos que se ha diseñado e implementado funciona correctamente.

El menú de acceso a las funcionalidades, muestra para cada perfil de usuario las funcionalidades correspondientes. Además, si se introduce en el navegador una url a una funcionalidad para la que el usuario no tiene permiso, se muestra un mensaje que notifica al usuario el acceso restringido.

### **6.4. PRUEBAS DE INTEGRACIÓN DE SERVICIO.**

Las pruebas de integración de servicio se han realizado integrando la aplicación en dos servidores distintos.

En el primer servidor gratuito, al no contar con las funcionalidades necesarias para conectar con bases de datos de *Microsoft Access*, únicamente se han podido realizar las pruebas de funcionamiento con aquellas funcionalidades que no utilizan dicha conexión. Posteriormente se ha procedido a integrar la aplicación en un servidor que funcione en un entorno local, y que cumpla los requisitos especificados. En este servidor se han podido realizar las pruebas correspondientes, concluyendo el correcto funcionamiento de la aplicación.

## 7. CONCLUSIÓN Y RESULTADOS.

---

---

### 7.1. CONSECUCIÓN DE OBJETIVOS.

El principal objetivo de la aplicación es permitir a los clientes de la empresa la descarga de archivos para la actualización de software y la descarga de modelos de funcionamiento para las herramientas de diseño y control de equipos de iluminación. Para cumplir con este objetivo se han diseñado e implementado las correspondientes funcionalidades que además cumplen con los requerimientos no funcionales requeridos.

Se ha creado también una funcionalidad que permite la gestión de los contenidos que se desean incluir en la aplicación: productos software, actualizaciones de software, modelos de funcionamiento. Además se han implementado las funcionalidades que permiten la visualización de estos contenidos. De esta manera se consigue cumplir con el objetivo de informar a los clientes, o futuros cliente, sobre los servicios que oferta la empresa, mejorando la imagen de ésta.

La aplicación integra un sistema de inserción, modificación y visualización de noticias. Con esto se refuerza el objetivo de mejorar la imagen de la empresa y se potencia el acceso al portal.

Para el correcto funcionamiento de la aplicación se ha creado un sistema de registro y autenticación de usuarios, complementado con el sistema de clasificación de usuarios mediante un perfil de usuario. Esto permite cumplir con los objetivos de seguridad y control de acceso de los usuarios. Esta funcionalidad, añadida a las de

gestión y visualización de contenidos y noticias, permite tener una base de datos actualizada, que era uno de los objetivos del proyecto.

Para cumplir con el objetivo de accesibilidad se ha diseñado e implementado un sistema de menús dinámicos. Este sistema permite la modificación de las funcionalidades asociadas a los distintos perfiles de usuario, lo que incrementa la flexibilidad de la aplicación y facilita las posibles ampliaciones que se realicen.

## **7.2. DESVIACIONES OBSERVADAS.**

El presente proyecto se ha desarrollado en el plazo de tiempo establecido en el apartado de **2.2. Planificación del proyecto**. La distribución de las horas diarias dedicadas al proyecto ha sufrido variaciones en el transcurso de éste, debido a otras actividades y a contratiempos. Pese a esto, se ha respetado promedio de tiempo semanal dedicado al proyecto, flexibilizando las horas de trabajo.

Debido a que no era necesario un control exhaustivo de los plazos de inicio y finalización de las tareas del proyecto, es difícil determinar si se han producido desviaciones en estos plazos. La inexperiencia en proyectos de esta envergadura y complejidad apunta a que si se han producido estas desviaciones, pese que no hayan afectado al transcurso global del proyecto.

## **7.3. LINEAS DE AMPLIACIÓN.**

El presente proyecto, es susceptible a mejoras o ampliaciones de las funcionalidades que lo integran. Una de estas ampliaciones posibles consiste en implantar una tienda virtual, que permitiera a los clientes adquirir los productos software. De esta manera se aumentarían los canales de venta de los productos.

Además de esta tienda, sería interesante incluir un método de adquisición de licencias software. Estas funcionalidades podrían estar vinculadas con el sistema de pago por Internet *PAY-PAL* o similares, lo que haría mucho más accesible para los clientes los servicios que oferta la empresa.

La empresa tiene previsto establecer un sistema de consultoría para terceras empresas instaladoras de servicios para locales de ocio. Este servicio, o funcionalidades relacionadas con éste, podrían ser una futura línea de ampliación del portal WEB.

Como complemento a la base de datos de la aplicación, se podría implementar un sistema de estadísticas, con el fin de obtener datos que ayuden a la empresa en futuros estudios de mercado.

Sería interesante realizar un estudio de posicionamiento WEB. De esta manera, se conseguiría ampliar las posibilidades de captar nuevos clientes.

#### **7.4. VALORACION PERSONAL DE LA EXPERIENCIA.**

En el presente proyecto se han aplicado buena parte de los conocimientos, recursos y métodos de trabajo aprendidos en el transcurso de la carrera universitaria. Este proyecto es un buen resumen de los contenidos asumidos, pero también una buena referencia para evaluar la forma de aplicar esos conocimientos y las capacidades del alumno.

Por otra parte, en muchos casos, incluido el mío, realizar un proyecto similar a éste puede ser una buena primera experiencia en proyectos de envergadura mayor. Además puede suponer el fin de una etapa, y el comienzo de otra que justo acaba de empezar.

# BIBLIOGRAFÍA.

## Libros.

- Dave Shea, The Zen Of Css Design: Visual Inlightenment For The Web. Addison-Wesley, 2005
- Javier Eguíluz Pérez, Introducción a CSS. Librosweb.es (editorial digital), 2009
- Javier Eguíluz Pérez, Introducción a JavaScript. Librosweb.es (editorial digital), 2009.
- Dave W. Mercer y VV.AA. Fundamentos PHP 5. ANAYA, 2005.
- Luke Welling & Laura Thomson, Desarrollo Web con PHP y MySQL. ANAYA, 2003.
- Elizabeth Naramore, Michael K. Glass & Yann Le Scouarnec, Desarrollo Web Con Php, Apache y Mysql. ANAYA, 2004.

## Referencias electrónicas.

- [ [www.ignside.net/man/php/cookies.php](http://www.ignside.net/man/php/cookies.php) ] Apuntes de PHP: Cookies, (20-4-2009).
- [ <http://www.php.net/> ] PHP: Hypertext Preprocessor, (4-4-2009).
- [ <http://www.php-es.com/> ] Manual de PHP, (4-4-2009).
- [ [www.csszengarden.com](http://www.csszengarden.com) ] css Zen Garden: The Beauty in CSS Design, (5-4-2009).
- [ [www.lsi.us.es/cursos/cursophp/](http://www.lsi.us.es/cursos/cursophp/) ] Desarrollo de sitios Web con PHP y MySQL, (10-4-2009).
- [ [www.desarrolloweb.com/articulos/ejemplo-conexion-php-access-dsn.html](http://www.desarrolloweb.com/articulos/ejemplo-conexion-php-access-dsn.html) ] Ejemplo de conexión con base de datos Access en PHP, (11-4-2009).
- [ <http://html.conclase.net/w3c/css1-es.html> ] Hojas de Estilo en Cascada, nivel 1, (5-4-2009).
- [ <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html> ] Manual de CSS, hojas de estilo, (5-4-2009).
- [ <http://dev.mysql.com/doc/refman/5.0/es/numeric-types.html> ] MySQL :: MySQL 5.0 Reference Manual :: 11.2 Tipos numéricos, (10-4-2009).

- [ <http://www.forosdelweb.com/f18/php-access-409842/> ] PHP y Access – Foros del Web, (27-4-2009).
- [ <http://www.desarrolloweb.com/articulos/1054.php> ] Tipos de datos de MySQL, (7-4-2009)
- [ <http://www.webexperto.com/articulos/art/197/forzar-descarga-de-archivos/> ] WebExperto – Forzar descarcga de archivos (PHP), (11-5-2009).
- [ [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador) ] Modelo-Vista-Controlador - Wikipedia, la enciclopedia libre, (20-7-2009).
- [ <http://es.wikipedia.org/wiki/Php> ] PHP - Wikipedia, la enciclopedia libre, (7-8-2009).
- [ <http://es.wikipedia.org/wiki/Html> ] HTML - Wikipedia, la enciclopedia libre, (7-8-2009).
- [ <http://es.wikipedia.org/wiki/Css> ] CSS - Wikipedia, la enciclopedia libre, (7-8-2009).
- [ [http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache) ] Servidor HTTP Apache - Wikipedia, la enciclopedia libre, (7-8-2009).
- [ <http://es.wikipedia.org/wiki/MySql> ] MySQL - Wikipedia, la enciclopedia libre, (7-8-2009).
- [ <http://es.wikipedia.org/wiki/PhpMyAdmin> ] phpMyAdmin - Wikipedia, la enciclopedia libre, (20-7-2009).
- [ <http://www.phpmyadmin.net/> ] phpMyAdmin, (7-8-2009).
- [ <http://winscp.net/eng/docs/lang:es> ] WinSCP :: Introducción, (7-8-2009).
- [ <http://tinymce.moxiecode.com/> ]TinyMCE – Home,. (10-8-2009).

## ANEXO.

### Referencia 1. Especificaciones Servidor.

En el siguiente recuadro se especifican algunas de las características de un servidor valido para la integración de la aplicación obtenido de [www.solo10.com/alojamiento-web/web-hosting-windows-plan-2.php](http://www.solo10.com/alojamiento-web/web-hosting-windows-plan-2.php).

<b>Detalles Técnicos</b>																							
<b>SISTEMA OPERATIVO:</b> :: Microsoft Windows 2003 Server	<b>LENGUAJES &amp; SCRIPTING</b> :: ASP.NET :: ASP (Active Server Pages) :: Soporte CGI Completo (Full) :: PHP4 / PHP5 :: PERL (Active Perl) :: Soporte Javascript y Flash																						
<b>SERVIDOR DE WEB (HTTPD):</b> :: IIS (Internet Information Server) :: Ejecución de scripts :: CGI-BIN con permisos completos :: Soporte para "Extensões Microsoft FrontPage 2000/2002" :: Gestión de Subdominios (Administrados desde panel de control) :: Soporte SSI (Server Side Includes) :: Páginas de Error personalizables	<b>BASE DE DATOS / HERRAMIENTAS</b> :: MS SQL Server Express 2005 :: Base de Datos MS Access :: MySQL :: Administración con phpMyAdmin :: Soporte ODBC Data Sources (es posible crear DSNs para MS SQL Server Express 2005, MS Access™, MySQL, dBBase™, MS Excel™ y Archivos Texto) :: Administración de DSN desde Panel de control																						
<b>SERVIDOR E-MAIL:</b> :: Administrador de Correos (configuración, altas, bajas, modificaciones) integrado al Panel de Control :: Servidor SMTP en LocalHost :: Servidor con soporte POP3 + IMAP :: Acceso Web al Email :: Alias de e-mail :: Autorepuestas e-mail :: Mensaje "vacaciones" (e-mail) :: Reenvío E-mail (forwarding) :: Filtros Anti-Spam (correo no deseado) :: Filtros Anti-Virus para correo a nivel servidor	<b>CUOTA / TRANSFERENCIA</b> :: Seguro manejo interno de cuotas utilizando MS DiskQuotaControl™ :: 800 Mb de espacio en disco :: 25 Gb de transferencia mensual																						
<b>SERVICIO FTP:</b> :: Servidor FTP :: Edición de la Configuración FTP :: Soporte FTP Anónimo :: Ilimitadas actualización FTP de su espacio web	<b>ACCESOS Y PANEL DE CONTROL:</b> :: Panel de Control "PLESK" :: Administrador :: Acceso FULL FTP																						
<b>SEGURIDAD:</b> :: Firewall (Cortafuego) de Protección :: Backups - es posible realizar copias de seguridad desde el panel de control	<b>GESTIÓN DE PERMISOS Y DIRECTORIOS</b> :: Administrador de Permisos integrado al panel de control																						
<b>ESTADÍSTICAS</b> :: Análisis de Estadísticas en Línea con Awstats™ :: Archivos LOG	<b>GESTIÓN DE ARCHIVOS</b> :: Administrador de Archivos integrado al Panel de Control :: Soporte para "Upload" (subida) de archivos desde el Panel de Control.																						
	<b>GESTIÓN DE USUARIOS</b> :: Administrador Completa de Usuarios :: Búsqueda de Usuarios																						
  <b>Win 800 Mb</b>																							
<b>i1 Año! \$135</b> <input data-bbox="1261 707 1372 729" type="button" value="ORDENAR!"/>																							
Características :: Precio / Ordenar :: Detalles Técnicos																							
<table border="0"> <tr> <td><b>✓ Gran Capacidad y Espacio para su web</b></td> <td><b>CARACTERÍSTICAS</b></td> </tr> <tr> <td><b>✓ Muy Fácil de usar y administrar</b></td> <td>:: 800 Mb espacio en disco</td> </tr> <tr> <td><b>✓ Obtenga ILIMITADOS emails con su propio dominio</b></td> <td>:: ILIMITADOS subdominios</td> </tr> <tr> <td><b>✓ Rápido servicio</b></td> <td>:: ILIMITADOS e-mails POP3</td> </tr> <tr> <td><b>✓ Guías paso a paso para subir sus páginas y configurar sus emails</b></td> <td>:: ILIMITADOS alias e-mail (redireccionadores)</td> </tr> <tr> <td></td> <td>:: ILIMITADOS Autorespuestas e-mail</td> </tr> <tr> <td></td> <td>:: 25 Gb. transf. mensual</td> </tr> <tr> <td></td> <td>:: ASP - ASP.NET - PHP - PERL</td> </tr> <tr> <td></td> <td>:: ACCESS - MySQL - Full CGI</td> </tr> <tr> <td></td> <td>:: Panel de Control</td> </tr> <tr> <td></td> <td>:: WebMail para leer su correo</td> </tr> </table>		<b>✓ Gran Capacidad y Espacio para su web</b>	<b>CARACTERÍSTICAS</b>	<b>✓ Muy Fácil de usar y administrar</b>	:: 800 Mb espacio en disco	<b>✓ Obtenga ILIMITADOS emails con su propio dominio</b>	:: ILIMITADOS subdominios	<b>✓ Rápido servicio</b>	:: ILIMITADOS e-mails POP3	<b>✓ Guías paso a paso para subir sus páginas y configurar sus emails</b>	:: ILIMITADOS alias e-mail (redireccionadores)		:: ILIMITADOS Autorespuestas e-mail		:: 25 Gb. transf. mensual		:: ASP - ASP.NET - PHP - PERL		:: ACCESS - MySQL - Full CGI		:: Panel de Control		:: WebMail para leer su correo
<b>✓ Gran Capacidad y Espacio para su web</b>	<b>CARACTERÍSTICAS</b>																						
<b>✓ Muy Fácil de usar y administrar</b>	:: 800 Mb espacio en disco																						
<b>✓ Obtenga ILIMITADOS emails con su propio dominio</b>	:: ILIMITADOS subdominios																						
<b>✓ Rápido servicio</b>	:: ILIMITADOS e-mails POP3																						
<b>✓ Guías paso a paso para subir sus páginas y configurar sus emails</b>	:: ILIMITADOS alias e-mail (redireccionadores)																						
	:: ILIMITADOS Autorespuestas e-mail																						
	:: 25 Gb. transf. mensual																						
	:: ASP - ASP.NET - PHP - PERL																						
	:: ACCESS - MySQL - Full CGI																						
	:: Panel de Control																						
	:: WebMail para leer su correo																						
<b>Precio</b>																							
¿Qué forma de pago prefiere?																							
<b>Pago Anual</b> Setup: GRATIS Unico pago de \$135.00 <input data-bbox="928 1358 1055 1381" type="button" value="ORDENAR!"/>	<b>Pago Mensual</b> Setup: \$10 Pagos mensuales de \$14.00 <input data-bbox="1214 1358 1341 1381" type="button" value="ORDENAR!"/>																						

### Referencia 2. Funciones de PHP para la gestión de BBDD.

A continuación se enumeran y describen algunas de las funcionalidades de PHP para la gestión de base de datos utilizadas en el proyecto.

- **mysql\_connect(servidor, usuario, password):** establece una conexión a un servidor MySQL. Todos los argumentos son opcionales, y si no hay, se asumen los valores por defecto ('localhost', usuario propietario del proceso del servidor, password vacío).
- **mysql\_select\_db(bbdd):** establece la base activa que estará asociada con el identificador de enlace especificado. Si no se especifica un identificador de

enlace, se asume el último enlace abierto. Si no hay ningún enlace abierto, la función intentará establecer un enlace como si se llamara a `mysql_connect()`. Todas las consultas posteriores harán referencia a esta base de datos.

- **`mysql_query(consulta, id_enlace)`** envía una consulta (a la base de datos activa en el servidor asociado con el *id\_enlace* dado).
- **`mysql_fetch_object(int id_resultado)`:** Devuelve un objeto con las propiedades que corresponden a la última fila extraída, o FALSE si no quedan más filas.
- **`mysql_num_rows(id_resultado)`:** regresa el número de filas en un resultado. Este comando es valido solo para las sentencias SELECT.

### Referencia 3. Código para la gestión de cookies.

En el siguiente recuadro se muestra el código del archivo `cookies_login.php`, utilizado en la funcionalidad de autenticarse, y necesario para la gestión de *cookies*.

```
<?php
//Archivo para guardar las cookies relacionadas con la
autentificacion de usuarios

$user = $_POST[ "usuario" ] ; //obtenemos el nombre de usuario del
                                formulario
$pass = $_POST[ "pass" ] ; //obtenemos el password del formulario

if ($user != NULL && $pass !=NULL){
    if($_POST[ 'recordar' ]){
        //se establecen las cookies a 1 día
        setcookie( "usuario" , $user , time() +3600*24 );
        //se establecen las cookies a 1 día encriptando el
        password con la función con SHA1
        setcookie( "pass_cookie" , SHA1($pass) , time() +3600*24 );

    }else{
        //se establecen las cookies a 1 hora
        setcookie( "usuario" , $user , time() +3600 );
        //se establecen las cookies a 1 hora encriptando el
        password con la función con SHA1
        setcookie( "pass_cookie" , SHA1($pass) , time() +3600 );
    }
}
header( "location: index.php" );

?>
```

## Referencia 4. Código para servir archivos.

En el siguiente recuadro se muestra el código del archivo *descargar\_archivo.php*, que se encarga de servir al usuario un archivo de manera segura.

```
<?php
//obtenemos el nombre del archivo a descargar
$file = $_POST["id"];
//preparamos la ruta del archivo a descargar
$path = "files_data";
$enlace = $path ."/".$file;
//servimos al usuario el archivo
header ("Content-Disposition: attachment; filename=".$file."\n\n");
header ("Content-Type: application/octet-stream");
header ("Content-Length: ".filesize($enlace));
readfile($enlace);
?>
```

## Referencia 5. Conexión con la base de datos de la aplicación.

En el siguiente recuadro se muestra el código del archivo *conect.php*, necesario para la conexión con la base de datos de la aplicación.

```
<?php
$host_conect="localhost"; //especificamos servidor para la conexión
$user_conect="root"; //especificamos usuario para la conexión
$pass_conect=""; //especificamos password para la conexión
$bdd_conect="2142563"; //especificamos nombre de la bbdd a conectar

//realizamos la conexión con el servidor de MySQL
$link_conect = mysql_connect($host_conect,$user_conect,$pass_conect);
if( !$link_conect){
    die('No se puede conectar');
} else {
    //realizamos la conexión con al BBDD
    if( !mysql_select_db($bdd_conect) ){
        die( 'No se puede usar la bbdd: ' . $bdd_conect );
    }else{
        $conect='OK';
    }
}
?>
```

## Referencia 6: Ejemplos de ejecución de consultas.

A continuación se muestra ejemplos de la ejecución de consultas, similares a las utilizadas en la aplicación.

### Visualizar información.

```
<?php
//preparamos la consulta a ejecutar.
$query_noticias = "SELECT *
    FROM `Noticia`
    ORDER BY Noticia.Fecha DESC";

//ejecutamos la consulta, obteniendo un identificador de resultado
$result_noticias = mysql_query( $query_noticias, $link_conect );

//almacenamos la información, recorriendo una por una las líneas del
//resultado
while ($row = mysql_fetch_object($result_noticias)) {

    //mostramos la información
    echo $row->Titular;
    echo $row->Texto;
}
?>
```

### Insertar información.

```
<?php
//obtenemos la información a insertar del formulario
$titular = $_REQUEST[ "titular" ];
$texto = $_REQUEST[ "texto" ];

$fecha = date ( "Y-m-d" ); // Fecha actual
//preparamos la consulta de insercción de información
$query_insertar_noticia = "INSERT INTO Noticia (Titular, Texto)
    VALUES ($titular, $texto)";
//ejecutamos la consulta de insercción.
$result_insertar_noticia = mysql_query($query_insertar_noticia,
    $link_conect);
?>
```

### Modificar información.

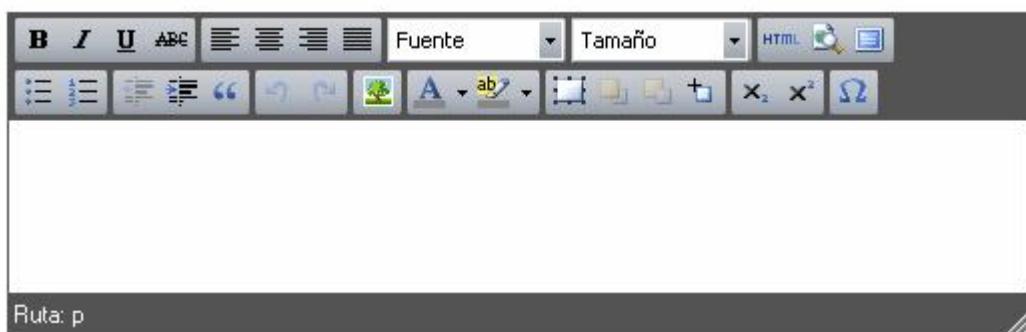
```
<?php
//obtenemos valores a modificar
$titular = $_REQUEST[ "titular" ];
$texto = $_REQUEST[ "texto" ];
//obtenemos identificador de noticia a modificar
$id_Noticia = $_REQUEST[ "noticia" ];

//preparamos la consulta de modificación
$query_editar_noticia = "UPDATE Noticia
    SET Titular = '$titular', Texto = '$texto'
    WHERE idNoticia = $id_Noticia";
//ejecutamos la consulta
$result_editar_noticia = mysql_query( $query_editar_noticia,
    $link_conect );
?>
```

## Referencia 7. Código para el uso de TinyMCE.

A continuación se muestra el código necesario para activar la herramienta *TinyMCE*, que debe ser incluido en el *<head>* del archivo. De este modo los *textarea* del archivo incorporan la herramienta.

```
<!-- incluimos el directorio de TinyMCE en la aplicación -->
<script type="text/javascript"
src=".//tiny_mce/tiny_mce.js"></script>
<!-- configuramos TinyMCE -->
<script type="text/javascript">
tinyMCE.init({
    // Opciones Generales (Idioma, tema,plugins)
    mode : "textareas",
    language : "es",
    theme : "advanced",
    skin : "o2k7",
    skin_variant : "black",
    plugins :
        "safari,pagebreak,style,layer,table,save,advhr,advimage,
        advlink,emotions,iespell,inlinepopups,insertdatetime,prev
        iew,media,searchreplace,print,contextmenu,paste,direction
        ality,fullscreen,noneditable,visualchars,nonbreaking,xhtm
        lextras,template",
    // Definimos las funcionalidades a insertar
    theme_advanced_buttons1 :
        "bold,italic,underline,strikethrough,|,justifyleft,
        justifycenter,justifyright,justifyfull,fontselect,fontr
        eselect,|,code,preview,fullscreen",
    theme_advanced_buttons2 :
        "bullist,numlist,|,outdent,indent,blockquote,|,undo,redo,
        |,image,|,forecolor,backcolor,|,insertlayer,moveforward,m
        ovebackward,absolute,|,sub,sup,|,charmap",
    theme_advanced_toolbar_location : "top",
    theme_advanced_toolbar_align : "left",
    theme_advanced_statusbar_location : "bottom",
    theme_advanced_resizing : true,
    template_replace_values : {
        username : "Some User",
        staffid : "991234"
    }
});
</script>
```



Joan Alsina Morillo

Sabadell, septiembre de 2009