



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



PROYECTO FINAL DE CARRERA

Desarrollo de un sitio web para una empresa

CÓDIGO P.F.C.: DISCA-265

ALUMNO: Sqalli Houssaini, Younes

DIRECTOR: Houcine Hassan Mohamed

Índice

1. Introducción.....	5
1.1 <i>Motivaciones del proyecto</i>	5
1.2 <i>Presentación y objetivos</i>	5
1.3 <i>Método de desarrollo</i>	7
1.4 <i>La tecnología ASP.NET</i>	8
1.5 <i>Organización de la memoria</i>	9
2. Especificación de requisitos.....	11
2.1 <i>Introducción</i>	11
2.1.1 <i>Propósito</i>	11
2.1.2 <i>Ámbito</i>	11
2.1.3 <i>Definiciones, acrónimos y abreviaturas</i>	11
2.1.4 <i>Referencias</i>	12
2.1.5 <i>Visión global</i>	12
2.2 <i>Descripción general</i>	13
2.2.1 <i>Perspectiva del producto</i>	13
2.2.2 <i>Funcionalidad del producto</i>	13
2.2.3 <i>Características de los usuarios</i>	14
2.2.4 <i>Restricciones generales</i>	14
2.3 <i>Requisitos específicos</i>	14
2.3.1 <i>Requisitos comunes de los interfaces</i>	14
2.3.1.1 <i>Interfaces de usuario</i>	14
2.3.1.2 <i>Interfaces de hardware</i>	15
2.3.1.3 <i>Interfaces de software</i>	15
2.3.1.4 <i>Interfaces de comunicación</i>	15
2.3.2 <i>Requisitos funcionales</i>	15
2.3.3 <i>Requisitos no funcionales</i>	19
2.3.3.1 <i>Requisitos de rendimiento</i>	19
2.3.3.2 <i>Mantenimiento</i>	19
2.3.3.3 <i>Limitaciones Hardware</i>	19
3. Análisis.....	20
3.1 <i>Diagrama de clases</i>	20
3.2 <i>Diagrama de casos de uso</i>	22
3.3 <i>Diagramas de secuencia</i>	25
4. Diseño	28
4.1 <i>Capa de presentación</i>	28
4.1.1 <i>Zona pública:</i>	29

4.1.2 Zona de administración:.....	31
4.2 Capa de negocio.....	37
4.3 Capa de persistencia	39
5. Implementación	44
5.1 Detalles de la implementación.....	44
5.2 Tecnologías usadas	50
5.2.1 Tecnologías y lenguajes.....	50
5.2.2 Soporte.....	53
6. Pruebas	55
7. Conclusiones	58
8. Bibliografía	59
9. Anexo.....	60
9.1 Crear una base de datos en visual studio:.....	60

Índice de figuras

Figura 1. Fases de desarrollo y documentación generada	8
Figura 2. Diagrama de clases	21
Figura 3. Actores de la aplicación	22
Figura 4. Caso de uso, gestión categoría	23
Figura 5. Caso de uso, gestión subcategoría.....	23
Figura 6. Caso de uso, gestión producto.....	23
Figura 7. Caso de uso, gestión marca	24
Figura 8. Caso de uso: iniciar sesión	24
Figura 9. Caso de uso cliente	24
Figura 10. Diagrama de secuencia iniciar sesión	25
Figura 11. Diagrama de secuencia añadir producto	26
Figura 12. Diagrama de secuencia modificar producto	26
Figura 13. Diagrama de secuencia eliminar producto	27
Figura 14. Arquitectura de tres capas.....	28
Figura 15. Página principal de la empresa	30
Figura 16. Página para iniciar una nueva sesión de administrador	32
Figura 17. Página principal del administrador	33
Figura 18. Página correspondiente a Categoría.....	34
Figura 19. Página correspondiente a Subcategorie	35
Figura 20. Página correspondiente a Producto	37
Figura 22. Página correspondiente a marca	37
Figura 23. Componentes por nivel de capa	38
Figura 24. Aspecto de dos controles FileUpload	47
Figura 25. Resultado de la utilización del control DataGridView	48
Figura 26. Ejemplo de la utilización del control DataList.....	50
Figura 27. Visualización de la página principal en Internet Explorer.....	55
Figura 28. Visualización de la página principal en Mozilla Firefox	56
Figura 29 . Visualización de la página principal en Google chrome.....	57

1. Introducción

1.1 Motivaciones del proyecto

La principal razón para la elección de este proyecto fin de carrera ha sido poner en práctica algunos conocimientos adquiridos en diversas asignaturas a lo largo de la carrera (bases de datos, programación orientada a objetos, ingeniería de software, ingeniería requerimientos...), además de aprender unas tecnologías nuevas como es .NET, el lenguaje de programación C# o el lenguaje de marcado HTML.

Desarrollaremos una aplicación software relativamente sencilla (un sitio Web) que nos permitirá orientar nuestros conocimientos a un punto de vista completamente práctico y aplicar todo lo que hemos aprendido de forma global y conjunta a lo largo del proyecto.

1.2 Presentación y objetivos

Internet se ha convertido en una de las herramientas de comunicación más potentes que existen actualmente a nivel mundial, el número de sus usuarios crece de manera rápida y las ventajas que presenta son muy variadas, eso hace que para todo negocio o empresa sea prácticamente una obligación tener una página web, una forma fácil y sencilla de dar a conocer los servicios y posibilidades que dispone la empresa.

Entre las muchas ventajas de tener una página web para una empresa podemos citar las siguientes:

- Un sitio le da más personalidad, seriedad y profesionalismo a una empresa.
- Un sitio es la publicidad más económica en relación al tiempo de publicación, 365 días del año, las 24 hs del día.
- Es la forma más económica de exponer productos. Los catálogos en papel y tinta pueden costar sumas elevadas, en cambio teniendo un sitio Web es cuestión de dar la dirección y crear un catálogo on-line.
- Si los competidores del cliente tienen un sitio, debe tener uno propio para tener mayor seriedad. Si sus competidores no lo tienen, debe tener uno propio para aventajar a sus competidores.

- Un sitio Web es un punto de venta mundial disponible en todo momento. Los costos de un local pueden ascender a varios miles por mes, en cambio los costos de un sitio en relación con sus ganancias son ínfimos.

Este proyecto fin de carrera (pfc) de la titulación Ingeniería Informática tiene como objetivo el desarrollo de un sitio Web para una empresa que se dedica a vender e instalar productos informáticos (ordenadores, impresoras, discos duros...) y productos de vigilancia. El sitio Web sólo ofrecerá información sobre los productos, no es una solución de comercio electrónico donde el cliente pueda realizar sus compras online, esta vía se deja abierta para posibles ampliaciones de funcionalidades.

Entre los productos de vigilancia que ofrece la empresa podemos encontrar cámaras fijas, giratorias, digitales, analógicas, equipadas con un zoom o un intensificador de luz, etc.

Con la ayuda de estos productos se puede grabar imágenes de vídeo de alta resolución en tiempo real, vigilar cualquier sitio a distancia vía internet desde un ordenador o un teléfono móvil, visualizar hasta 64 cámaras desde diferentes sitios en un sólo ordenador, etc.

Entre los servicios que ofrece la empresa podemos encontrar:

- Instalación de Sistemas de Alarmas
- Instalación de Sistemas de Vigilancia y Control
- Instalación y mantenimiento de equipos informáticos

El principal objetivo de nuestra página web es transmitir una imagen de modernidad a los clientes ofreciéndoles la posibilidad de poder consultar la mayor cantidad de información y servicios que ofrece la empresa desde cualquier lugar y sin la necesidad desplazarse físicamente.

La aplicación desarrollada pretende también ser un escaparate para nuevos compradores y ayudará a buscar clientes potenciales que no conocían la empresa o no le daban el peso que realmente tiene.

Diseñaremos la página web para cubrir las necesidades tanto de los vendedores como de los clientes, tomando en cuenta en ambos casos un concepto gráfico,

dinámico e interactivo con el usuario, También consideraremos la facilidad de uso, que el interfaz sea agradable y que el cliente pueda realizar una navegación de forma sencilla y eficiente.

Para llevar a cabo este proyecto implementaremos dos zonas principales:

- Zona de usuarios: cualquier usuario podrá visitar la web e informarse sobre la empresa, su localización, los productos ofrecidos, contactar con la empresa, etc. Todo mediante un área común para todos los usuarios que será la parte pública de la web.
- Zona de administración: Esta zona estará reservada para el administrador del sitio web que tendrá la posibilidad de gestionar todo el contenido de la página añadiendo o quitando información de nuestra base de datos de forma rápida y sencilla, desde cualquier ordenador conectado a internet, sólo tiene que introducir un nombre de usuario y una contraseña.

Por su estructura y diseño, el sitio Web que desarrollaremos podría ser válido para cualquier empresa del sector, sólo hay que adaptarlo a sus necesidades y características principales.

1.3 Método de desarrollo

Puesto que vamos a desarrollar un tipo de proyecto que no es novedoso y que disponemos de todas las especificaciones desde el principio, el enfoque o método más adecuado que podemos seguir es el que corresponde con el ciclo de vida clásico (también llamado encascada), este enfoque metodológico tiene la ventaja de que su planificación es sencilla y la calidad del producto resultante es alta.

Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido a día de hoy.

En la siguiente figura vemos las fases y la documentación que ha generado cada una de ellas.

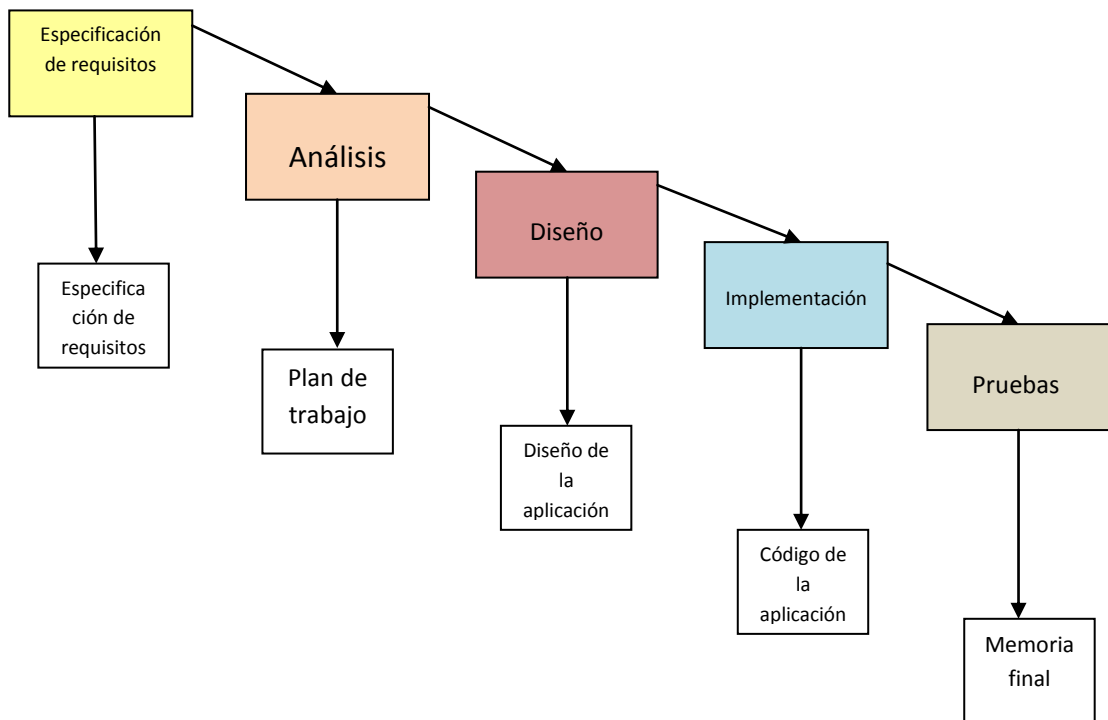


Figura 1. Fases de desarrollo y documentación generada

1.4 La tecnología ASP.NET

Para el desarrollo de la aplicación se ha elegido la tecnología ASP.NET frente a otras tecnologías como por ejemplo PHP, por las siguientes razones:

- ASP.NET tiene un tipado fuerte de datos frente a PHP, esto permite compilar el código y verificar errores antes de su puesta en producción, mientras que en PHP es más difícil saber si se está utilizando un tipo de datos incorrecto.
- Con ASP.NET se puede trabajar con varios lenguajes de programación mientras que PHP sólo permite trabajar con un tipo de lenguaje y esto restringe el número de desarrolladores.
- El lenguaje de programación que utilizaremos será C# que es un lenguaje orientado a objetos, con todas las ventajas que ello implica, mientras que si bien PHP en sus últimas versiones es orientado a objetos, sus librerías no lo son.
- ASP.NET como es una tecnología relativamente novedosa se ha desarrollado intentando solventar las carencias que se observan en PHP o Java.

- ASP.NET es mucho ms rápido de desarrollar que PHP debido al patrón de diseño utilizado de ASP.NET, frente al patrón clásico utilizado por PHP o las antiguas versiones de ASP.NET.

1.5 Organización de la memoria

Esta memoria contiene toda la información necesaria para el desarrollo del proyecto, esta información nos determina los pasos claves para llevar a buen término los objetivos planteados.

La estructura de la memoria se divide en siete puntos:

- **Introducción:** se hace una presentación del proyecto y se explican los objetivos que se pretenden lograr mediante el mismo.
- **Especificación de requisitos (ERS):** Permite a los clientes describir claramente lo que se desea obtener mediante un cierto software y a los desarrolladores entender qué quiere exactamente el cliente.
Con una buena gestión de requisitos se logra crear software de buen rendimiento que satisface realmente las necesidades del usuario.
- **Análisis:** Se empieza por documentarse sobre las posibles soluciones a aplicar, tanto a nivel técnico como gráfico para llegar a una solución idónea. Luego se preparan los diagramas y por último se genera un prototipo de la interfaz gráfica.
- **Diseño:** Es un proceso que extiende, refina y reorganiza los aspectos detectados en el proceso de análisis, para generar una especificación rigurosa del sistema de información siempre orientada a la obtención de la solución del sistema software.
- **Implementación:** Se muestra el desarrollo del software a partir de la ERS, análisis y diseño basado en una arquitectura multicapa (en concreto, organizado en tres niveles: interfaz, lógico y de persistencia).

- Pruebas: Se comprueba si los objetivos inicialmente propuestos han sido alcanzados y si la página web funciona correctamente en diferentes entornos (diferentes navegadores). Además, se comprueba el correcto funcionamiento de la aplicación mediante la validación de las distintas partes que la componen.
- Conclusiones: Se exponen las diferentes ideas extraídas tanto en el ámbito personal como a nivel técnico tras haber finalizado el proyecto.
- Bibliografía: Se detallan todas las fuentes de información consultadas para la elaboración del proyecto, tanto de tipo impreso (libros, revistas u otros proyectos de fin de carrera) como de tipo electrónico (pagina web, ebook...).

2. Especificación de requisitos

La especificación de requisitos de software (ERS) es una de las tareas más importantes en el ciclo de vida del desarrollo de software, puesto que en ella se determinan los “planos” de la nueva aplicación.

En cualquier proyecto software los requisitos son las necesidades del producto que se debe desarrollar. Por ello, en la fase de especificación de requisitos se deben identificar claramente estas necesidades y documentarlas.

En este proyecto vamos a seguir el estándar IEEE 830.

2.1 Introducción

2.1.1 Propósito

El propósito de esta especificación de requisitos es dar una descripción general del proyecto y exponer sus funcionalidades y los requisitos y restricciones principales que debe cumplir. A partir de esta especificación podremos realizar un análisis y un diseño que se ajusten a los requisitos aquí expuestos.

Esta especificación de requisitos va dirigida al realizador del proyecto. A posteriori puede servir como una justificación de algunas decisiones tomadas al desarrollar este proyecto fin de carrera.

2.1.2 Ámbito

El título del proyecto realizado es “Desarrollo de un sitio web para una empresa” y como su nombre lo indica se trata de realizar una página web para permitir a los clientes de la empresa consultar la información deseada desde internet y sin la necesidad de desplazarse físicamente.

2.1.3 Definiciones, acrónimos y abreviaturas

Definiciones:

- Sitio web: Un sitio web es una colección de páginas web relacionadas y comunes a un dominio de internet o subdominio en la World Wide Web en Internet.

- Pagina web: Una página web es un documento HTML/XHTML que es accesible generalmente mediante el protocolo HTTP de Internet.
- Web: La World Wide Web (WWW) o Red informática mundial es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

Acrónimos:

- IEEE: Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos).
- HTML: HyperTextMarkup Language (lenguaje de marcado de hipertexto).
- CSS: Cascading Style Sheets (Hojas de estilo en cascada).
- ASP: Active Server Pages.
- ERS: especificación de requisitos de software.
- SQL: structured query language (lenguaje de consulta estructurado).
- HTTP: Hypertext Transfer Protocol (en español protocolo de transferencia de hipertexto).

2.1.4 Referencias

[1] El estándar IEEE Std 830-1998.

[2] Ejemplos de otras memorias.

2.1.5 Visión global

La parte de la especificación de requisitos está conformada de tres secciones que son la Introducción, la Descripción general y los Requisitos Específicos. En esta primera sección se procura proporcionar una visión general de lo que es el documento de especificación de requisitos. En la segunda sección se da una descripción general del sistema a construir, para conocer sus funciones principales, los datos requeridos, y sus restricciones, entre otras cosas que afecten su desarrollo, aunque no se entra en los detalles de cada uno de estos factores y, por último, la tercera sección Debe contener una de los requisitos que debe cumplir el sistema a desarrollar.

El nivel de detalle de los requisitos debe ser el suficiente para diseñar un sistema que satisfaga los requisitos y las pruebas puedan determinar si éstos se satisfacen.

2.2 Descripción general

2.2.1 Perspectiva del producto

La aplicación desarrollada en este proyecto es independiente y no forma parte de ningún sistema mayor.

El producto está destinado a cualquier persona conectada a Internet y que busca información sobre las nuevas tecnologías relacionadas a los sistemas de vigilancia, cámaras de seguridad, etc.

2.2.2 Funcionalidad del producto

Nuestro sitio web se compone de dos zonas, en cada una de ellas se pueden realizar acciones bien distintas:

Funcionalidades en la zona de usuarios:

- Consultar todos los productos ofrecidos y la información relacionada con cada uno de ellos (precio, características, disponibilidad, etc.).
- Visualizar las imágenes de los productos.
- Consultar las ofertas existentes de algunos productos con descuento.
- Informarse sobre la empresa (dirección, teléfono...).
- Contactar con la empresa.

En la zona de administración se realizan todas las operaciones para el mantenimiento y la actualización de la base de datos:

- Identificación del administrador mediante un nombre de usuario y una contraseña.
- Insertar, modificar y eliminar categorías.
- Insertar, modificar y eliminar subcategorías.
- Insertar, modificar y eliminar productos.
- Insertar nuevas marcas o eliminar marcas existentes.
- Cambiar los datos de un administrador.

2.2.3 Características de los usuarios

Hay dos tipos de usuarios que pueden acceder al sitio Web:

- Clientes: Cualquier persona con conexión a internet puede acceder a la página Web para consultar toda la información ofrecida en la misma.
- Administradores: Tienen acceso a la zona de administración y la posibilidad de gestionar todo el contenido de la base de datos, no necesitan ninguna formación técnica, sólo deben tener unos conocimientos básicos en la informática.

2.2.4 Restricciones generales

- C# es el lenguaje de programación utilizado para el desarrollo de la aplicación. Para la elaboración de las páginas Web se utilizara el lenguaje de marcado HTML.
- Los usuarios podrán acceder a la aplicación desde diferentes tipos de máquinas (PC, Mac, etc.) mediante cualquier navegador de Internet.
- Se utiliza una base de datos relacional.
- Para proteger los datos, solo los administradores pueden acceder a la base de datos después de identificarse introduciendo un nombre de usuario y una contraseña.

2.3 Requisitos específicos

2.3.1 Requisitos comunes de los interfaces

2.3.1.1 Interfaces de usuario

La interfaz gráfica es la cara visible de la aplicación y es la forma de presentar el producto a los usuarios, por eso, la interfaz gráfica tiene que ser muy intuitiva y fácil de utilizar tanto por los clientes como por los administradores.

Los usuarios tienen que ver en pantalla toda la información necesaria, y la interacción se basará en selecciones de tipo menú y en acciones físicas sobre elementos de código visual como pueden ser los iconos, botones, imágenes, barras de desplazamiento...

Las entradas al sistema se realizarán desde los dispositivos de entrada habituales que son el teclado y el ratón.

2.3.1.2 Interfaces de hardware

Para acceder al sitio web de la empresa, los usuarios (sean clientes o administradores) necesitarán un ordenador con conexión a Internet o un teléfono móvil con tecnología 3G/4G.

La calidad de visualización de la página Web dependerá de la tarjeta gráfica del ordenador y de la resolución de pantalla que utiliza el usuario.

2.3.1.3 Interfaces de software

En el lado del cliente, cualquier navegador de Internet servirá para visualizar la página Web, pero se recomienda instalar versiones más actuales (Internet Explorer 6.0 o superior). La aplicación funcionará en cualquier sistema operativo.

En el lado del servidor, puesto que la tecnología usada es ASP.NET, es necesario instalar el servidor web Internet Information Server (IIS) y se recomienda instalar el servidor de base de datos SQL Server 2005.

2.3.1.4 Interfaces de comunicación

Como se ha mencionado en un punto anterior, una conexión a Internet es imprescindible para poder visualizar las páginas Web en el navegador, para que esta conexión funcione, el usuario debe tener instalado en su equipo los protocolos TCP/IP y el protocolo HTTP.

2.3.2 Requisitos funcionales

En esta parte, vamos a definir las acciones fundamentales que debe realizar la aplicación al recibir información, procesarla y producir resultados.

Administrador:

-Autenticación:

- Entrada:
Nombre de usuario y contraseña.

- Proceso:
A partir de los datos introducidos, se comprueba que el usuario está registrado en la base de datos como administrador.
Si es así, el usuario podrá acceder a la zona de administración para gestionar la base de datos.
- Salida:
Nada.

-Gestión de categorías:

Añadir:

- Entrada:
Nombre de la categoría y descripción.
- Proceso:
Insertar un registro con la información de la nueva categoría en la base de datos.
- Salida:
Mensaje de confirmación del alta de la categoría.

Modificación:

- Entrada:
Nueva descripción de la categoría.
- Proceso:
Actualizar el registro de la categoría (con la información introducida) en la base de datos.
- Salida:
Mensaje de confirmación de la actualización de la categoría.

Eliminación:

- Entrada:
No se introduce ninguna información, sólo se selecciona la categoría que se quiere eliminar.
- Proceso:
Se comprueba que no existen subcategorías que pertenecen a esta categoría. Si es así, se borra el registro de la categoría seleccionada de la base de datos.
- Salida:
Confirmación del borrado del registro.

-Gestión de sub-categorías:

Añadir:

- Entrada:

Nombre, descripción y una de las categorías existentes en la base de datos.

- Proceso:
Insertar un registro con la información de la nueva subcategoría en la base de datos.
- Salida:
Mensaje de confirmación del alta de la subcategoría.

Modificación:

- Entrada:
Nueva descripción de la subcategoría.
- Proceso:
Actualizar el registro de la subcategoría en la base de datos.
- Salida:
Mensaje de confirmación de la actualización de la subcategoría.

Eliminación:

- Entrada:
No se introduce ninguna información, sólo se selecciona la subcategoría que se quiere eliminar.
- Proceso:
Si no hay ningún producto que pertenece a esta subcategoría, se borra el registro de la subcategoría seleccionada de la base de datos.
- Salida:
Confirmación del borrado del registro.

-Gestión de productos:

Añadir:

- Entrada:
Código, nombre, descripción, precio, ruta de la foto del producto, marca, subcategoría, ficha técnica, es nuevo (sí/no), está en oferta (sí/no).
- Proceso:
Insertar un registro con la información del nuevo producto en la base de datos.
- Salida:
Mensaje de confirmación del alta del producto.

Modificación:

- Entrada:
Nombre, nueva descripción del producto, está en oferta (sí/no).
- Proceso:
Actualizar el registro del producto (con la información introducida) en la base de datos.

- Salida:
Mensaje de confirmación de la actualización del producto.

Eliminación:

- Entrada:
No se introduce ninguna información, sólo se selecciona el producto que se quiere eliminar.
- Proceso:
Borrar el registro del producto seleccionado de la base de datos.
- Salida:
Confirmación del borrado del registro.

-Gestión de marcas:**Añadir:**

- Entrada:
Nombre de la marca.
- Proceso:
Insertar un registro con la información de la nueva marca en la base de datos.
- Salida:
Mensaje de confirmación del alta.

Eliminación:

- Entrada:
No se introduce ninguna información, sólo se selecciona la marca que se quiere eliminar.
- Proceso:
Si no hay ningún producto en la base de datos con esta marca, se borra el registro de la marca seleccionada de la base de datos.
- Salida:
Confirmación del borrado del registro.

2.3.3 Requisitos no funcionales

2.3.3.1 Requisitos de rendimiento

El único requisito para que la aplicación funcione de manera correcta y que los tiempos de respuesta no sean largos, es tener una buena conexión a Internet y un ordenador con prestaciones aceptables.

2.3.3.2 Mantenimiento

El administrador será el encargado del mantenimiento del sitio Web, lo hará de forma periódica actualizando la información, las ofertas de productos, las imágenes y ampliando las funcionalidades de la aplicación si es necesario.

2.3.3.3 Limitaciones Hardware

Se recomienda utilizar un equipo informático de prestaciones altas para soportar un mayor número de usuarios simultáneamente conectados.

3. Análisis

Después de describir las funcionalidades de la aplicación y algunas de sus características en el apartado de especificación de requisitos, intentaremos realizar un modelado conceptual orientado a objetos. Los modelos obtenidos en esta etapa nos permitirán entender mejor el funcionamiento de nuestra aplicación, servirán para explicar con detalle cuales son los conceptos relevantes en la aplicación y como se relacionan entre ellos y nos serán de gran ayuda en las próximas etapas de diseño e implementación.

Para llevar a cabo el análisis de la aplicación utilizaremos el lenguaje de modelado UML (Unified Modeling Language) que es un lenguaje gráfico para visualizar, especificar, construir y documentar proyectos, dispone de diferentes tipos de diagramas, los cuales muestran varios aspectos de las entidades representadas.

En nuestro análisis describiremos el diagrama de clases y los diagramas de casos de uso que veremos a continuación.

3.1 Diagrama de clases

Es el diagrama principal a la hora de analizar y diseñar aplicaciones Web. El diagrama de clases representa las clases más importantes que componen la aplicación y las relaciones entre ellas.

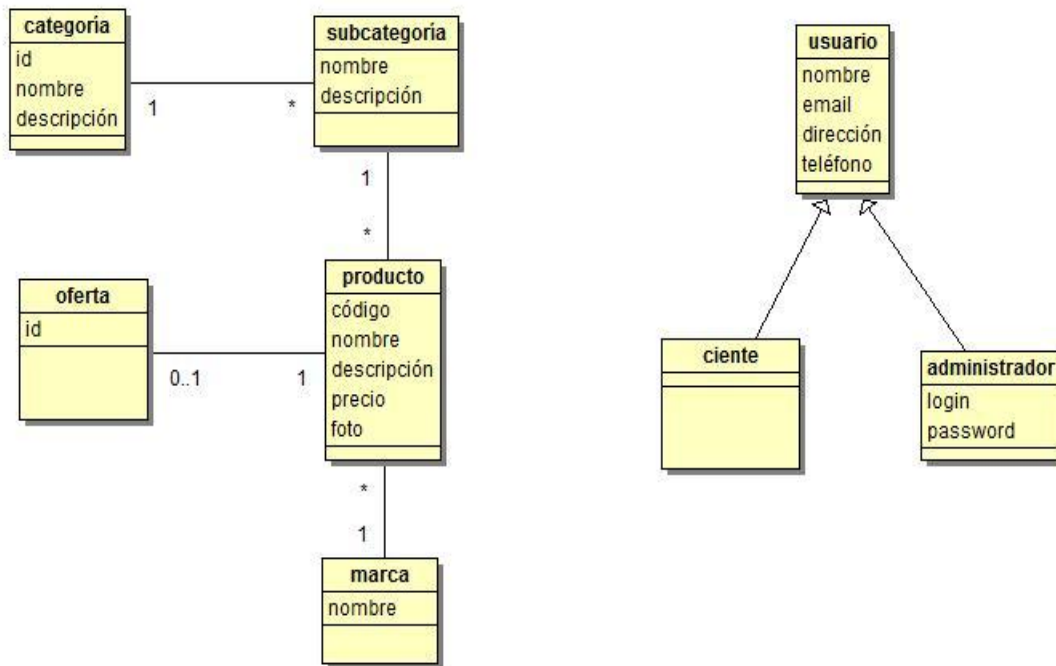


Figura 2. Diagrama de clases

A continuación daremos una pequeña descripción de las diferentes clases que aparecen en la figura 2.

Usuario: un usuario de la aplicación puede ser un cliente normal o un administrador, cada uno de ellos tiene unas funcionalidades diferentes que puede ejercer en la aplicación.

Administrador: deriva de la clase usuario, un administrador tiene que estar registrado en la base de datos y sólo puede acceder a su área introduciendo un nombre de usuario y una contraseña válidos. El administrador se encarga principalmente de gestionar la base de datos.

Cliente: deriva de la clase usuario, representa a cualquier persona conectada a internet y que quiere consultar el catálogo de productos ofrecidos por la empresa, no necesita identificarse para poder acceder a la página principal de la empresa.

Producto: la clase producto almacena toda la información sobre los productos de la empresa. Sólo un administrador puede añadir, eliminar o modificar objetos en esta clase. La clase producto puede estar relacionada o no con la clase oferta.

Oferta: esta clase está relacionada con un sólo producto para indicar que este último está en oferta (lleva un descuento).

Marca: la clase marca representa a todas las marcas de los productos ofrecidos por la empresa.

Subcategoría: esta clase agrupa a un conjunto de productos del mismo tipo.

Categoría: una categoría se puede dividir en muchas subcategorías. En nuestra aplicación tenemos a dos categorías: videovigilancia e informática.

3.2 Diagrama de casos de uso

Es un diagrama que da más detalles sobre el comportamiento de la aplicación desde el punto de vista del usuario, de otra forma podemos decir que los diagramas de casos de uso representan las funciones que la aplicación puede ejecutar. Es fácil de interpretar, lo que hace que sea especialmente útil en la comunicación con el cliente.

Uno de los componentes de un diagrama de casos de uso son los actores, en nuestro caso el actor es el usuario de la aplicación, que puede ser un cliente cualquiera o un administrador.

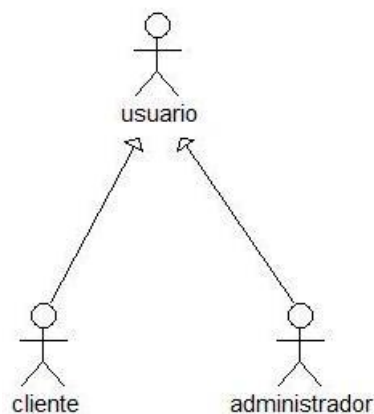


Figura 3. Actores de la aplicación

Las funcionalidades que desarrolla cada uno de los dos actores ya se han explicado en la parte de especificación de requisitos, aquí sólo vamos a ver los diagramas de casos de uso.

Administrador:

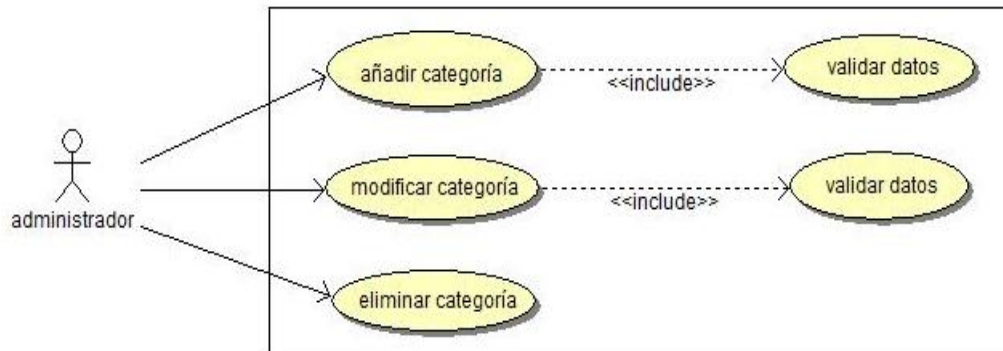


Figura 4. Caso de uso, gestión categoría

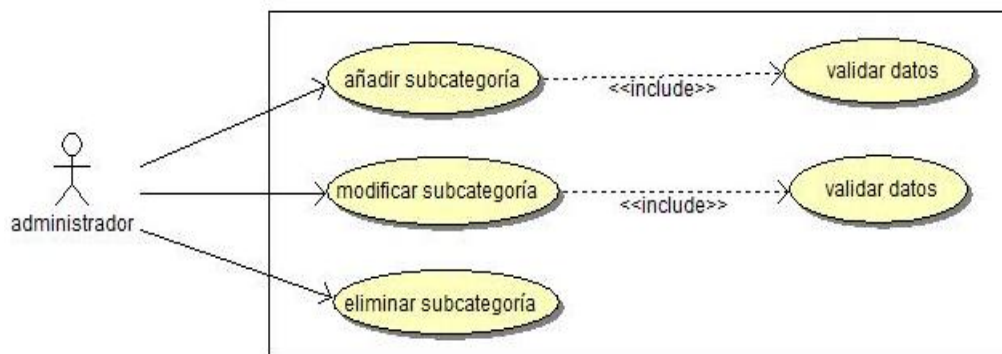


Figura 5. Caso de uso, gestión subcategoría

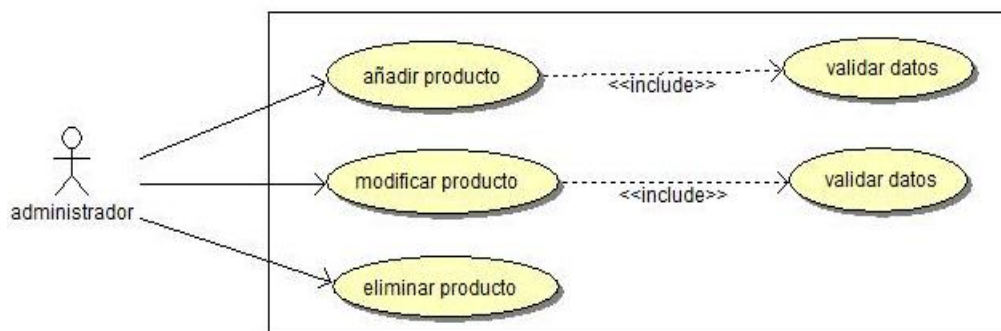


Figura 6. Caso de uso, gestión producto

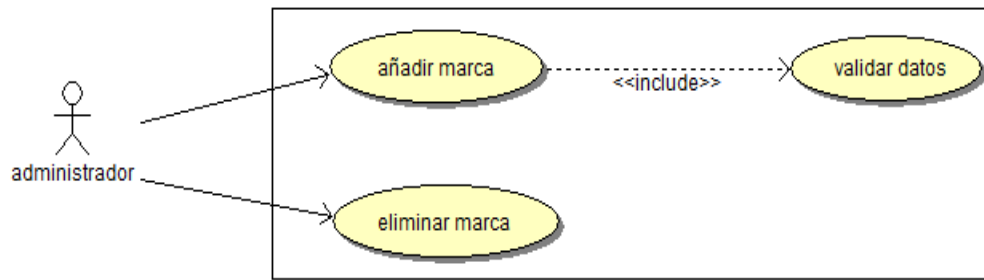


Figura 7. Caso de uso, gestión marca

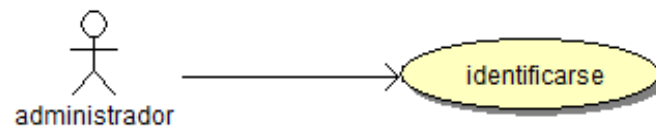


Figura 8. Caso de uso: iniciar sesión

Ciente:

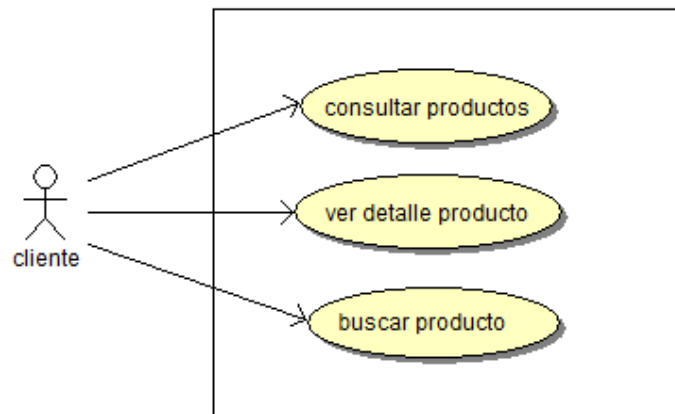


Figura 9. Caso de uso cliente

3.3 Diagramas de secuencia

Un diagrama de secuencia describe la interacción entre un conjunto de objetos en una aplicación y especifica los mensajes que estos objetos envían entre ellos para realizar una tarea determinada.

Normalmente, para cada caso de uso en la aplicación, se modela un diagrama de secuencia correspondiente.

En esta parte, vamos a ver los diagramas de secuencia correspondientes a los casos de uso de gestión de un producto y de iniciar sesión, el resto de los diagramas son casi idénticos.

Todos los casos de uso han sido explicados en el apartado de especificación de los requisitos.

Iniciar sesión:

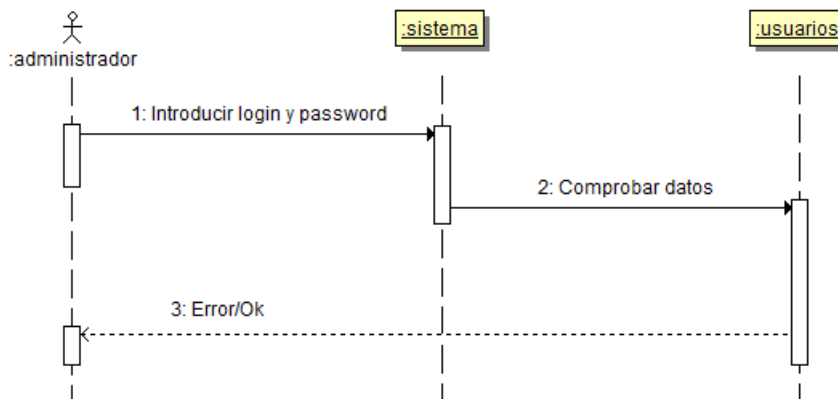


Figura 10. Diagrama de secuencia iniciar sesión

Añadir producto:

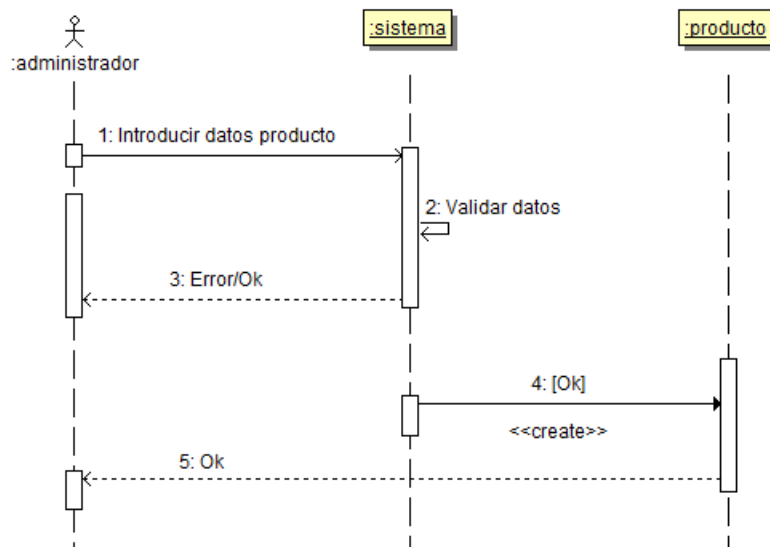


Figura 11. Diagrama de secuencia añadir producto

Modificar producto:

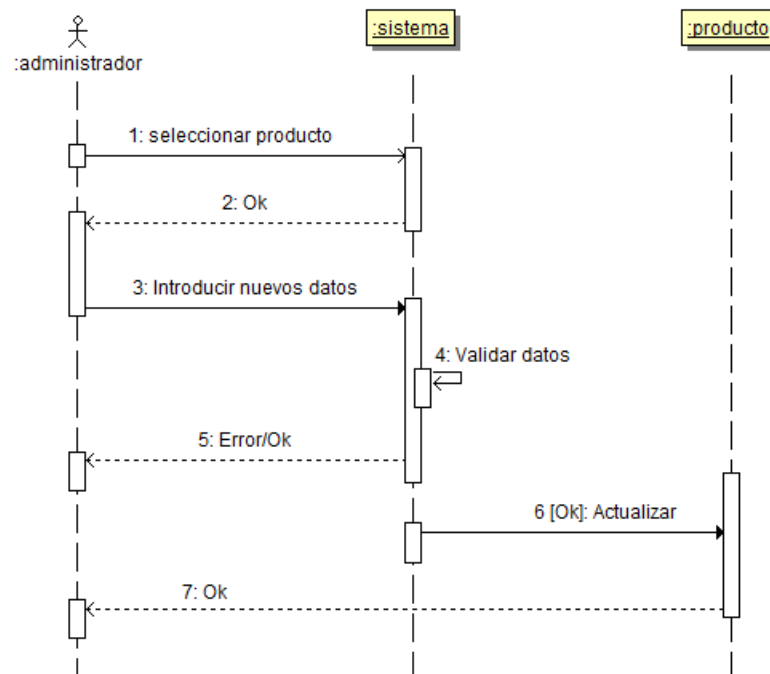


Figura 12. Diagrama de secuencia modificar producto

Eliminar producto:

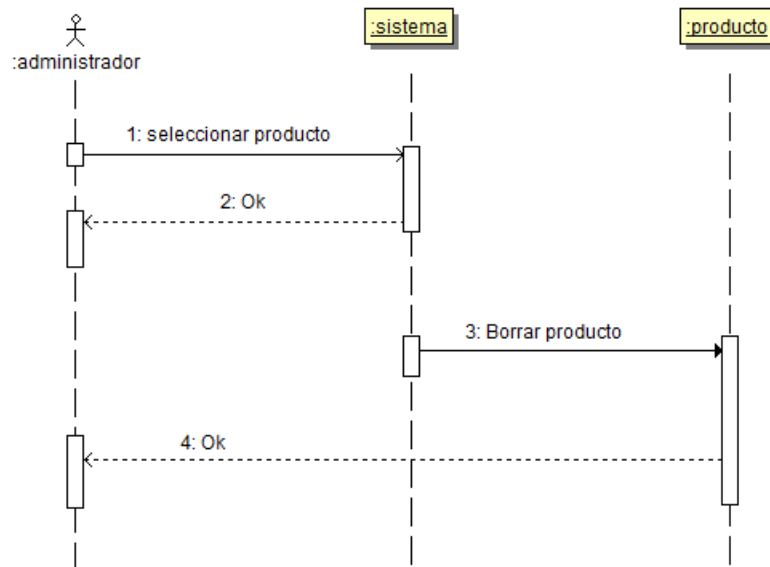


Figura 13. Diagrama de secuencia eliminar producto

4. Diseño

La aplicación web que se desarrolla en este proyecto es un programa informático que puede dar servicio simultáneamente a múltiples usuarios que lo ejecutan a través de internet. El diseño de la aplicación se basa en lo que se conoce como una arquitectura de tres capas, donde los diferentes actores y elementos implicados en la misma se encuentran distribuidos en tres bloques o capas que son:

- Capa de presentación.
- Capa de negocio.
- Capa de persistencia.

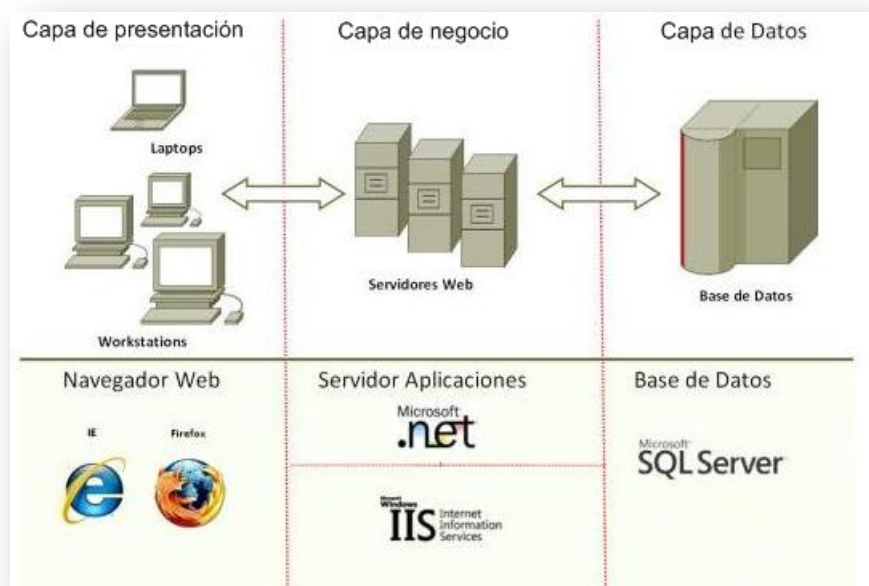


Figura 14. Arquitectura de tres capas

4.1 Capa de presentación

Se trata de la capa con la que interactúa el usuario de la aplicación. Realiza principalmente dos funciones, la primera es encargarse de capturar los datos de

usuario con los que opera la aplicación y enviárselos a ésta, la segunda de las funciones que realiza esta capa es la de presentar al usuario los resultados generados por la aplicación.

En una arquitectura Web de tres capas, la capa de presentación está implementada por el navegador. Mediante las llamadas páginas Web, los navegadores solicitan datos al usuario a través de la pantalla y presentan resultados en la misma.

Como hemos mencionado en apartados anteriores, nuestra aplicación se compone de dos zonas (dos páginas), la zona pública que ofrece a los clientes toda la información necesaria sobre los productos, y la zona privada que está reservada a los administradores registrados para permitirles gestionar la base de datos.

Para el desarrollo de las interfaces de usuarios utilizaremos el lenguaje de marcado HTML y las hojas de estilo CSS.

4.1.1 Zona pública:

Al diseñar la página principal, hemos tenido en cuenta la facilidad que debe tener el usuario para navegar entre las diferentes secciones de la página.

Cualquier usuario de internet puede acceder a la página Web principal de la empresa. La página tiene el siguiente aspecto:



Figura 15. Página principal de la empresa

Como podemos observar en la figura 15, la página principal de la empresa se compone de varias zonas:

Cabecera de la página (1):

Está formada por el logotipo y el nombre de la empresa que se sitúan en la parte izquierda, el resto de la cabecera lo ocupa una imagen que agrupa algunos productos ofrecidos por la empresa.

Menú horizontal (2):

Se compone de una zona para buscar productos (no está implementada y se dejará para posibles ampliaciones de la aplicación) y un conjunto de enlaces que permitirán al usuario navegar en las diferentes secciones de la página Web.

Menú vertical (3):

Se compone de un conjunto de enlaces a las diferentes subcategorías, cada una de ellas agrupa a un conjunto de productos del mismo tipo.

En la parte inferior de esta zona hay una animación sencilla que consiste en un conjunto de imágenes de los productos, que se irán repitiendo.

Pie de la página (4):

En esta zona se encuentra la información necesaria para ponerse en contacto con la empresa, como el nombre, la dirección, el teléfono, el fax o la dirección del correo electrónico.

Zona de información (5):

Es la zona encargada de acceder a la base de datos de la empresa para ofrecer al usuario toda la información que busca, en esta zona se mostrarán los detalles de los productos, la información de contacto con la empresa, los productos favoritos, etc.

Cuando el usuario navega por la aplicación, la información en esta zona irá cambiando según lo que está buscando, el resto de las zonas vistas anteriormente se mantendrán iguales.

4.1.2 Zona de administración:

Antes de acceder a esta zona, el administrador tiene que identificarse introduciendo un nombre de usuario y una contraseña. La interfaz que nos permite autenticarse es la siguiente:

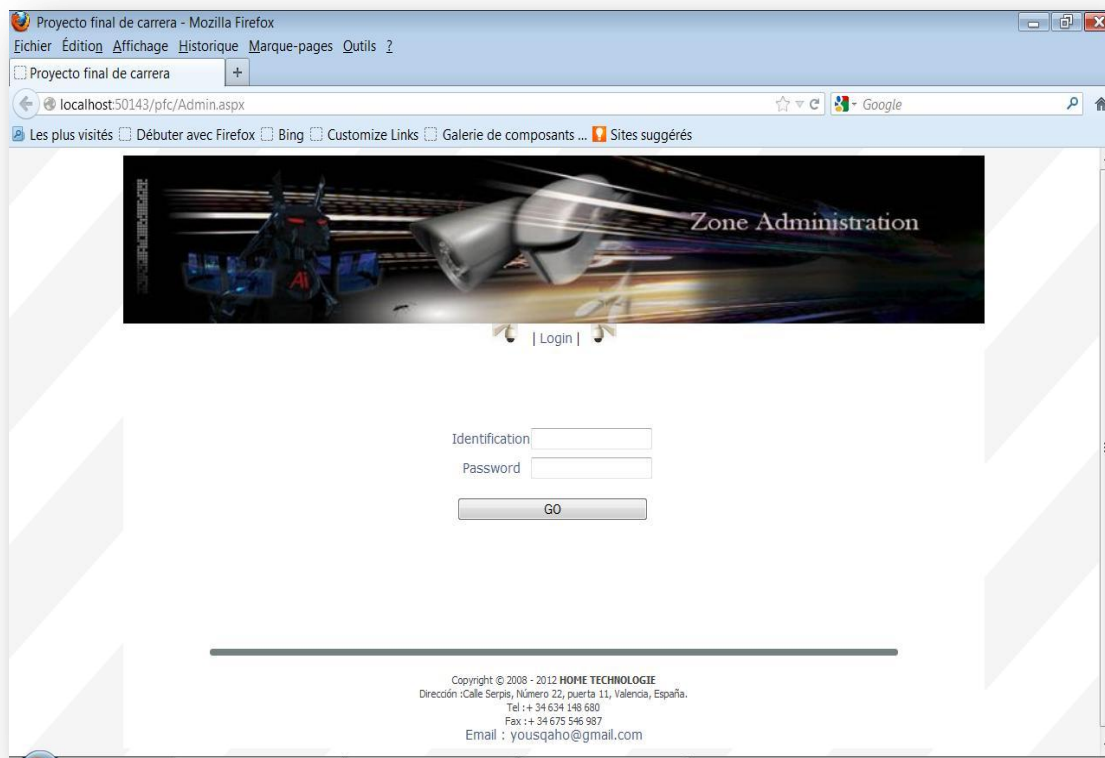


Figura 16. Página para iniciar una nueva sesión de administrador

Los datos de conexión introducidos se comprueban en la tabla 'utilisateurs', si son válidos, el administrador accede a la página donde puede realizar las operaciones de gestión de la base de datos, la primera interfaz que nos sale nos mostrará un mensaje de la bienvenida.

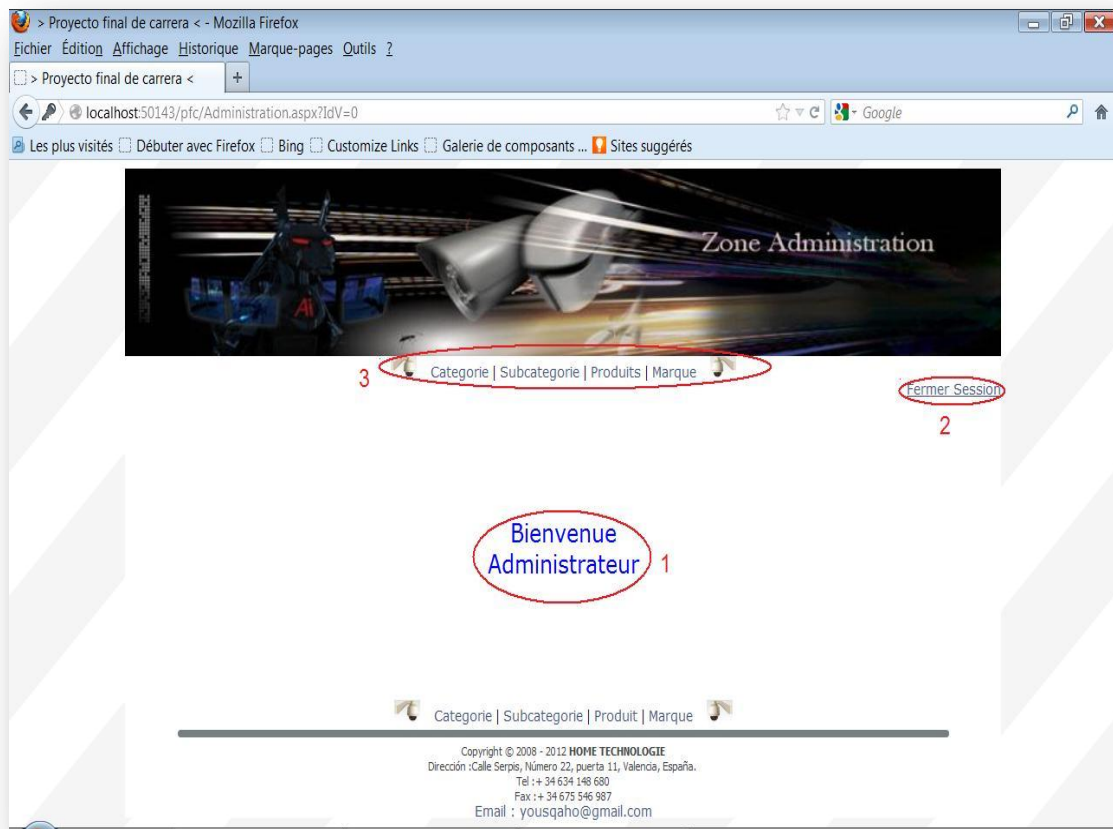


Figura 17. Página principal del administrador

En esta captura podemos ver el mensaje de bienvenida en el centro de la página (1), arriba de la página hay cinco enlaces, el que está más a la derecha (2) es para el cierre de la sesión, los otros enlaces (3) son para la gestión de la base de datos y cada uno de ellos nos llevará a la página correspondiente a una de las tablas de la base de datos.

Si hacemos un clic por ejemplo en el enlace 'categorie' nos llevará a la siguiente página:

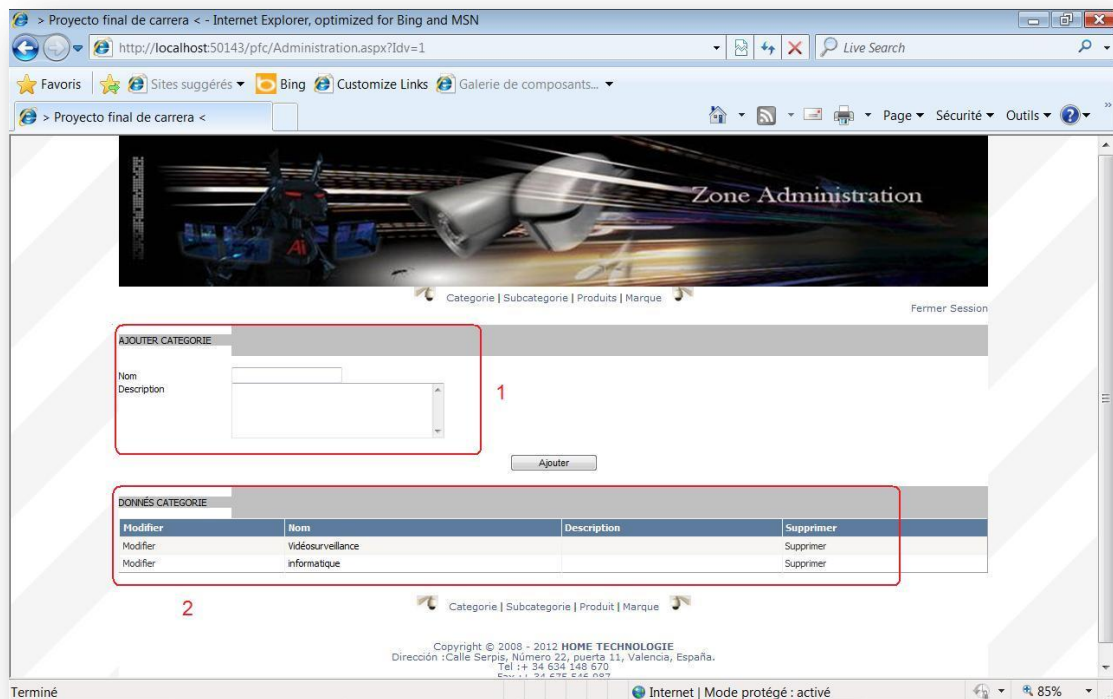



Figura 18. Página correspondiente a Categoría

Como podemos observar, la zona (1) contiene el formulario que nos permite introducir los datos de una nueva categoría, después de introducir el nombre y la descripción pulsamos sobre el botón 'Ajouter' y la nueva categoría se almacena en nuestra base de datos.

La zona (2) nos muestra una tabla con toda la información sobre las categorías existentes en la base de datos, en esta zona podemos modificar algunos datos o seleccionar un registro para eliminarlo de la base de datos.

Si navegamos por el resto de los enlaces podremos acceder a la información sobre las subcategorías, los productos y las marcas, con la opción de añadir nuevos registros, modificar alguna información o eliminar registros de la base de datos. A continuación mostramos las capturas de pantalla de dichas páginas.



Zone Administration

[Categorie](#) | [Subcategorie](#) | [Produits](#) | [Marque](#)

[Fermer Session](#)

AJOUTER SUBCATEGORIE

Nom

Description

Categorie Vidéosurveillance ▾

DONNÉS SUBCATEGORIE

Categorie Vidéosurveillance ▾

Modifier	Nom	Description	Supprimer
Modifier	Accessoires	Accessoires vidéosurveillance	Supprimer
Modifier	All in one Cameras		Supprimer
Modifier	Box Cameras		Supprimer
Modifier	Dome Cameras		Supprimer
Modifier	DVR		Supprimer
Modifier	DVR Card		Supprimer
Modifier	Hidden Cameras		Supprimer
Modifier	IP speed Dome Camera	IP speed Dome Camera	Supprimer
Modifier	Mini Speed Dome		Supprimer
Modifier	Miniature Cameras		Supprimer
Modifier	Network IR Cameras		Supprimer
Modifier	Speed Dome Cameras		Supprimer
Modifier	Weatherproof IR Cameras		Supprimer
Modifier	Wireless Cameras		Supprimer

[Categorie](#) | [Subcategorie](#) | [Produit](#) | [Marque](#)

Copyright © 2008 - 2012 HOME TECHNOLOGIE
 Dirección : Calle Serpis, Número 22, puerta 11, Valencia, España.
 Tel : + 34 634 148 680
 Fax : + 34 675 546 987
 Email : yousqah@gmail.com

Figura 19. Página correspondiente a Subcategorie



AJOUTER PRODUIT

Code

Nom

Description

Prix

Prix Revendeur

Photo 100/100 px

Photo 200/200 px

Marque

Subcategorie

En Offre

En Accueil

Promo

Nouveauté

Fiche Technique

DONNÉES PRODUIT

Subcategorie







Modifier	Code	Nom	Description	Photo	Marque	Subcategorie	En Offre	Offre Accueil	Visible?	Fiche Technique	Supprimer
Modifier	LPI523S	Network/IP Cameras LPI523S			SONY	Network IR Cameras	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Color 1/3" SONY, 420 TV Lines, Weatherproof IR Network Cameras IR LED: 75X23PCS IR working distance: 20M Mpeg-4 Weight:1.1kg	Supprimer
Modifier	LNIR60S	Network/IP Cameras LNIR60S	Color 1/3" SONY, 420 TV Lines, Weatherproof IR Network Cameras, IR LED: 75X60PCS IR working distance: 50M Lens: Board Lens 8mm/F2.0(optional), Software for 16 CH included Mpeg-4, Weight:1.5kg		SONY	Network IR Cameras	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Supprimer
Modifier	LNIS36S	Network/IP Cameras LNIS36S	Color 1/3" SONY, 420 TV Lines, Weatherproof IR Network Cameras, IR LED: 75X36PCS IR working distance: 30M Lens: Board Lens 6mm/F2.0(optional), Software for 16 CH included Mpeg-4, Weight:1.2kg		SONY	Network IR Cameras	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Supprimer
Modifier	LNIS48S	Network/IP Cameras LNIS48S	Color 1/3" SONY, 420 TV Lines, Weatherproof IR Network Cameras, IR LED: 75X48PCS IR working distance: 40M Lens: Board Lens 6mm/F2.0(optional), Software for 16 CH included Mpeg-4, Weight:1.3kg		SONY	Network IR Cameras	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Supprimer
Modifier	LNS100	Network/IP Cameras LNS100	Color 1/3" SONY, 420 TV Lines, Network Video Server, Single, Support CIF, Mpeg-4, Weight:700g		SONY	Network IR Cameras	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Supprimer
Modifier	LNCBS	Wirless System LNCBS	Color 1/3" SONY, 420 TV Lines, Standard Network Cameras, Lens: CS Lens 4mm(optional) Software for 16 CH included Mpeg-4, Weight:1.0kg		SONY	Network IR Cameras	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		Supprimer

Figura 20. Página correspondiente a Producto

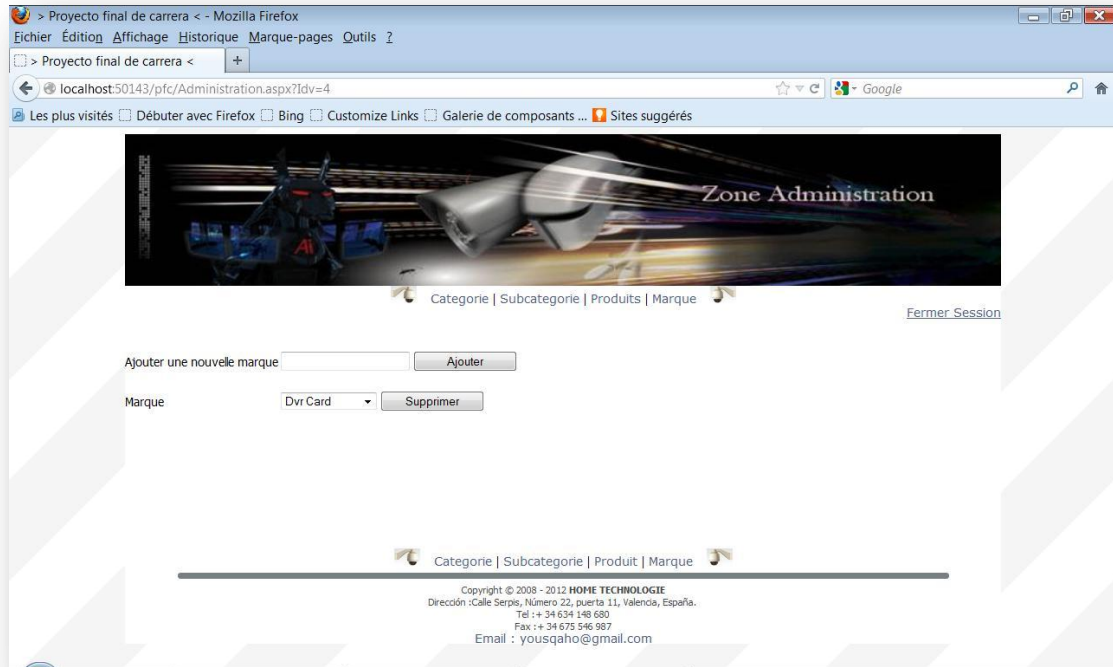


Figura 21. Página correspondiente a marca

Como podemos observar, todas las páginas de la zona de administración tienen el mismo diseño, sólo cambia el formulario para la introducción de los datos y la tabla que contiene la información, que dependen de la sección de la base de datos que queremos gestionar.

4.2 Capa de negocio

La capa de negocio está construida por la aplicación web en sí, ésta se encuentra instalada en una maquina independiente, conocida como servidor, a la que acceden los clientes a través de la red.

La aplicación de la capa de negocio es ejecutada por un motor de aplicación especial capaz de permitir que una misma instancia de la misma pueda dar servicio a múltiples clientes. Además de este motor, los servidores necesitan otro software conocido como servidor Web, que sirva de interfaz entre la aplicación y el cliente, realizando el dialogo HTTP con este. Puesto que nuestra aplicación está desarrollada

con la tecnología .NET de Microsoft, el servidor Web que utilizamos es Internet Information Services, conocido también como IIS.

Así pues, de forma resumida podríamos decir que las funciones de la capa de negocio consisten en:

- Recoger los datos enviados desde la capa de presentación.
- Procesar la información y, en general, implementar la lógica de aplicación, incluyendo el acceso a los datos.
- Generar las respuestas para el cliente.

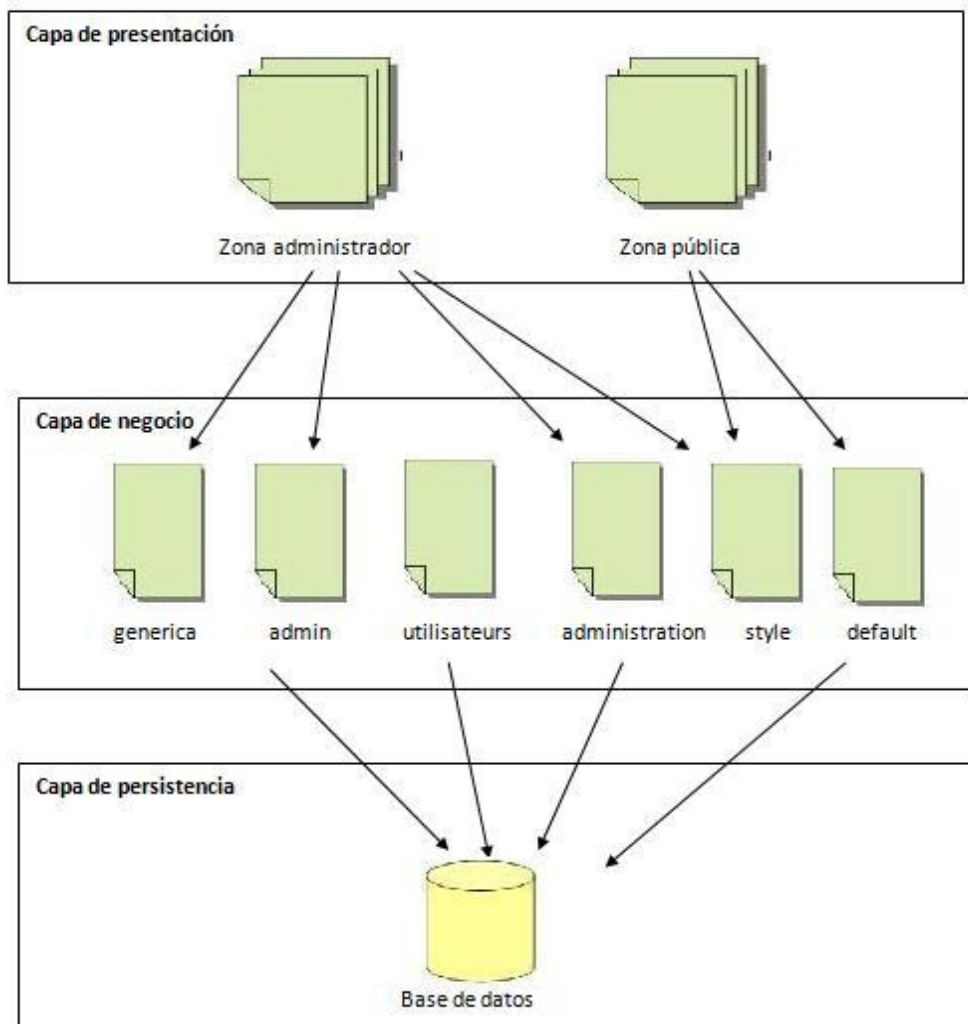


Figura 22. Componentes por nivel de capa

Como podemos ver en la figura, la capa de negocio se compone de seis ficheros, uno de ellos es el fichero Admin que se encarga de recoger los datos introducidos por el administrador para identificarse (nombre de usuario contraseña) y los comprueba con la base de datos, si el usuario está registrado como administrador lo lleva a su zona correspondiente. El resto de los ficheros son los encargados de implementar la lógica de la aplicación.

4.3 Capa de persistencia

La capa de persistencia tiene como misión el almacenamiento permanente de la información manejada por la aplicación en una base de datos que nos permite extraer los datos necesarios cuando los necesitamos.

Una de las etapas más importantes en el desarrollo del proyecto consiste en el diseño de una base de datos robusta, para esta tarea utilizaremos las llamadas bases de datos relacionales. Distribuiremos la información entre diferentes tablas, relacionándolas entre sí a través de un campo común que permita identificar los registros de una tabla que se corresponden con los de otra.

Para el proceso de almacenamiento y obtención de datos utilizaremos las sentencias SQL y como servidor de base de datos utilizaremos Microsoft SQL Server 2005 que está muy bien integrado con Visual Studio y es fácil de administrar, de hecho es la base de datos predeterminada en visual studio.

Después de especificar los requisitos del proyecto con detalle, podemos pasar a identificar las tablas de la base de datos y las relaciones entre ellas para definir la estructura de la base de datos.

A continuación veremos las tablas que componen nuestra base de datos.

Empezamos mostrando la estructura de la tabla 'Utilisateurs' que almacenará los detalles de los administradores que son los encargados de gestionar la base de datos y que tienen que introducir un nombre de usuario y una contraseña para poder acceder a la base de datos.

La tabla 'utilisateurs' tiene seis columnas, cinco de ellas almacenan el login del administrador, su contraseña (Password), su nombre (Nom), su correo electrónico (Email) y su teléfono (tel), mientras que la sexta indica si el usuario es un administrador. El campo login se define como la clave primaria de la tabla.

Nombre de columna	Tipo de datos	Permitir valores nulos
Login	Varchar(100)	0
Password	Varchar(50)	1
Nom	Varchar(100)	1
Admin	char(3)	1
Email	Varchar(50)	1
tel	Varchar(20)	1

Tabla 1. Estructura de la tabla utilisateurs

La tabla 'Marque' almacena los nombres de todas las marcas que se pueden encontrar en los almacenes de la empresa. Esta tabla se compone de una sola columna (Nom) que representa el nombre de la marca.

Nombre de columna	Tipo de datos	Permitir valores nulos
Nom	Varchar(100)	0

Tabla 2. Estructura de la tabla marque

La tabla 'Produits' almacena la información relativa a los productos disponibles, esta información incluye el código, nombre, marca, fotos, etc. Cada producto almacenado tiene un código único, que se define como clave principal de la tabla.

Nombre de columna	Tipo de datos	Permitir valores nulos
code	Varchar(100)	0
nom	Varchar(300)	1
description	text	1
prix	Varchar(50)	1
photo	text	1
marque	Varchar(100)	1
subcategorie	Varchar(50)	1
offre	bit	1
Offre_acceuil	bit	1
Photo_big	text	1
Visible	bit	1
⋮	⋮	⋮

Tabla 3. Estructura de la tabla produits

La tabla 'offres' almacena todos los productos que están en oferta, tiene dos campos, el primero es la clave primaria de la tabla (id) y el segundo hace referencia al producto ofertado (code_produit).

Nombre de columna	Tipo de datos	Permitir valores nulos
id	int	0
Code_produit	Varchar(100)	0

Tabla 4. Estructura de la tabla offres

La tabla 'categorie' almacena información sobre las categorías de los productos, se compone de tres columnas que almacenan el identificador de la categoría (id) y que representa la clave primaria de la tabla, el nombre de la categoría (nom) y una breve descripción (description).

Nombre de columna	Tipo de datos	Permitir valores nulos
id	int	0
Nom	Varchar(50)	1
description	Varchar(200)	1

Tabla 5. Estructura de la tabla categorie

Cada categoría representada en la última tabla se compone de varias subcategorías que representamos en la siguiente tabla. La estructura de la tabla 'subcategorie' contiene tres campos, uno es el nombre (nom) y es la clave primaria de la tabla, y los otros dos son la descripción y el campo 'categorie' que hace referencia a la categoría al que pertenece.

Nombre de columna	Tipo de datos	Permitir valores nulos
Nom	Varchar(50)	0
Description	Varchar(200)	1
categorie	int	1

Tabla 6. Estructura de la tabla subcategorie

Después de crear las tablas de la base de datos, tenemos que establecer relaciones entre ellas, en la siguiente figura se muestran estas relaciones.

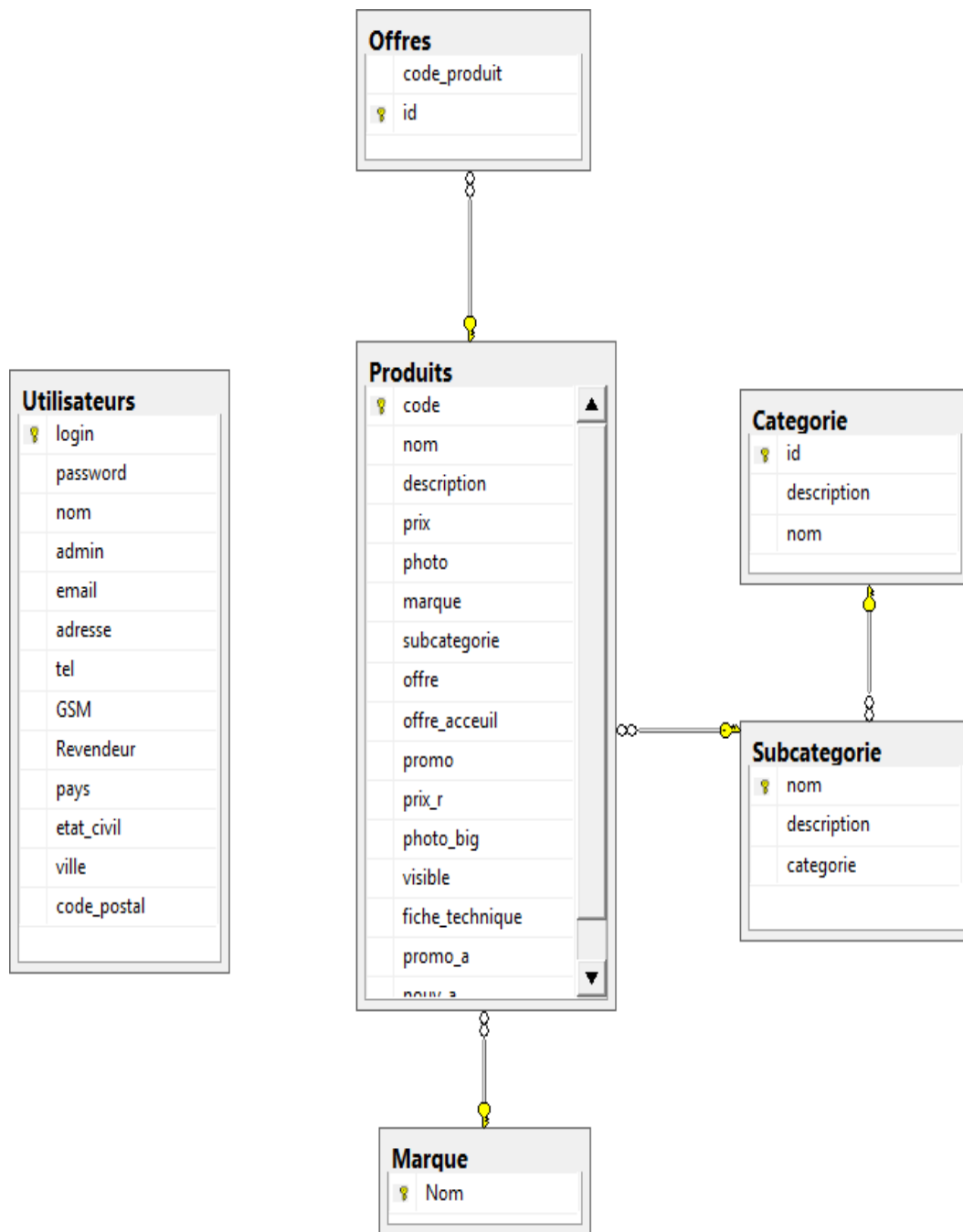


Figura 20. Estructura de la base de datos

En la tabla siguiente se explican las relaciones entre las tablas de la base de datos.

Tablas relacionadas	relación	observaciones
Categorie--Subcategorie	One-to-many	Una categoría puede dividirse en muchas subcategorías, pero una subcategoría sólo puede pertenecer a una categoría
Subcategorie--Produits	One-to-many	Una subcategoría puede tener muchos productos, pero un producto sólo puede pertenecer a subcategoría
Produits--offres	One-to-one	Un producto sólo puede pertenecer a una oferta, y una oferta sólo puede componerse de un producto
Marque-- Produits	One-to-many	Una marca puede tener muchos productos, pero un producto sólo puede ser de una marca

5. Implementación

5.1 Detalles de la implementación

En las etapas anteriores hemos realizado un análisis completo de la aplicación y hemos diseñado los formularios necesarios. En esta parte, implementaremos la lógica de negocio que realiza los objetivos que se vieron al plantear el proyecto.

La plataforma .NET nos proporciona una serie de elementos y componentes que nos ofrecen gran potencia y facilidad a la hora de la implementación, Entre estos elementos están:

- los formularios Web que representan las páginas Web de la aplicación a través de las cuales se producirá la interacción con el usuario.
- los controles web y controles HTML que son los elementos gráficos que permiten realizar el diseño de la interfaz de usuario.

Nuestra aplicación Web contiene tres formularios, dos para el administrador (Admin.aspx y administration.aspx) y uno para los clientes (default.aspx), para construir estos componentes, o mejor dicho, interfaces, utilizaremos el lenguaje de marcado estándar HTML acompañado con hojas de estilo (CSS).

Para el diseño de la aplicación, utilizaremos muchos controles de servidor que proporcionan una interfaz gráfica basada en HTML para facilitar al cliente la interacción con la aplicación, entre estos controles hay botones, Label, TextBox, HyperLink, DropDownList, etc. A continuación veremos unos ejemplos de la utilización de estos controles.

Mediante el control label podemos presentar en el interior de la página un texto informativo al usuario, mientras que el control TextBox se utiliza para capturar datos de usuario, el siguiente fragmento de código nos muestra un ejemplo de la utilización de los dos controles:

```
<tr>
  <td align="left" style="width: 106px; height: 30px">
    <asp:Label ID="Label3" runat="server" Text="Password"
    CssClass="blue-text"></asp:Label>
  </td>
  <td style="width: 261px; height: 30px">
    <asp:TextBox ID="password" runat="server" TextMode="Password"
    Width="140px"></asp:TextBox>
  </td>
</tr>
```

A veces es necesario comprobar que un determinado campo de texto haya sido rellenado, para esta situación, utilizaremos el control de validación `RequiredFieldValidator` que lo vinculamos a un determinado control de la interfaz de usuario para forzar la introducción de un valor en él, convirtiendo así al control en un campo requerido.

En el siguiente ejemplo vemos como el control de validación está vinculado al control `TextBox` mediante el atributo `ControlToValidate` que toma como valor el ID del `TextBox`, si el usuario deja el campo de texto vacío, le sale un mensaje en rojo advirtiéndole de que el campo no debe estar vacío.

```
<td align="left" style="height: 19px">
<asp:TextBox ID="nomCategorie" runat="server"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator6"
runat="server" ControlToValidate="nomCategorie"
ErrorMessage="Nom ne peut pas etre vide">
</asp:RequiredFieldValidator>

</td>
```

El control `DropDownList` es una lista desplegable de selección simple, el siguiente ejemplo muestra una utilización del control:

```
<td align="left" style="height: 21px">
<asp:DropDownList ID="categorie_de_subcategorie" runat="server"
DataSourceID="SqlDataSource1"
DataTextField="nom" DataValueField="id">
</asp:DropDownList>
</td>
```

El control `HyperLink` representa un enlace HTML. Su única propiedad destacable es `NavigateUrl`, en la que se especificará la URL de la página solicitada al ser activado el enlace. El siguiente ejemplo muestra un control `HyperLink`:

```
<asp:HyperLink ID="Subcat" runat="server" CssClass="blue-link"
NavigateUrl="~/Administration.aspx?Idv=2">Subcategorie
</asp:HyperLink>
```

Algunos controles que utilizamos exponen eventos que son disparados al realizar el cliente algún tipo de acción sobre el control, como, por ejemplo, un clic en un botón.

En el siguiente ejemplo vemos como el evento 'click' del control `LinkButton` está asociado con el método denominado `session_Click`:

```
<asp:LinkButton ID="session" runat="server"
  CssClass="blue-link"
  OnClick="session_Click" >Fermer Session
</asp:LinkButton>
```

El código del método 'session_click' se encuentra en el archivo Administration.aspx.cs, lo que hace es cerrar la sesión del usuario conectado y redireccionarlo a la página principal de la empresa.

```
protected void session_Click(object sender, EventArgs e)
{
    Session["user"] = null;
    Response.Redirect("Default.aspx");
}
```

Para introducir un nuevo producto o modificar uno existente, el administrador tiene que seleccionar el archivo correspondiente a la foto del producto para enviarlo al servidor, para facilitar esta tarea, empleamos el control FileUpload que permite al administrador buscar y seleccionar fácilmente el archivo que quiere transferir, proporcionando un botón 'examinar' y un cuadro de texto para introducir el nombre del archivo.

En el siguiente bloque de código contiene un control FileUpload:

```
<tr>
  <td align="left" style="width: 154px; height: 22px">
    <asp:Label ID="Label19" runat="server" CssClass="labelStyle"
      Text="Photo 100/100 px"></asp:Label></td>

  <td align="left" style="height: 22px">

    <asp:FileUpload ID="photo_produit" runat="server" />

    <asp:RequiredFieldValidator ID="RequiredFieldValidator9"
      runat="server" ControlToValidate="photo_produit"
      ErrorMessage="Insérer le chemin de la photo du produit">
    </asp:RequiredFieldValidator>&nbsp;  </td>
</tr>
```

En el navegador Web, el control FileUpload tiene el siguiente aspecto:

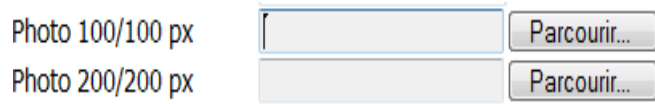


Figura 23. Aspecto de dos controles FileUpload

Para rellenar la página Web principal de la empresa con información, primero tenemos que extraer los datos necesarios de la base de datos. Para conectarse a la base de datos e interactuar con ella, utilizaremos el control de origen de datos SqlDataSource que tiene la información de la conexión, la información de la consulta y los parámetros necesarios para que podamos vincularnos a varios objetos de la interfaz (como DataList o GridView), para que se muestren en la página Web.

```
<asp:SqlDataSource ID="SqlDataSource5" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
SelectCommand="SELECT * FROM [Subcategorie] WHERE
([categorie] = @categorie)">
<SelectParameters>

<asp:ControlParameter ControlID="categorie_selection"
DefaultValue="-1" Name="categorie"
PropertyName="SelectedValue" Type="Int32" />

</SelectParameters>
</asp:SqlDataSource>
```

Una vez tenemos un control de origen de datos funcional en nuestra página Web, podemos conectarlo a un control de interfaz para que muestre los datos, que recuperamos de la base de datos, en la página Web. En nuestra aplicación usaremos los controles de interfaz GridView, DataGrid y DataList.

Como podemos observar en la zona de administración, se puede seleccionar algunos elementos de la base de datos para representarlos en una tabla como demuestra la figura siguiente:

Modifier	Nom	Description	Supprimer
Modifier	Accessoires	Accessoires vidéosurveillance	Supprimer
Modifier	All in one Cameras		Supprimer
Modifier	Box Cameras		Supprimer
Modifier	Dome Cameras		Supprimer
Modifier	DVR		Supprimer
Modifier	DVR Card		Supprimer
Modifier	Hidden Cameras		Supprimer
Modifier	IP speed Dome Camera	IP speed Dome Camera	Supprimer
Modifier	Mini Speed Dome		Supprimer
Modifier	Miniature Cameras		Supprimer
Modifier	Network IR Cameras		Supprimer
Modifier	Speed Dome Cameras		Supprimer
Modifier	Weatherproof IR Cameras		Supprimer
Modifier	Wireless Cameras		Supprimer

Figura 24. Resultado de la utilización del control DataGrid

Esto se puede conseguir gracias al control DataGrid que es uno de los controles más potentes y complicados de ASP.NET, DataGrid nos permite mostrar los elementos de un origen de datos en una tabla ordenada.

El código HTML correspondiente a la tabla de la figura 25 es el siguiente:

```
<asp:DataGrid ID="DataGridSubCategorie" runat="server"
AutoGenerateColumns="False" CellPadding="4" CellSpacing="1"
DataSourceID="SqlDataSource5" ForeColor="#333333" GridLines="None"
OnItemCommand="DataGridSubCategorie_ItemCommand" Width="100%">

<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White"/>
<ItemStyle BackColor="#F7F6F3" />
<AlternatingItemStyle BackColor="White" />
<EditItemStyle BackColor="#999990" ForeColor="White" />
<SelectedItemStyle BackColor="#E2DED6" Font-Bold="True"
ForeColor="#333333" />

<Columns>
<asp:EditCommandColumn CancelText="Fermer" CausesValidation="False"
EditText="Modifier"
HeaderText="Modifier" UpdateText="Mettre &#224; jour">
</asp:EditCommandColumn>
<asp:BoundColumn DataField="categorie" HeaderText="categorie"
ReadOnly="True" Visible="False">
</asp:BoundColumn>
<asp:BoundColumn DataField="Nom" HeaderText="Nom" ReadOnly="True">
</asp:BoundColumn>
<asp:BoundColumn DataField="Description" HeaderText="Description">
</asp:BoundColumn>
<asp:ButtonColumn CommandName="Delete" HeaderText="Supprimer"
Text="Supprimer">
</asp:ButtonColumn>
</Columns>
</asp:DataGrid>
```


Otro de los controles de interfaz que utilizamos es el control DataList, que sirve para crear listas de datos y mostrarlas en la página Web en un formato que se puede definir mediante plantillas y estilos. El control DataList resulta útil para los datos que se encuentran en una estructura de repetición, como una tabla.

El siguiente fragmento de código nos muestra la implementación de un control DataList vinculado al origen de datos SqlDataReader como se puede ver en el atributo DataSourceID:

```
<asp:DataList ID="DataList5" runat="server"
DataSourceID="SqlDataSource6" Height="73px"
RepeatColumns="3" RepeatDirection="Horizontal" Width="602px">

    <ItemTemplate>
        <table border="0" cellpadding="0" cellspacing="0" style="height:
46px" width="100%">
            <tr>
                <td align="center" style="height: 16px; background-color:
#f5f5f5" valign="middle">
                    
                    <asp:HyperLink ID="HyperLink7" runat="server"
                        CssClass="blue-link"
                        NavigateUrl='<%# DataBinder.Eval(Container,
"DataItem.categorie", "Default.aspx?IdV=1&Cat={0}&Subcat=")+Eval("subc
ategorie") %>'
                        Text='<%# Eval("subcategorie") %>'>
                    </asp:HyperLink>
                </td>
            </tr>
        </table>
    </ItemTemplate>
</asp:DataList>
```

El resultado de este código lo podemos ver en la página Web principal y debería parecerse a lo que aparece en la siguiente figura:

SUBCATEGORIES		
. Accessoires	. All in one Cameras	. Box Cameras
. Dome Cameras	. DVR	. DVR Card
. Hidden Cameras	. IP speed Dome Camera	. Mini Speed Dome
. Miniature Cameras	. Network IR Cameras	. Speed Dome Cameras
. Weatherproof IR Cameras	. Wireless Cameras	

Figura 25. Ejemplo de la utilización del control DataList

Una técnica que hemos utilizado en el diseño de la aplicación es dividir cada una de las dos páginas principales (default.aspx y administration.aspx) en diferentes partes, mostrando solo una parte cada vez, utilizando enlaces para desplazarnos de una parte a otra. Para implementar esta técnica hemos utilizado los controles MultiView y View.

El control View gestiona el contenido de una sección de la página: un control por fragmento, todos los objetos View se almacenan juntos en un objeto MultiView que hace que un objeto View sea visible en un determinado momento.

5.2 Tecnologías usadas

Para la realización del trabajo, se necesita adquirir previamente unos conocimientos sobre algunas tecnologías que creemos que se adaptaban mejor a las necesidades de nuestro proyecto.

5.2.1 Tecnologías y lenguajes

.NET: Por .NET entendemos el conjunto de tecnologías Microsoft destinadas al desarrollo de aplicaciones informáticas. Estas tecnologías se basan en la utilización de un modelo de desarrollo unificado, lo que significa que podrán ser utilizadas tanto

para el desarrollo de aplicaciones que se ejecuten en entorno Windows, como para las que vayan a ejecutar en entorno web o incluso para aquellas cuyo entorno de ejecución se encuentra en un dispositivo electrónico.

De todo este conjunto de tecnologías y herramientas que componen .NET podría realizarse la siguiente clasificación:

- Lenguajes de programación.
- Librerías de clases.
- Tecnología de ensamblados.
- Entorno de desarrollo.

ASP.NET: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Se utiliza mucho para construir sitios web dinámicos, aplicaciones web y servicios web XML. ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Las páginas de ASP.NET, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web.

Visual C#: es un lenguaje de programación orientado a objetos creado específicamente por Microsoft para trabajar sobre la plataforma .NET y, sin duda, el preferido por la mayoría de los programadores para el desarrollo de todos los tipos de aplicaciones que se van a ejecutar bajo esta plataforma.

Para comprender más la importancia que tiene vamos a comentar algunas de las principales características que proporciona este lenguaje:

- C# es una evolución del lenguaje C/C++, de hecho su sintaxis es muy parecida a la de C y sus características de orientación a objetos similares a las de C++. Al mismo tiempo, y sin perder la potencia y robustez de estos lenguajes, C# no permite la utilización de punteros, una característica bastante peligrosa que suele crear bastantes problemas a los programadores.

- C# incorpora una serie de características propias, como son los indizadores, las propiedades o los tipos genéricos, que lo hacen más potente y fácil de utilizar que a sus predecesores.
- Los programas escritos en C# pueden hacer uso de todas las librerías de clases proporcionadas por .NET Framework y beneficiarse de las ventajas del código administrado.

HTML: HTML, siglas de HyperText Markup Language (lenguaje de marcado de hipertexto), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.1

HTML también sirve para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

CSS: Es un lenguaje utilizado en la presentación de documentos HTML. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los desarrolladores web para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc.

La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación.

Algunas opciones básicas del lenguaje CSS por ejemplo pueden ser el poder cambiar el color de algunas típicas etiquetas HTML como <H1> (h1 es una etiqueta en el lenguaje HTML destinada a mostrar un texto como encabezado, en tamaño grande). Pero también hay funciones algo más complejas, como introducir espaciado entre elementos <DIV> (div es una etiqueta HTML para identificar una determinada región o división de contenido dentro de una página web) o establecer imágenes de fondo.

CSS es muy intuitivo y sencillo una vez se llega a aprender, ya que para su definición siempre se hace uso de un identificador de etiqueta HTML (como por ejemplo <H1>),

y luego indicamos con qué aspecto queremos que se muestren todas las etiquetas <H1> que aparezcan en un documento. Al igual que con <H1> podemos definir cómo queremos que se muestren las distintas partes del documento HTML, pudiendo en cada caso definir sus propiedades (color, tipo de fuente, tamaño, espacio, imagen) con algún determinado valor deseado.

SQL: El nombre "SQL" es una abreviatura de Structured Query Language (Lenguaje de consultas estructurado), es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales.

El SQL es un lenguaje muy parecido al lenguaje natural; concretamente, se parece al inglés, y es muy expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales. Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros.

5.2.2 Soporte

Microsoft Visual Studio: Uno de los grandes logros de Microsoft con el lanzamiento de la plataforma .NET ha sido la creación de un único entorno de desarrollo para la construcción de las aplicaciones. Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows, con el que podemos crear cualquier tipo de aplicación .NET, desde programas para Windows hasta utilidades para una agenda electrónica, pasando, por supuesto, por la creación de aplicaciones para entorno Web, pudiendo además utilizar para ello cualquiera de los lenguajes soportados por la plataforma tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.

El nuevo framework está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF). El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento. La nueva versión del visual estudio no solo ofrece estas mejoras sino una gran cantidad más.

Macromedia Dreamweaver MX 2004

Dreamweaver es una aplicación en forma de estudio pero con más parecido a un taller destinado para la construcción y edición de sitios y aplicaciones Web basados en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

La gran baza de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que este programa, sus rutinas están hechas en Javascript-C lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino, rutinas de Javascript

que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo ponga a su gusto.

Como editor WYSIWYG que es, Dreamweaver oculta el código HTML de cara al usuario, haciendo posible que alguien no entendido pueda crear páginas y sitios web fácilmente.

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Un aspecto de alta consideración de Dreamweaver es su arquitectura extensible. Es decir, permite el uso de "Extensiones". Las extensiones, tal y como se conocen, son pequeños programas, que cualquier desarrollador web puede escribir (normalmente en HTML y Javascript) y que cualquiera puede descargar e instalar, ofreciendo así funcionalidades añadidas a la aplicación.

Estas tecnologías cumplen sobradamente con nuestras necesidades y están muy extendidas, por lo que no supondrá una dificultad encontrar documentación en la cual apoyarnos durante el desarrollo del proyecto.

6. Pruebas

Como hemos visto antes, la capa de presentación está implementada por el navegador. Un navegador es un tipo especial de software basado en una interfaz gráfica que utilizan los usuarios para interactuar con una aplicación Web.

Existen diferentes marcas y modelos de navegadores desarrollados por distintos fabricantes software. Una de las pruebas que realizaremos es comprobar que nuestro sitio Web se visualiza correctamente en los navegadores más conocidos.

Internet Explorer:

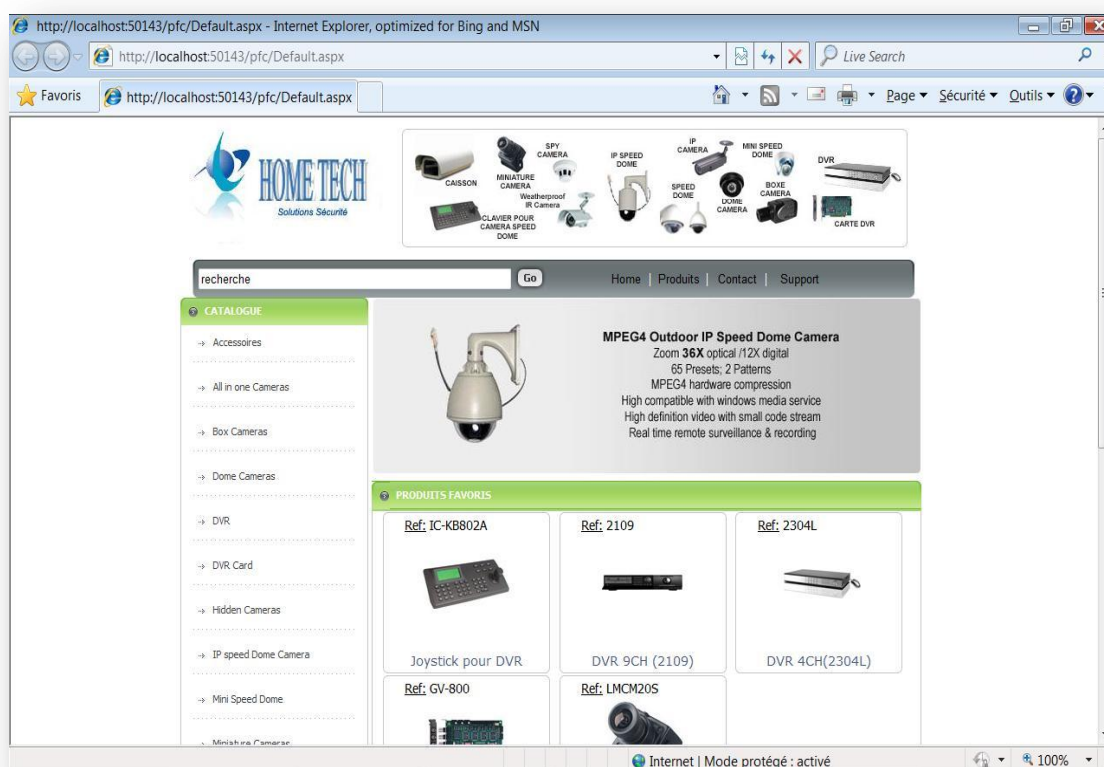


Figura 26. Visualización de la página principal en Internet Explorer

Mozilla Firefox:

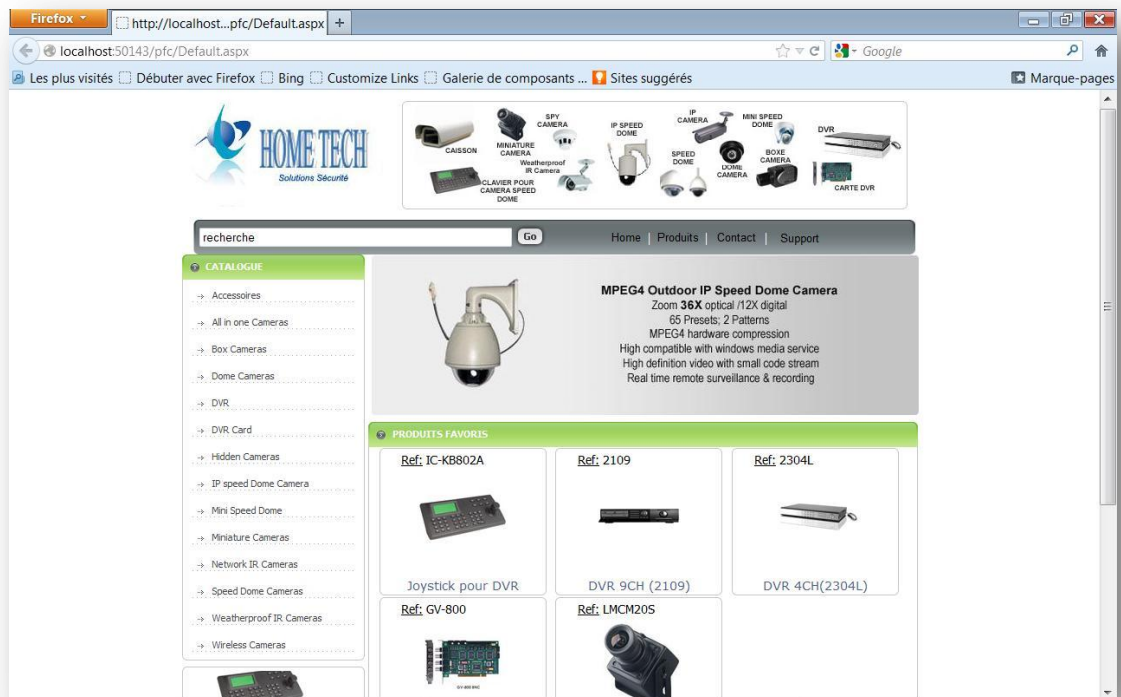


Figura 27. Visualización de la página principal en Mozilla Firefox

Google chrome:

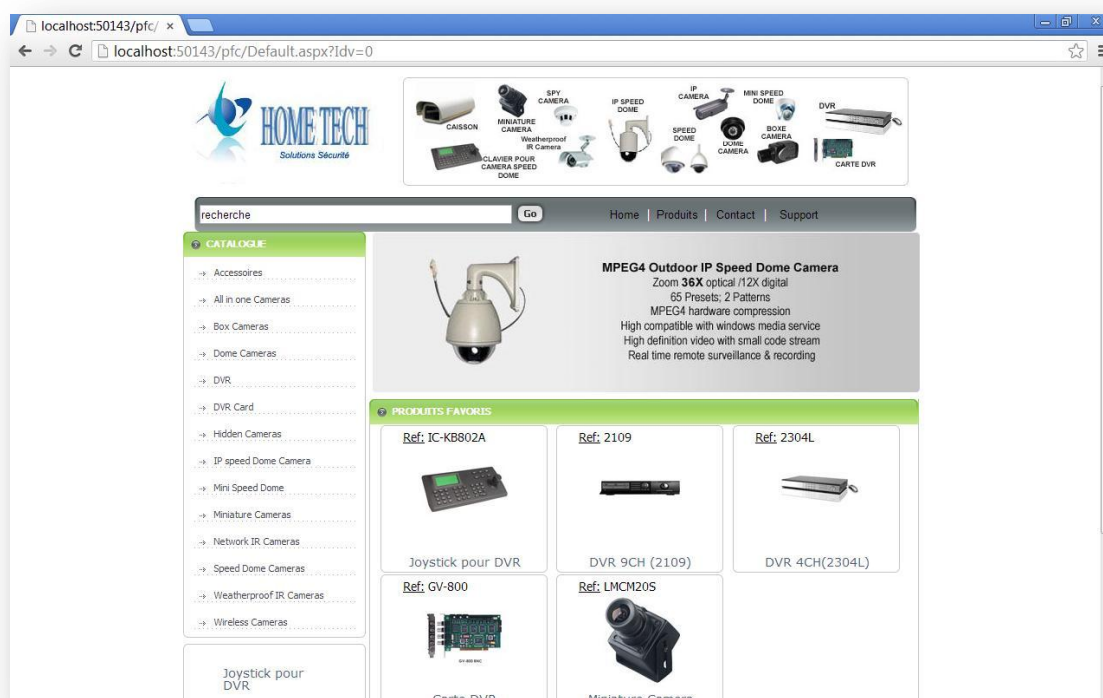


Figura 28 . Visualización de la página principal en Google chrome

La resolución de pantalla que hemos utilizado es de 1280x800 píxeles, ésta es la resolución ideal para visualizar la página de la mejor forma.

Como podemos ver en las tres últimas figuras, la página Web se visualiza perfectamente en los tres navegadores (está centrada y sin errores) aunque notamos que en Internet Explorer se ve un poquito más grande.

Además de la visualización de la página principal en diferentes navegadores, hemos realizado otras pruebas para comprobar el correcto funcionamiento del sitio Web:

- Comprobar que sólo los administradores registrados pueden acceder a la zona reservada para la gestión de la base de datos.
- Insertar información en todas las tablas de la base de datos para poder probar el funcionamiento de todos los casos de uso.
- Verificar que todos los enlaces en la página funcionan de manera correcta.

7. Conclusiones

La realización de este proyecto final de carrera nos ha permitido aprender todos los pasos que se deben seguir para el desarrollo de una aplicación Web, hemos puesto en práctica muchos de los conocimientos adquiridos durante la carrera, además, hemos aprendido muchas tecnologías nuevas (sobre todo relacionadas a la web) que nos serán muy útiles en futuros proyectos.

Al final del proyecto hemos obtenido un sitio Web que cumple todos los requisitos impuestos al inicio del mismo. El sitio web es muy simple y sólo ofrecerá información sobre los productos ofrecido por una empresa, no es una solución de comercio electrónico donde el cliente pueda realizar sus compras online, esta vía se deja abierta para posibles ampliaciones de funcionalidades.

Al final, podemos decir que la realización de este proyecto ha sido muy positiva porque nos ha permitido aprender muchas cosas nuevas y crecer profesionalmente. Todo eso nos ayudará en nuestra vida laboral como informáticos.

8. Bibliografía

- Programación con ASP.NET 2.0/ Jesse Liberty y Dan Hurwitz
- Desarrollo de aplicaciones Web con ASP.NET 2.0/ Antonio Martín Sierra
- Proyectos profesionales C#/ Arora, Geetanjali
- Página de Microsoft : <http://msdn.microsoft.com>
- Índice de las etiquetas HTML:
<http://www.virtualnauta.com/es/html/html-guia-de-referencia-etiquetas.php>
- Apuntes de la asignatura 'Ingeniería de la programación'
- Proyectos de años anteriores

9. Anexo

9.1 Crear una base de datos en visual studio:

Con las herramientas de administración de datos de visual studio podemos crear bases de datos de SQL Server y rellenarlas con datos y tablas.

Para crear la base de datos de la aplicación Web seguimos los siguientes pasos:

1. Abrir el explorador de soluciones de visual studio, hacer un clic con el botón derecho del ratón sobre la carpeta App_Data y elegir la opción Agregar nuevo elemento.
2. Hacer clic en Base de datos SQL, escribir el nombre de la base de datos y luego pulsar el botón Agregar.
Visual studio crea dos archivos, uno con la extensión .mdf y el otro con la extensión .ldf y luego abre la ventana de Explorador de servidores seleccionando la base de datos que se acaba de crear.

Una vez creada la base de datos, podemos agregar tablas y datos:

1. En la ventana Explorador de servidores, hacer doble clic en conexiones de datos y luego expandir la base de datos.
2. Hacer clic con el botón derecho del ratón en la carpeta Tablas y seleccionar Agregar nueva tabla. El foco se desplaza automáticamente hacia el panel de definición de una nueva tabla.
3. Agregar las columnas deseadas a la tabla, especificando el nombre de cada columna, el tipo de dato que contiene y si este dato admite valor nulo o no.
4. Hacer clic con el botón derecho del ratón en la columna que va a ser la clave principal de la tabla, y, a continuación seleccionar Establecer clave principal.
5. De manera opcional, seleccionar una columna de tipo entero para que sea la columna de identidad, luego, en el panel Propiedades de columna correspondiente a la columna seleccionada, hacer doble clic en la sección Especificación de identidad y establecer un sí como valor de Identidad.

6. Presionar CTRL+S para dar un nombre a la tabla y luego guardarla. Si cerramos el panel de definición de tabla sin proporcionarle un nombre, se nos pedirá especificar un nombre y la tabla se guardará automáticamente.

Para rellenar una tabla de datos:

1. En la ventana Explorador de servidores, expandir primero la sección Conexiones de datos y luego la base de datos con la que se quiere trabajar, a continuación, hacer doble clic en la carpeta Tablas.
2. Hacer clic con el botón derecho del ratón en la tabla en la que se quiere agregar datos, y, a continuación, seleccionar mostrar datos de tabla.
3. En la tabla que aparece, rellenar las columnas con los datos que se desee insertar.