

PROYECTO FINAL DE CARRERA

# DISEÑO E IMPLEMENTACIÓN DE UNA TIENDA VIRTUAL

CURSO 2009/2010

Código proyecto: DISCA 29-17A /09



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

**Autor**

Francisco Aroca Martínez

**Director**

Sergio Sáez Barona

## Agradecimientos.

Quiero dar las gracias a mi mujer y a mi hija por su apoyo y comprensión incondicional durante estos tres años. Sabré recompensaros por vuestra paciencia. Os quiero.

## Índice general

1. INTRODUCCIÓN.....	5
2. ESPECIFICACIÓN DE REQUISITOS.....	7
2.1. INTRODUCCIÓN.....	7
2.1.1. PROPÓSITO.....	7
2.1.2. ÁMBITO.....	7
2.1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	8
2.1.4. REFERENCIAS.....	9
2.1.5. VISIÓN GLOBAL.....	9
2.2. DESCRIPCIÓN GENERAL.....	9
2.2.1. PERSPECTIVA DEL PRODUCTO.....	9
2.2.2. FUNCIONES DEL PRODUCTO.....	10
2.2.3. CARACTERÍSTICAS DEL USUARIO.....	10
2.2.4. RESTRICCIONES GENERALES.....	10
2.2.5. SUPUESTOS Y DEPENDENCIAS.....	11
2.3. REQUISITOS ESPECÍFICOS.....	11
2.3.1. REQUISITOS DE INTERFACES EXTERNAS.....	11
2.3.2. REQUISITOS FUNCIONALES.....	12
2.3.3. REQUISITOS DE RENDIMIENTO.....	16
2.3.4. RESTRICCIONES DE DISEÑO.....	16
2.3.5. ATRIBUTOS.....	17
3. ANÁLISIS.....	18
3.1. INTRODUCCIÓN.....	18
3.2. DIAGRAMA DE CLASES.....	18
3.3. CASOS DE USO.....	19
3.4. DIAGRAMAS DE SECUENCIA.....	22
4. DISEÑO.....	29
4.1. INTRODUCCIÓN.....	29
4.2. ARQUITECTURA MULTICAPA.....	29
4.2.1. NIVEL DE PRESENTACIÓN.....	30
4.2.2. NIVEL DE LÓGICA DE LA APLICACIÓN.....	33
4.2.3. NIVEL DE DATOS O PERSISTENCIA.....	33

4.3. DIAGRAMA ENTIDAD RELACIÓN.....	34
4.4. DISEÑO LÓGICO.....	35
5. IMPLEMENTACIÓN.....	37
5.1. TECNOLOGÍAS.....	37
5.1.1. HTML.....	37
5.1.2. CSS.....	38
5.1.3. JAVASCRIPT.....	38
5.1.4. PHP.....	38
5.1.5. MYSQL.....	39
5.2. HERRAMIENTAS.....	39
5.3. DETALLES DE IMPLEMENTACIÓN.....	39
5.3.1. IDENTIFICACIÓN.....	39
5.3.2. ACCESO RESTRINGIDO.....	41
5.3.3. LOGOUT.....	43
5.3.4. COMPROBACIÓN DE CAMPOS.....	43
5.3.5. VALIDACIÓN DEL CORREO ELECTRÓNICO.....	44
5.3.6. VALIDACIÓN DEL CÓDIGO POSTAL.....	44
5.3.7. MODIFICAR ESTADO PEDIDO.....	45
6. EVALUACIÓN Y PRUEBAS.....	47
6.1. EVALUACIÓN.....	47
6.2 PRUEBAS.....	47
6.2.1 PRUEBAS DE VALIDACIÓN.....	47
6.2.2. COMPROBACIÓN DE ENLACES ROTOS.....	48
6.2.3. COMPROBACIÓN CON OTROS NAVEGADORES.....	49
6.2.4. OTRAS PRUEBAS.....	50
7. CONCLUSIÓN.....	62
8. BIBLIOGRAFÍA.....	64

# 1. Introducción.

Nuestro objetivo va a ser el desarrollo de una tienda virtual enfocada a la venta de productos informáticos. La aplicación a desarrollar, Info-Star ofrecerá la funcionalidad que disponen las tienda en la red, dispondrá de un catálogo de productos de material informático donde se podrá visualizar sus características técnicas más importantes y por supuesto tendremos la opción de poder comprarlos.

Nuestra aplicación tendrá clasificados los productos por categorías, además para poder facilitar la localización de nuestro material informático, se dispondrá de una pequeña aplicación que realizará la búsqueda por palabras clave.

Dependiendo del tipo de usuario se dispondrá de unas opciones o de otras, tendremos tres tipos de usuarios.

El usuario anónimo será el que menos funcionalidades tenga, podrá navegar por nuestra web, ver productos y añadirlos al carro, pero no podrá efectuar la finalización de la compra hasta que no se registre o se identifique, una vez realizada cualquiera de estas operaciones se convertirá en usuario registrado el cual ya podrá finalizar la compra.

El usuario con más privilegios será el usuario administrador, que además de disponer de las funcionalidades anteriores podrá gestionar el catálogo de productos, administrar clientes, categorías...

En resumen se trata de desarrollar una aplicación para facilitar la venta de material informático así como ayudar a su gestión, de una forma sencilla y clara para los usuarios y el administrador de la aplicación.

Los objetivos concretos consistirán en:

- Mostrar un catálogo de los productos a nuestros posibles clientes.
- Permitir la compra de los productos que aparezcan en nuestra aplicación.
- Facilitar el mantenimiento de dicho catálogo.

Los resultados y productos que se piensan obtener:

- Una aplicación web para dar a conocer nuestros productos.
- Facilitar la gestión tanto del material como de los usuarios.

El método para guiar el desarrollo del proyecto se basará en las siguientes etapas:

- La especificación de requisitos basada en la definición de casos de uso.
- El proceso de análisis basado en modelos UML.
- La definición de la arquitectura basada en tres capas.

## Diseño e implementación de una tienda virtual

- La codificación de un prototipo de la aplicación.
- La realización de pruebas técnicas del sitio Web.

Los recursos disponibles consistirán en:

- Un entorno de diseño Web basado en herramientas convencionales. WAMP el cual incluye PHP5, MySQL y Apache.
- Un servidor Web.
- Un plazo de dos meses.
- Un alumno con dedicación completa.

## 2. Especificación de requisitos.

Nuestra memoria correspondiente al desarrollo de una aplicación de una tienda virtual de productos informáticos, se va a desarrollar en base al estándar de especificación de requisitos de software IEEE 830/98.

### 2.1. Introducción.

En este apartado vamos a pasar a describir, el propósito, su ámbito, definiciones, acrónimos y abreviaturas, las referencias de la especificación y la visión global de nuestro proyecto.

#### 2.1.1. Propósito.

La ERS (especificación de requisitos) nos servirá de base para desarrollar nuestra aplicación Web en forma de una tienda virtual de venta de productos informáticos, detallar los requisitos de diseño de interfaz, contenidos y funcionalidad de la misma. Así como la especificación de los objetivos que queremos conseguir.

#### 2.1.2. Ámbito.

Nuestro proyecto consistirá en la creación de una tienda virtual enfocada a la venta de productos informáticos, la cual estará dirigida hacia todo tipo de usuarios, tanto usuarios domésticos como a usuarios de ámbito profesional.

En nuestra Web Info-Star se podrá navegar dentro de un catálogo de productos, de forma libre sin necesidad de que el usuario se registre, estas consultas se harán a través de las distintas categorías existentes además de poder realizar una búsqueda, con una pequeña aplicación mediante palabras clave.

Dentro de las categorías se encontrarán todos los productos clasificados, será el administrador el responsable de realizar la gestión de este material así como su correcta organización.

Todos los productos constarán de una descripción, fotografía, precio y se podrá añadir a nuestro carrito de la compra, para efectuar la compra de cualquier producto añadido a nuestro carrito, el usuario deberá darse de alta en nuestra base de datos o si ya lo está, identificarse.

La gestión de nuestros Web será realizada por un usuario especial, el administrador, en el cual recaerá la responsabilidad de realizar el mantenimiento del catálogo de productos, altas y bajas de material, así como de gestionar pedidos y usuarios.

### 2.1.3. Definiciones, acrónimos y abreviaturas.

**Tienda virtual:** Este término hace referencia a la aplicación informática capaz de ofrecer acceso a productos donde estos se pueden comprar y pagar por mediación de cualquier medio electrónico a través de la red. Donde se suelen dar servicios muy similares a los de una tienda física

**Login:** Es el nombre con el que se identifica a un usuario, que con anterioridad ha realizado un proceso de registro, suele formarse con un código alfanumérico de varios dígitos. Dependiendo de la aplicación se le pueden asignar distintos roles y accesos.

**Password:** palabra de paso o clave de acceso, permite el acceso a una persona, se le pueden asignar permisos. Compuesta al igual que el *login* por un conjunto alfanumérico de varios dígitos.

**Usuario anónimo:** Usuario que visita el portal y del cual no se tiene información de registro.

**Usuario registrado:** Usuario que ha realizado el proceso de registro y por lo tanto se dispone de información personal para identificarlo y personalizar su visita a la Web.

**Servidor:** Máquina que se encarga de ejecutar la aplicación para que los usuarios, clientes, puedan acceder a recursos y páginas.

**Cliente:** Cualquier elemento de un sistema de información que requiere un servicio mediante el envío de solicitudes al servidor.

**Navegador:** Aplicación para visualizar documentos WWW y navegar por Internet. En su forma más básica son aplicaciones hipertexto que facilitan la navegación por los servidores de navegación de Internet.

**Protocolo:** Conjunto de reglas y/o procedimientos para la transmisión de datos que ha de ser observado por los dos extremos de un proceso comunicacional.

#### **Acrónimos.**

**HTML:** *HyperText Markup Language*. Lenguaje de marcado de hipertexto, es el lenguaje estándar para describir el contenido y apariencia de las páginas en la Web.

**Web:** Por este término se conoce a WWW (*World Wide Web*), la red de redes, es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de internet.

**HTTP:** *Hyper Text Transfer Protocol*, protocolo de transferencia de hipertexto, es el método más común de intercambio de información en la Web.

**PHP:** Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas, usado en el lado del servidor.

**MySQL:** Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

**SQL:** Lenguaje de consulta estructurado, *structured query language*, es un lenguaje declarativo de acceso a base de datos relacionales que nos permite especificar diversos tipos de operaciones en estas.



**WAMP:** Usado para describir un sistema de infraestructuras de internet que usa, *Windows* como sistema operativo, Apache como servidor Web, MySQL como gestor de base de datos y PHP como lenguaje de programación.

**CSS:** *Cascading Style Sheets*, hojas de estilo en cascada, es un lenguaje usado para definir la presentación de un documento escrito en HTML o XML.

**JavaScript:** es un lenguaje de *scripting* basado en objetos no tipado y liviano, utilizado para acceder a objetos en aplicaciones, no requiere compilación.

**UML:** Unified Modeling Language, Lenguaje Unificado de Modelado, se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema software.

**IEEE:** *Institute of Electrical and Electronics Engineers*, es una asociación técnico-profesional mundial dedicada a la estandarización.

#### 2.1.4. Referencias.

WORLD WIDE WEB CONSORTIUM (W3C). Documentos, guías y estándares. [Consulta: 02 de noviembre de 2010]. Disponible en: <http://www.w3c.es>

WIKIPEDIA LA ENCICLOPEDIA LIBRE. Definiciones. [Consulta: 05 de noviembre de 2005]. Disponible en: <http://es.wikipedia.org>

REAL ACADEMIA ESPAÑOLA. Ortografía. [Consultas: 08 de noviembre de 2010]. Disponible en: <http://rae.es>

Buendía García, Félix. Guía para la realización y supervisión de proyectos de final de carrera (PFC) en el ámbito de la web. Valencia: Editorial UPV. ISBN 9788483633250.

APUNTES ISG. Esquemas y división proyecto. [Consulta: 02 de noviembre de 2010]. Disponible en: <http://poliformate.upv.es>

#### 2.1.5. Visión global.

Vamos a centrar la especificación de requisitos en la descripción de la aplicación, sus características, sus restricciones generales, sus funciones, sus supuestos y dependencias, que son las que nos aportarán una mayor información del proyecto a desarrollar.

Cuando terminemos de desarrollar la descripción general, pasaremos a la descripción de los requisitos específicos de nuestra aplicación.

### 2.2. Descripción general.

En la descripción general se aborda la perspectiva del producto, los distintos usuarios que utilizarán la aplicación, restricciones, supuestos y dependencias de nuestro proyecto.

#### 2.2.1. Perspectiva del producto.

Info-Star, nuestra aplicación, realizará las funcionalidades normales que implementa cualquier tienda virtual, venta y búsqueda de productos, en nuestro caso material informático, la gestión y administración tanto de productos como de usuarios y pedidos.

## 2.2.2. Funciones del producto.

Clasificaremos las funciones de nuestra aplicación en bloques.

**Funciones de búsqueda.** Cualquier usuario que acceda a nuestro portal Web, este registrado o no, podrá realizar búsquedas de material en el catálogo a través de las diferentes categorías definidas así como una búsqueda automatizada. Cuando el usuario encuentre el producto se mostrará la imagen del mismo, con su precio, opción para visualizar sus detalles y añadir a al carrito de la compra.

**Funciones de compra.** Para poder finalizar la compra de un producto el usuario deberá identificarse o en el caso que no esté registrado realizar la operación de registro.

**Funciones de gestión.** Las funciones de gestión solo podrán llevarse a cabo por el administrador, será el encargado de realizar altas y bajas de productos así como de la edición o rectificación de estos. También gestionará las distintas categorías del catálogo de productos, creando o eliminando según crea conveniente, además de poder eliminar usuarios, consultar pedidos y modificar su estado, así como listar productos, usuarios, categorías y pedidos.

## 2.2.3. Características del usuario.

Nuestro sistema constará de tres tipos de usuarios que son los que podrán interactuar con nuestra Web: Usuario anónimo, usuario registrado y usuario administrador.

**Usuario anónimo:** Podrá realizar consultas al catálogo de productos y navegar de forma libre por la zona pública de la Web. Ver productos, descripciones de productos...

**Usuario registrado:** Es aquel usuario que habrá efectuado su alta en el sistema y que por lo tanto sus datos están en nuestra base de datos. Estos usuarios podrán realizar las funciones de usuario anónimo y podrán efectuar finalizar las compras añadidas al carrito de la compra.

**Usuario administrador:** Este usuario será el encargado de realizar todas las funciones de gestión del sistema, modificaciones, listados, mantenimiento de la base de datos y comprobación del buen funcionamiento de la Web.

## 2.2.4. Restricciones generales.

Las consideraciones de seguridad que existirán en nuestra aplicación serán la restricción de control y acceso a la información, ya que dependiendo de nuestro *login* y *password* dispondremos de unos privilegios para poder realizar determinadas funciones.

Será muy importante cumplir con la ley de protección de datos para así mantener la información de nuestra base de datos protegida ante accesos no deseados. Esto será requisito indispensable en nuestro servidor que será donde se almacene toda la información delicada.

En el parte del cliente cualquier equipo sobremesa o portátil, con una configuración media baja, que soporte un navegador Web estándar serían suficientes para poder ser utilizados, ya que los requisitos necesarios para visualizar nuestra aplicación son escasos.

Por el lado del servidor dependeremos del número de clientes que estén conectados de forma simultánea, ya que esto haría que nuestro servidor funcionara de forma inadecuada. La solución sería aumentar la potencia del servidor para evitar la saturación debida a la ejecución de mucho código, muchos accesos a nuestra base de datos...

Otro posible problema del lado del servidor, de fácil solución actualmente, es el ancho de banda para servir las páginas, solución contratar mayor ancho de banda.

Algo muy importante son las políticas de seguridad en el servidor, así evitaremos posibles intrusiones en el sistema. Es fundamental que la configuración de nuestro servidor Apache este optimizada al máximo. No dejar la configuración que viene por defecto cuando instalamos Apache, cambios de contraseña y número de puerto de acceso son fundamentalmente acciones que debemos realizar para optimizar la seguridad.

Las copias de seguridad de la base de datos, páginas del servidor y configuraciones del servidor, se deben realizar de forma periódica.

### 2.2.5. Supuestos y dependencias.

Nuestra aplicación Info-Star, se podrá ejecutar en cualquier sistema operativo actual, basado en tecnología *Windows* o *Unix*. Ya que vamos a utilizar para su desarrollo una aplicación del tipo WAMP, *Windows Apache MySQL PHP*.

De esta manera con una única aplicación nos facilitará el trabajo y configuración de todos sus componentes, evitaremos las instalaciones múltiples que pueden resultar complicadas y a veces incompatibles entre ellas.

Para poder efectuar las pruebas es necesario instalar Apache en la máquina funcionará como servidor. En nuestro caso el servidor tendrá instalado *Windows 7*. El desarrollo de la aplicación se realizará con *HTML* y *PHP*, para su edición se utilizará *Dreamweaver* y *NetBeans*. Para la gestión de la base de datos se utilizará *PhpMyAdmin*.

## 2.3. Requisitos de específicos.

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema.

### 2.3.1. Requisitos de interfaces externas.

#### **Interfaces de usuario.**

En la interfaz de usuario de nuestra aplicación inicialmente se accederá a una página principal donde en el lado izquierdo aparecerán todas las categorías de nuestro catálogo de productos, en la parte central tendremos un listado con artículos, que cambiará a la descripción de cada producto si se le pulsa en la opción de detalles, recaerá sobre la imagen del producto.

El apartado de carrito nos permitirá visualizar los productos que hemos ido añadiendo, esta opción nos dejará eliminar artículos o cambiar las unidades del mismo.

Dispondremos de un apartado que nos permitirá identificarnos, y en el caso de que el usuario no sea socio, le permitirá realizar su respectivo registro enviando un formulario y procediendo a su alta.

Según los permisos de cada tipo de usuario, se podrá acceder a las distintas opciones de la aplicación.

Se dispondrá de una aplicación que nos facilitará la búsqueda de productos, introduciendo una palabra buscará los artículos que la incluyan.

Diseño e implementación de una tienda virtual

Por último dispondremos de un logout para poder desconectarnos de la sesión.

### **Interfaces *hardware*.**

Los clientes necesitarán una máquina que disponga de acceso a internet, actualmente cualquier equipo que soporte un navegador Web, con una tarjeta de red, sería lo recomendado, o bien un modem.

La máquina que realizará la función de servidora dispondrá de un repositorio en cual almacenará toda la información del catálogo de productos además de la base de datos con la información de los usuarios registrados. Esta máquina necesitará una conexión a internet con una banda ancha mayor, para así poder asistir las peticiones de los clientes.

### **Interfaces *software*.**

Nuestra aplicación va a ser desarrollada bajo el sistema operativo *Windows 7*, la versión *ultimate*, sobre nuestro *Windows* instalaremos nuestro sistema WAMP, que incorporará Apache como servidor, la versión 2.2.12, MySQL versión 5.1.37 como gestor de base de datos, el cual incluye una interfaz amigable llamada PHPMyAdmin versión 3.2.0.1 y como lenguaje de implementación PHP versión 5.3.0.

Para el desarrollo del interfaz web utilizaremos Dreamweaver que nos ayudará con el HTML, las hojas de estilo CSS y con PHP.

Por último NetBeans nos ayudará a programar en PHP.

Los clientes que accedan a nuestra Web no será necesario que dispongan de un sistema operativo concreto, el navegador deberá estar actualizado a las últimas versiones para evitar visualizaciones erróneas o falta de *plug-ins*.

Nuestra aplicación va a ser desarrollada con una arquitectura de tres capas.

### **Interfaces de comunicaciones.**

Los protocolos de comunicaciones van a ser TCP/IP entre los clientes y la Web y la comunicación entre los navegadores y el servidor HTTP.

Los usuarios realizarán su comunicación a través de Internet y el administrador tendrá la posibilidad de trabajar vía Internet o en local.

## **2.3.2. Requisitos funcionales.**

### **Función de autenticación de usuario (funcionalidad del usuario y administrador)**

**Introducción:** Mediante la introducción de usuario y contraseña, los usuarios registrados y el administrador accederán a la aplicación y a sus características asociadas.

**Entradas:** Tanto para el campo de usuario y contraseña se requerirán códigos alfanuméricos.

**Proceso:** Se realizará una comprobación de que el usuario su contraseña coinciden dentro de nuestra base de datos, si su identificación es correcta se mostrará una pantalla de bienvenida.

## Diseño e implementación de una tienda virtual

En caso contrario aparecerá una pantalla con un aviso de acceso incorrecto y se dará la posibilidad de registro.

**Salida:** en el caso que la identificación sea correcta nos llevará a una pantalla con un mensaje de bienvenida pudiendo acceder a las opciones específicas del usuario correspondiente. En caso negativo nos aparecerá una pantalla de advertencia.

### **Funciones de visualización de productos (funcionalidad de cualquier usuario)**

**Introducción:** Una vez seleccionada una categoría nos aparecerá un listado de productos, el usuario selecciona uno de los productos de la lista para visualizarlos.

**Entradas:** Categoría y producto seleccionados.

**Proceso:** Nuestra aplicación abrirá el correspondiente vínculo que hace referencia a las características y al producto seleccionado dentro de la categoría, todas las características del producto serán recuperadas del correspondiente repositorio.

**Salida:** En la pantalla del cliente aparecerán todos los detalles del producto seleccionado.

### **Funciones de registro de usuario (funcionalidad del usuario anónimo)**

**Introducción:** El nuevo usuario tendrá acceso a un formulario donde deberá introducir sus datos para darse de alta en nuestra aplicación.

**Entradas:** Se introducirán en los campos del formulario tanto datos personales como nombre de usuario, contraseña y correo electrónico.

**Proceso:** La primera parte del registro realizará una comprobación de todos los campos del formulario a rellenar, que no sean espacios en blanco, que los campos no estén vacíos, comprobación del correo y clave, la segunda parte realizará la verificación del nombre de usuario, para evitar duplicidades, si existiera lanzaremos un mensaje de advertencia, para finalmente insertar una tupla, con estos datos en nuestra base de datos.

**Salida:** Un mensaje diciendo que la inserción se ha efectuado con éxito.

### **Funciones de compra (funcionalidad del usuario registrado)**

**Introducción:** Desde esta función se podrá realizar una compra de los productos seleccionados, siempre que el usuario este registrado.

**Entradas:** El usuario selecciona una serie de productos del catálogo.

**Proceso:** Nuestro usuario podrá ir añadiendo productos a su carrito, será en el momento final de la compra cuando se verificará que el usuario este dado de alta, en caso que no sea así le enviará a una pantalla para que pueda hacerlo o bien le permita registrarse.

**Salida:** Aparecerá en la pantalla del cliente un mensaje de pedido realizado de forma correcta, asignando un número de pedido y factura a su compra por último terminará vaciando el carrito.

### **Función de añadir nueva categoría (funcionalidad el administrador)**

**Introducción:** Nuestro administrador a través de un formulario podrá introducir nuevas categoría dentro de nuestra aplicación.

**Entradas:** Una nueva línea con el nombre de la categoría.

**Proceso:** Una nueva entrada en la base de datos correspondiente con la nueva categoría.

**Salida:** Una nueva página con un mensaje de éxito.

### **Función de borrar categoría (funcionalidad el administrador)**

**Introducción:** Mediante un formulario nuestro administrador podrá eliminar categorías de nuestra aplicación.

**Entradas:** Debemos seleccionar la categoría a eliminar.

**Proceso:** Nuestro administrador deberá seleccionar una categoría a eliminar dentro del listado existente, para después confirmar pulsando el botón eliminar.

**Salida:** Mostraremos una página de éxito si la operación se ha podido llevar a cabo.

### **Función listar categorías (funcionalidad el administrador)**

**Introducción:** El administrador podrá efectuar una consulta de todas las categorías.

**Entradas:** Los valores para realizar la búsqueda.

**Proceso:** se efectuará una consulta en la tabla correspondiente a las categorías.

**Salida:** Se obtendrá en pantalla un listado de todas las categorías actuales.

### **Función añadir nuevo producto (funcionalidad el administrador)**

**Introducción:** Igual que en las anteriores opciones, el administrador a través de un formulario tendrá la opción de añadir un nuevo producto dentro de una categoría del catálogo existente.

**Entradas:** Todas las características del producto, nombre, marca, precio, descripción...

**Proceso:** Una vez introducidos todos los datos, realizaremos la inserción dentro de nuestra base de datos de productos.

**Salida:** Una página mostrando el éxito si se ha podido realizar la operación.

### **Función borrar producto (funcionalidad el administrador)**

**Introducción:** Nuestro administrador podrá efectuar el borrado de producto de nuestra aplicación.

**Entradas:** Deberemos seleccionar el producto a eliminar.

**Proceso:** Una vez seleccionada la opción de borrar producto, deberemos seleccionar el producto a eliminar del listado existente. Finalmente confirmaremos la operación con el botón de borrado.

**Salida:** Nuevamente aparecerá una página con un mensaje de operación realizada de forma correcta si se ha podido llevar a cabo.

### **Función listar productos (funcionalidad el administrador)**

**Introducción:** El administrador podrá efectuar una consulta de todos los productos que estén datos de alta.

**Entradas:** Los valores para realizar la búsqueda.

**Proceso:** Se efectuará una consulta en la tabla de productos.

**Salida:** Se obtendrá en pantalla un listado de todos los productos que se dispongan en la tabla de productos.

### **Función listar usuarios (funcionalidad el administrador)**

**Introducción:** El administrador podrá efectuar una consulta de todos los usuarios que estén registrados.

**Entradas:** Los valores para realizar la búsqueda.

**Proceso:** Se efectuará una consulta en la tabla correspondiente a los usuarios registrados.

**Salida:** Se obtendrá en pantalla un listado de todos los usuarios registrados.

### **Función eliminar usuario (funcionalidad el administrador)**

**Introducción:** Desde esta opción podremos efectuar el borrado de un usuario registrado.

**Entradas:** Los datos correspondientes al usuario que queremos eliminar.

**Proceso:** Deberemos seleccionar del listado de socios el usuario que deseamos borrar, una vez seleccionado se eliminará la tupla de nuestra tabla correspondiente.

**Salida:** Como es habitual lanzaremos una página con un mensaje de éxito si se ha efectuado de forma correcta.

### **Función listar pedidos (funcionalidad el administrador)**

**Introducción:** Dentro de esta función se podrá obtener un listado de pedidos realizados.

**Entradas:** Los parámetros de búsqueda.

**Proceso:** Nuestra aplicación efectuará una consulta dentro de la tabla correspondiente.

**Salida:** El listado con los pedidos realizados.

### **Función cambiar estado pedido (funcionalidad el administrador)**

**Introducción:** El administrador una vez comprobado el listado de pedidos, podrá cambiar el estado de los pedidos.

**Entradas:** Información del estado de transferencias, mediante consulta manual de cuentas banco.

**Proceso:** Una vez el administrador verifique que se han efectuado las transferencias bancarias, por el importe de los productos solicitados, listará los pedidos y realizará el cambio de estado a pendiente, preparado, agotado o servido.

**Salida:** Volverá a la página principal de administración.

## **2.3.3. Requisitos de rendimiento.**

Inicialmente no se han determinado requisitos específicos de eficiencia, aunque serían necesarios algunos requerimientos para que nuestra aplicación se convirtiera en un producto cuya finalidad fuera su uso a nivel profesional.

## **2.3.4. Restricciones de diseño.**

Vamos a pasar a describir los estándares y las limitaciones de nuestro producto.

### **Estándares cumplidos.**

Nuestro producto debe cumplir las especificaciones actuales de cualquier página Web dinámica actual, tanto las de lenguaje HTML como las de los estándares seguidos por las hojas CSS. De esta forma tendremos la máxima compatibilidad con cualquiera de los navegadores modernos y los costes de mantenimiento de nuestra aplicación serán menores.

Deberemos cumplir con la actual ley de protección de datos, para asegurarnos de que la información que dispones en nuestra base de datos no es filtrada para ningún uso ilícito y asegurar al usuario poder dar de baja la información de carácter personal.

### **Limitaciones Hardware.**

Inicialmente las posibles limitaciones a nivel *hardware* residen en la conexión que se disponga en el lado del cliente, ya que actualmente es muy raro que algún equipo no cumpla las características mínimas para garantizar un funcionamiento correcto.



## Diseño e implementación de una tienda virtual

El sistema operativo utilizado tampoco debería ser ningún impedimento para la correcta visualización y funcionamiento de nuestra aplicación.

En cuanto al lado del servidor sus limitaciones vendrán dadas por la capacidad del mismo para satisfacer un número de accesos determinados de forma simultánea.

### 2.3.5. Atributos.

#### **Mantenimiento.**

El mantenimiento de nuestro sistema se realizará de una forma continuada por el administrador enviando sus peticiones al equipo de desarrolladores de la Web así como los posibles problemas que puedan ir surgiendo.

#### **Seguridad.**

Deberemos organizar un buen sistema de protección de información ya que en nuestra base de datos tendremos los datos personales de los usuarios que han efectuado su registro. Teniendo en cuenta que nuestro sistema funcionará en la red, donde es más viable un posible problema de seguridad.

## 3. Análisis

### 3.1. Introducción.

En este apartado vamos a abordar la fase de análisis, con ayuda de la metodología UML se desarrollará un modelo de la aplicación donde diferenciaremos a los actores que interactúan con los objetos del sistema mediante distintas relaciones.

Análisis por definición es el proceso de construcción de un modelo, o especificación detallada del problema del mundo real al que nos enfrentamos. Está desprovisto de consideraciones de diseño e implementación. Todo esto nos ayuda a crear un sistema robusto y mantenible.

La notación que se va a utilizar es la proporcionada por el estándar UML. En nuestro caso usaremos los diagramas de clases, casos de uso y diagramas de secuencia.

### 3.2. Diagrama de clases.

El primer paso va a ser la realización del diagrama de clases, es el diagrama principal para el modelado orientado a objetos. Representa las clases del sistema junto con sus relaciones estructurales agregación y asociación. También se definirán los atributos y operaciones.

Con el diagrama de clases entenderemos que funciones pueden ser llevadas a cabo y quien las puede realizar. Como se pueden desarrollar y que tipo de objetos interactúan entre sí.

Pasamos a describir nuestro diagrama.

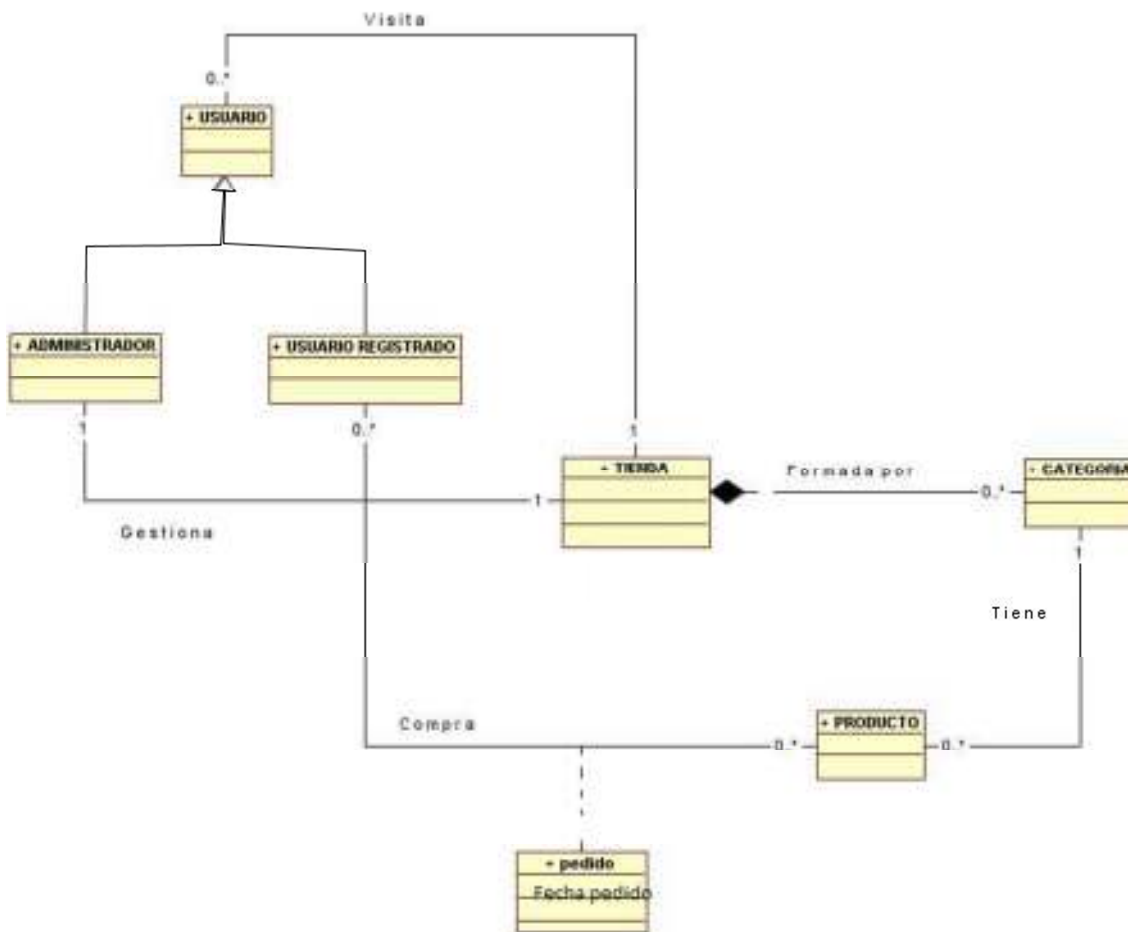
Tenemos tres tipos de usuarios, el usuario, que representará al usuario anónimo, es el más básico de ellos, solo podrá realizar consultas de productos y ver sus características asociadas.

El usuario registrado será una especialización del usuario corriente el cual tendrá la opción añadida de poder efectuar las compras de los productos.

Por otro lado tendremos el administrador que dispondrá de la posibilidad de gestionar el catálogo de productos, usuarios y listados de pedidos.

La clase principal de nuestra aplicación es Tienda, que se compone distintas categorías, las cuales a su vez están formadas por productos.

A continuación presentamos nuestro diagrama de clases, con las características descritas anteriormente:



### 3.3. Casos de uso.

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software.

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final

En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

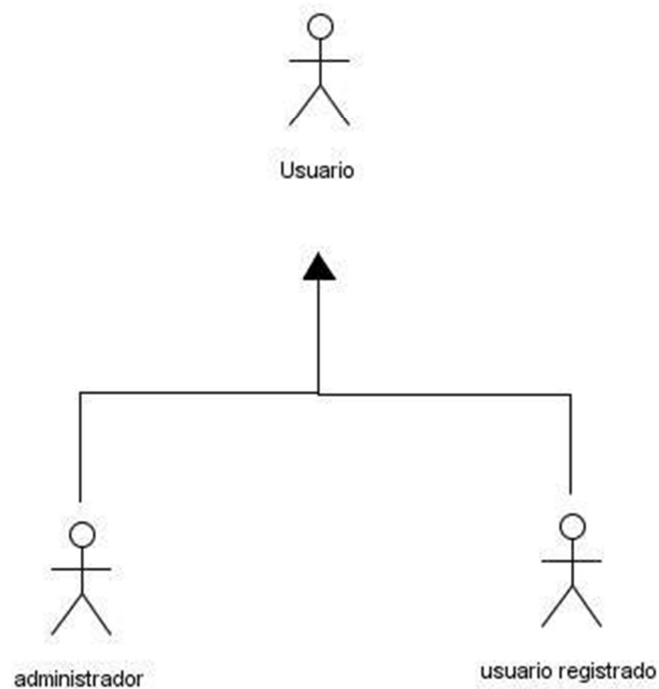
Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones.

## Diseño e implementación de una tienda virtual

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

Después de esta breve introducción de los casos de uso pasamos a mostrar cada uno de ellos.

Comenzamos por mostrar los actores que intervienen en nuestro sistema, que como ya sabemos son usuario, usuario registrado y administrador. Estos dos últimos especializan del usuario.

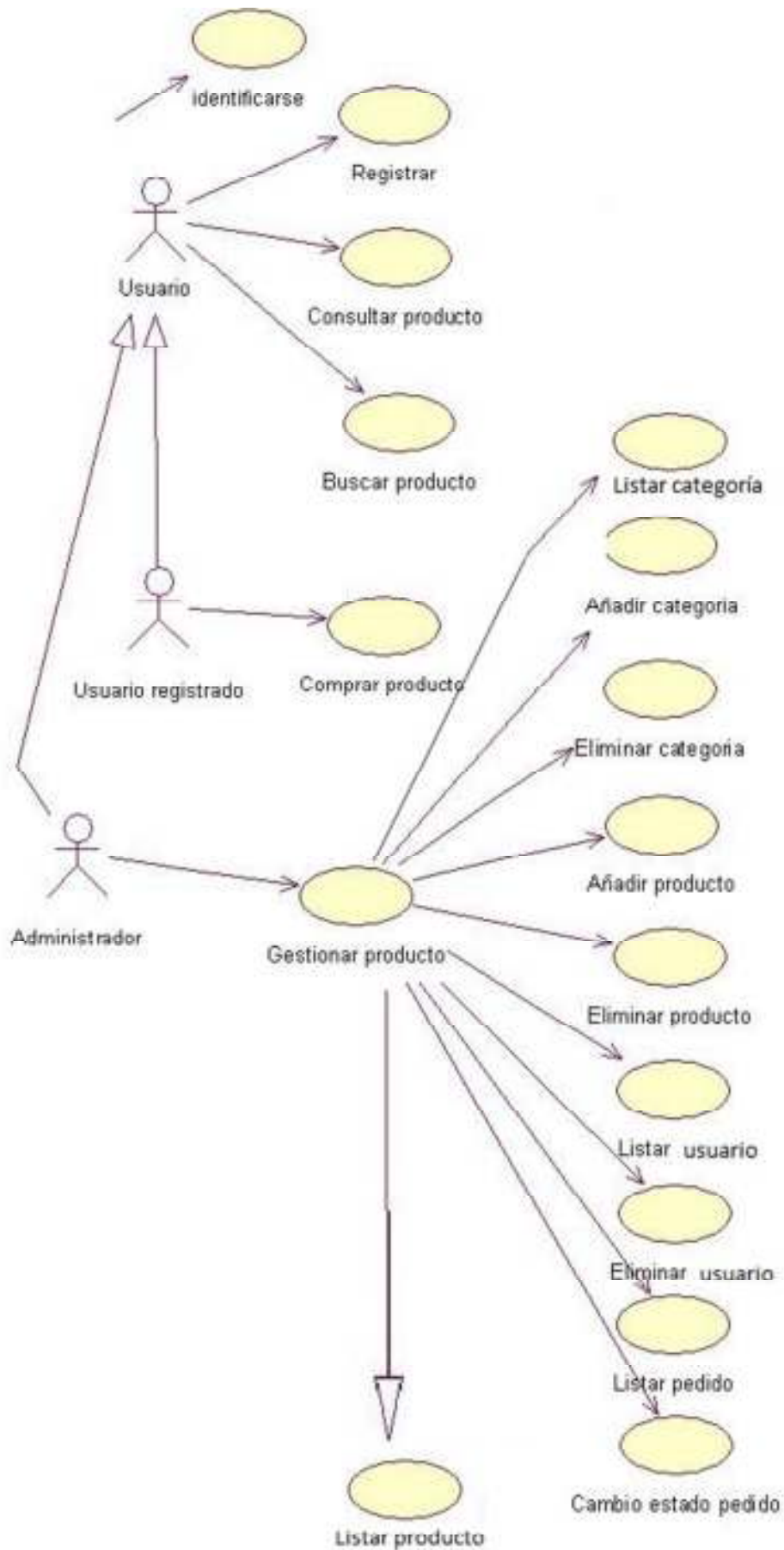


En el siguiente diagrama tenemos las funcionalidades de todos los actores, empezamos por el usuario, las funciones que podrá realizar serán, identificarse, registrarse, realizar búsqueda de productos y ver la información general de los productos del catálogo, es decir navegar a nivel general por la Web.

Además el usuario puede proceder al registro, convirtiéndose en usuario registrado.

El usuario registrado además de poder realizar todo lo comentado anteriormente se le añadirá la funcionalidad de realizar los pedidos de los productos que crea oportuno.

# Diseño e implementación de una tienda virtual



Por último el administrador dispone de las funcionalidades de gestión de productos, añadiendo, listando o quitando productos de nuestro listado, administración de categorías, pudiendo añadir nuevas categorías, listarlas o bien eliminar alguna de ellas, gestión de usuarios, teniendo la posibilidad de realizar un listado de usuarios registrados además de poder borrarlos, y por último gestión de pedidos, consistente en el cambio de estado del pedido y la posibilidad de listarlos.

### 3.4. Diagramas de secuencia.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. El diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

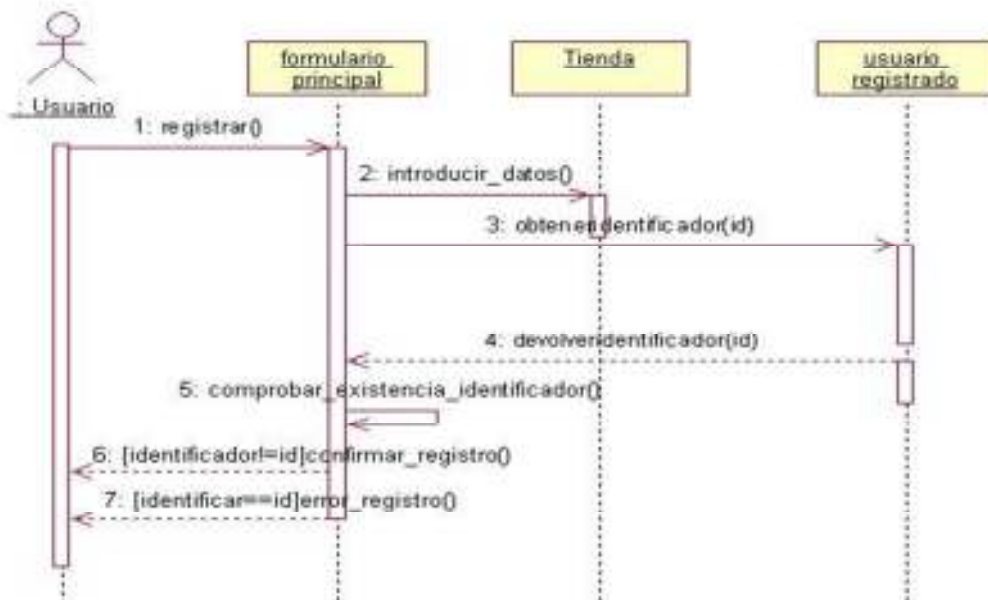
Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

#### Escenario: Registrar.

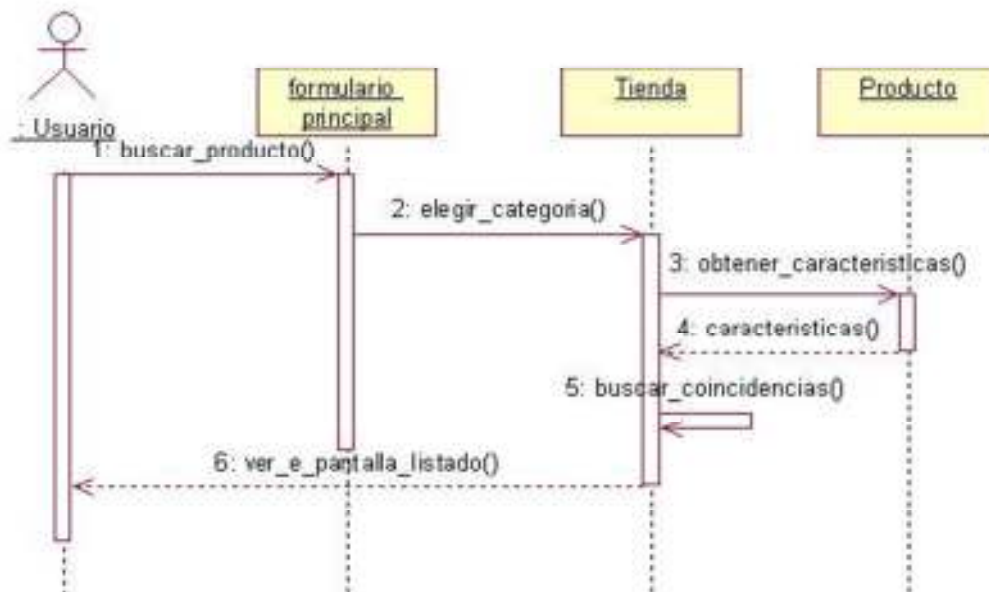
Desde esta opción, el usuario no registrado podrá darse de alta en la base de datos de nuestra aplicación.

Inicialmente se nos pedirá que rellenemos un formulario con los datos personales, así como nuestro identificador y contraseña. Se comprobará que el identificador elegido no este dado de alta por otro usuario en nuestra base de datos, si ya existe mostraremos un mensaje advirtiéndolo, en caso contrario se efectuará el alta, con su respectiva pantalla de confirmación.



**Escenario: Buscar producto.**

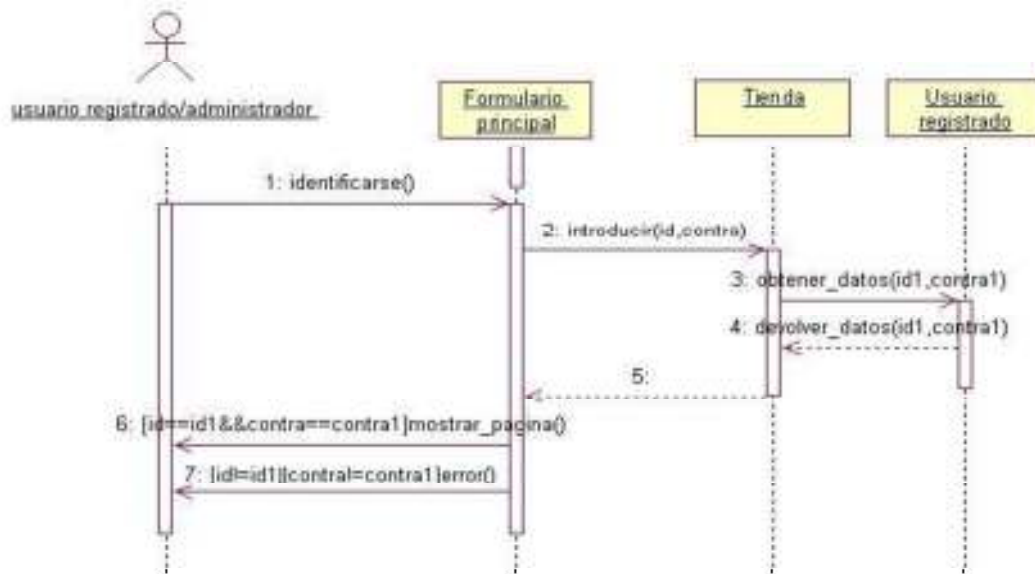
Desde esta opción cualquier usuario puede efectuar la búsqueda de un producto. Se buscarán las coincidencias dentro de la categoría seleccionada y se mostrarán los resultados en un listado.



**Escenario: identificarse.**

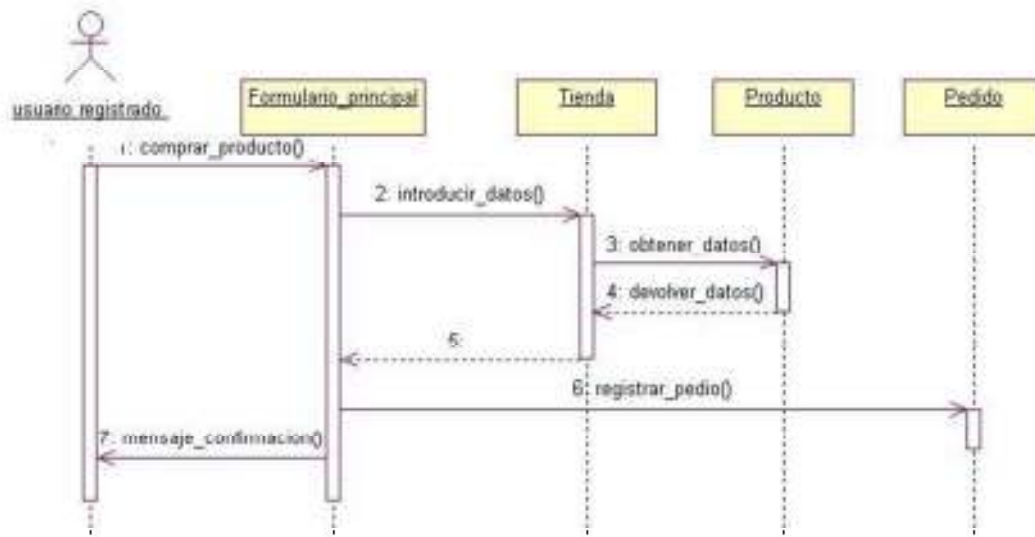
En la pantalla principal nos aparecerá la opción de identificación, para que los usuarios registrados o los usuarios administradores puedan introducir su identificador y correspondiente contraseña, para poder tener la funcionalidad correspondiente a su rol.

## Diseño e implementación de una tienda virtual



### Escenario: Comprar producto.

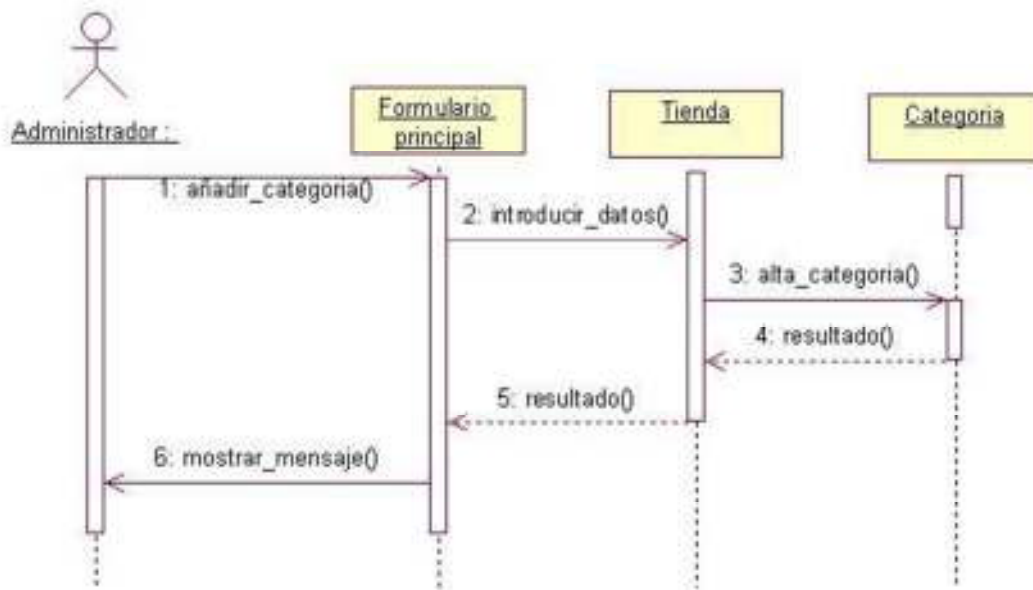
Una vez seleccionados los productos a comprar, se confirma que se quiere realizar la compra, se guardan los datos del pedido en nuestra base de datos se envía un mensaje de confirmación o en caso contrario un mensaje de advertencia.



### Escenario: alta categoría.

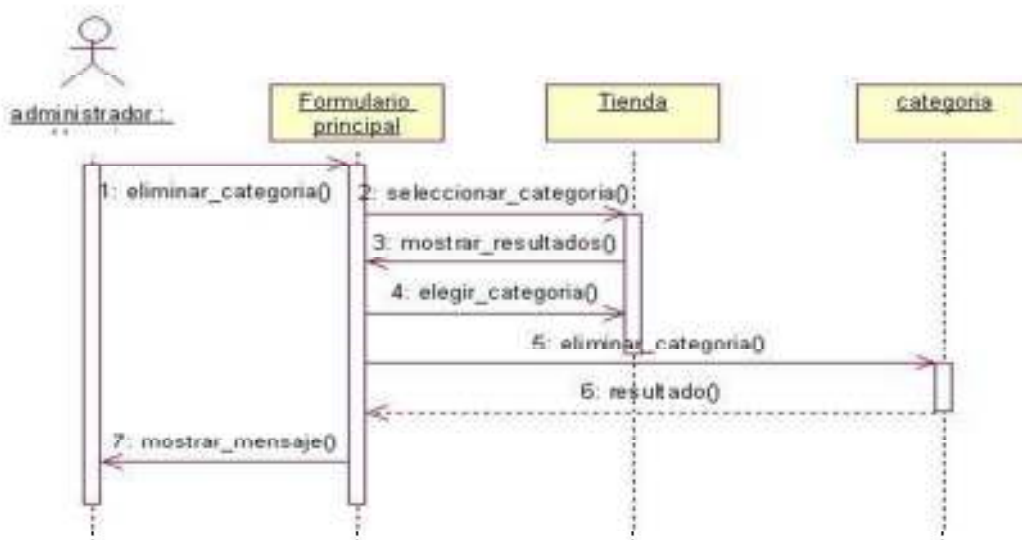
El administrador una vez identificado, tiene la opción de introducir nuevas categorías en nuestro catálogo de productos. Para ello introducirá el nombre de la nueva categoría y se realizará el alta en nuestra base de datos.





**Escenario: baja categoría.**

Al igual que la operación de alta será el administrador el encargado de efectuar las bajas de categorías. Se seleccionará la categoría que queremos borrar de nuestra base de datos y pulsando el botón de eliminar efectuaremos la operación indicada.

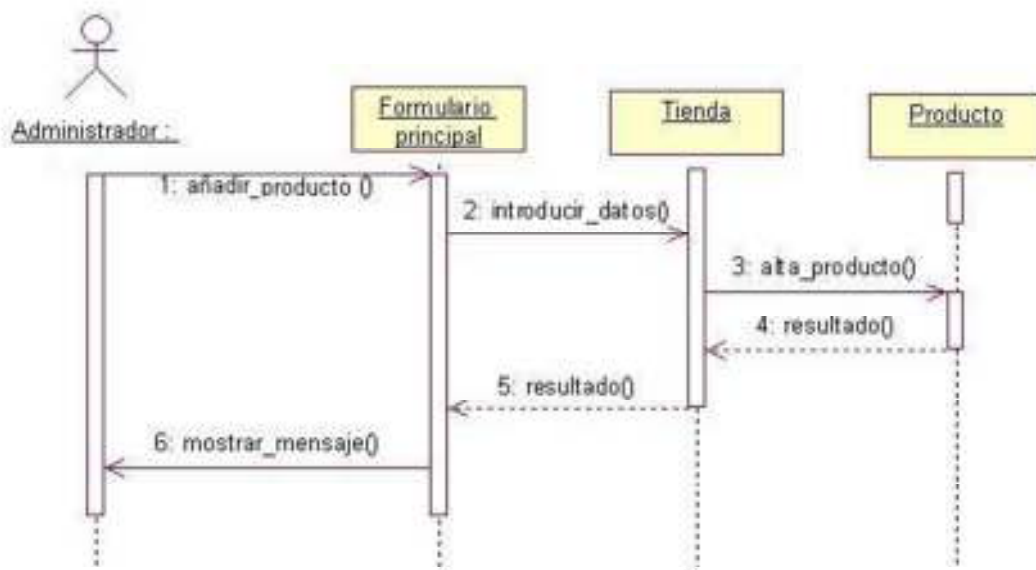


**Escenario: alta producto.**

Nuestro administrador, tendrá la función de dar de alta productos, para realizar esta operación deberá rellenar un pequeño formulario con el nombre del producto, precio, características...

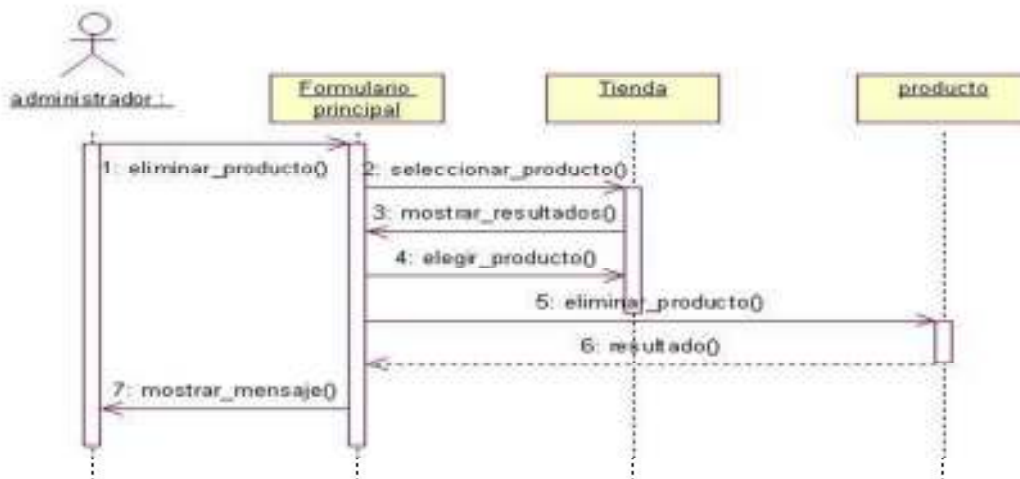
## Diseño e implementación de una tienda virtual

Una vez enviado el formulario se procederá al alta del producto.



### Escenario: Baja producto.

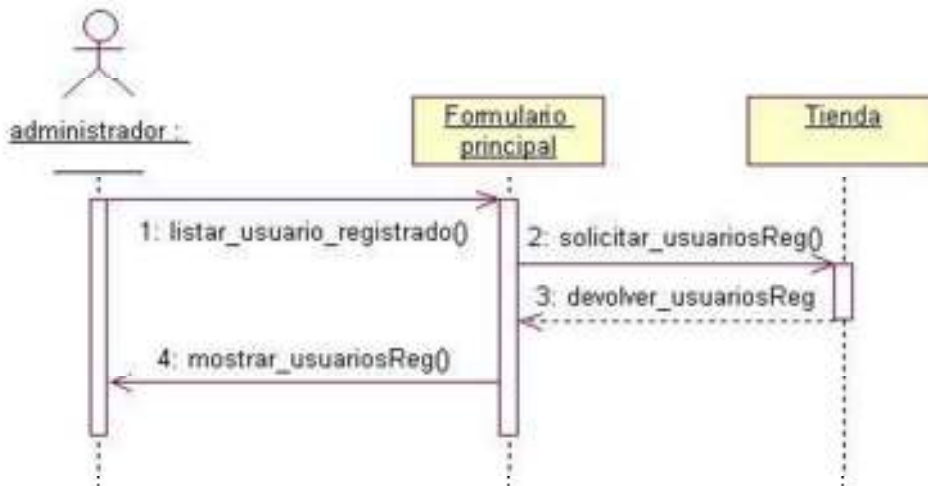
Esta operación consiste en seleccionar un producto y proceder a su eliminación de nuestra base de datos.



### Escenario: Listar usuarios registrados.

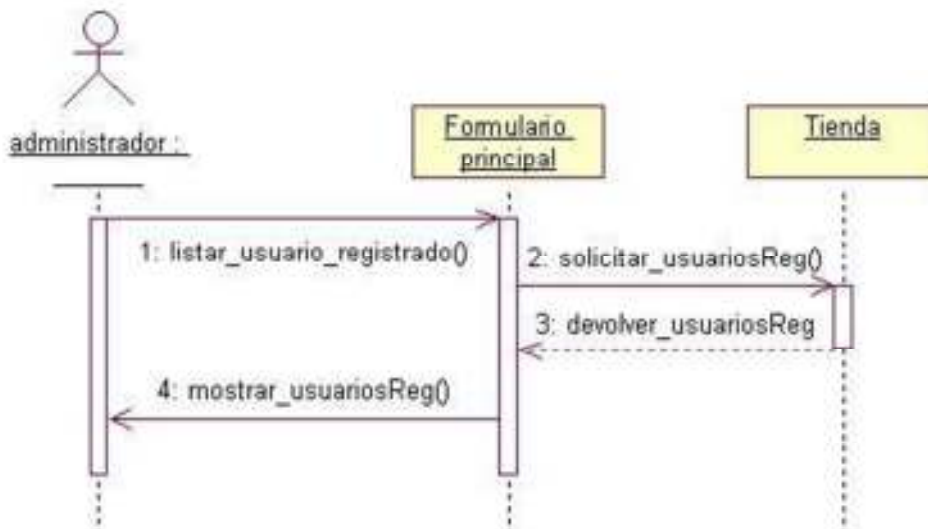
Dentro de esta opción el usuario realiza una petición de todos los usuarios que están registrados dentro de nuestra base de datos.

## Diseño e implementación de una tienda virtual



### Escenario: Eliminar usuario registrado.

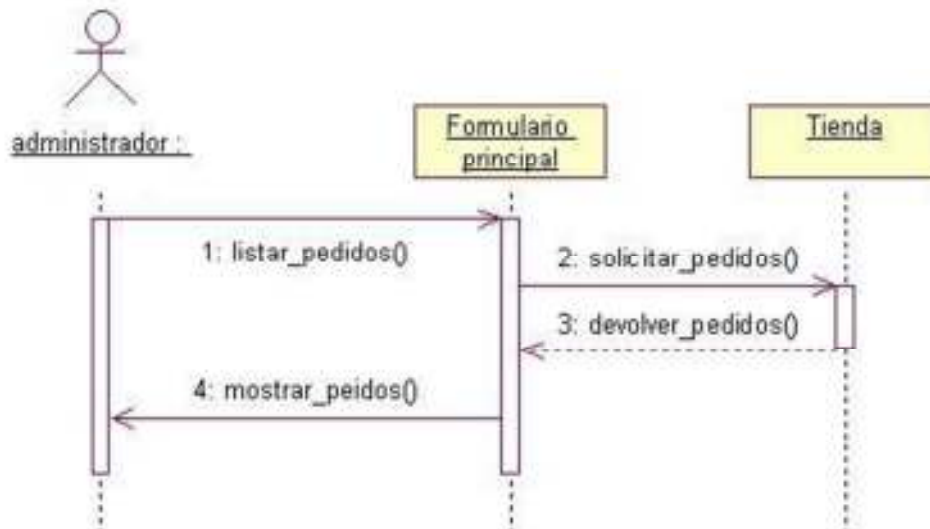
Una vez listado los usuarios seleccionaremos el usuario que se desee eliminar de la base de datos.



### Escenario: listar pedidos.

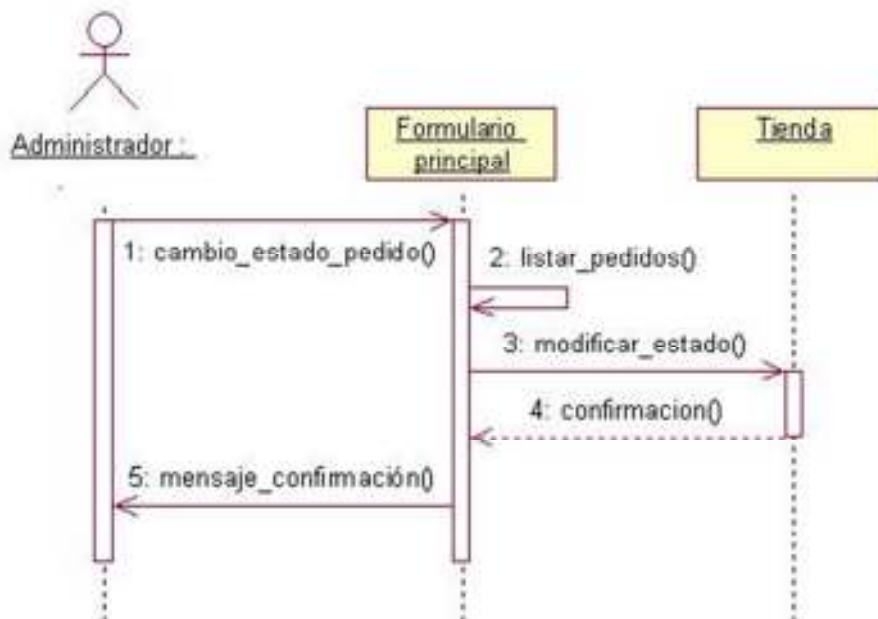
El administrador desde esta opción obtiene un listado con los pedidos efectuados.

## Diseño e implementación de una tienda virtual



### Escenario: Cambio estado pedido.

El administrador dispone de esta funcionalidad consistente en seleccionar el producto y modificar el estado en que se encuentra.



## 4. Diseño.

### 4.1. Introducción.

En esta fase realizaremos una descripción de nuestro proyecto con un mayor nivel de detalle, teniendo en cuenta las características de un entorno Web.

A partir de los modelos obtenidos en la fase anterior, análisis, plantearemos como llevar a cabo la implementación de la aplicación Web pero, sin entrar en los detalles de una tecnología concreta. Definiendo la arquitectura que define los principales componentes de nuestra aplicación en el contexto de un entorno Web.

### 4.2. Arquitectura multicapa.

Una arquitectura multicapa es un conjunto ordenado de subsistemas, cada uno de los cuales está constituido en términos de los que tiene por debajo y proporciona la base de la implementación de aquellos que están por encima de él.

Los objetos de cada capa suelen ser independientes, aunque suelen haber dependencias entre objetos de distintas capas.

Existe una relación cliente/servidor entre las capas inferiores, que son las que proporcionan los servicios, y las capas superiores, los usuarios de estos servicios.

El diseño de nuestra aplicación se basa en una arquitectura multicapa, más concretamente en una arquitectura de tres capas:

- Capa de presentación, también conocida como interfaz gráfica, se encarga de la presentación de los resultados al usuario y la recogida de información del usuario al sistema.
- Capa de negocio o lógica de la aplicación, proporciona la funcionalidad de la aplicación. Es el encargado de realizar todas las operaciones a nivel aplicación.
- Capa de datos o persistencia, este nivel es el encargado de almacenar toda la información de nuestra aplicación, además de asegurar el acceso a la información de una forma controlada y segura. Este nivel lo forman la base de datos y el software de gestión de base de datos.



Imagen1

#### 4.2.1. Nivel de presentación.

En nuestra aplicación existen tres perfiles distintos por lo tanto tres vistas, la del usuario anónimo, usuario registrado y la del administrador. El diseño de la interfaz será similar en estas tres vistas, con la diferencia de que dependiendo del usuario este disponga de más o menos opciones en su menú.

La organización de nuestra aplicación Web será la siguiente:

Parte superior, contendrá el logotipo junto al nombre de nuestra tienda.

Nos permitirá darnos de alto o bien realizar el registro de nuevo usuario, además de visualizar nuestro carrito de la compra y salir de la sesión de nuestro usuario.

El usuario anónimo, dispondrá de las opciones de búsqueda de productos del catálogo, al igual que el resto de usuarios.

El usuario registrado podrá completar las compras además de poder realizar las funciones del usuario anónimo.

Al administrador se le añaden las opciones de gestión de categorías, producto, socios registrados y compras. Que incorporan tanto las opciones de altas y bajas así como cambio de estados en las compras. Y permite realizar todo tipo de listados.

En el marco izquierdo de las principales pantallas incorporaremos el catálogo de productos informáticos, dividido en categorías.

En el centro de la pantalla aparecerá la descripción del producto cuando sea seleccionado, así como los formularios de registro. Si es el administrador el usuario identificado, entonces utilizaremos la pantalla central para toda la presentación de las acciones que se puedan llevar a cabo.

Las siguientes capturas corresponden con los distintos perfiles que hemos definido en nuestra aplicación y que a continuación pasamos a explicar.

## Diseño e implementación de una tienda virtual

Esta primera pantalla corresponde a la plantilla utilizada para la realización del parte accesible por cualquier usuario, a excepción del carrito y la zona de administración que han sido elaboradas con otras plantillas.



Plantilla1

En la zona que numerado con un 1 hemos ubicado el logotipo de nuestra empresa ficticia, Info-star, este logo aparecera en la misma zona en todas nuestras páginas independientemente del perfil de usuario que acceda a la aplicación.

En el número dos hemos posicionado nuestro menu de navegación justo debajo del logotipo de la empresa, el cual se compone de las siguientes opciones:

- **Inicio**, nos lleva hasta la pagina principal, en nuestro caso index.php.
- **Login** desde aquí insertando nuestro nombre de usuario y password se nos permite identificarnos siempre que estemos dados de alta.
- **Dar de alta**, esta opción permite al usuario anónimo rellenar un formulario para registrarse.
- **Carrito** en este apartado podremos realizar la consulta de los productos que se van añadiendo para su posterior compra.
- **Administrar** podremos acceder, siempre que seamos usuario administrador, a las opciones de gestión.
- **Logout** este botón permite eliminar la sesión actual, es decir borrará todos los productos del carro y la sesión de usuario.

En el lateral izquierdo hemos ubicado la zona donde aparecen desplegadas las distintas categorías, corresponde con la zona número 3, la categorías aparecerán en toda las páginas de usuario anónimo y registrado a excepción del carrito de la compra.

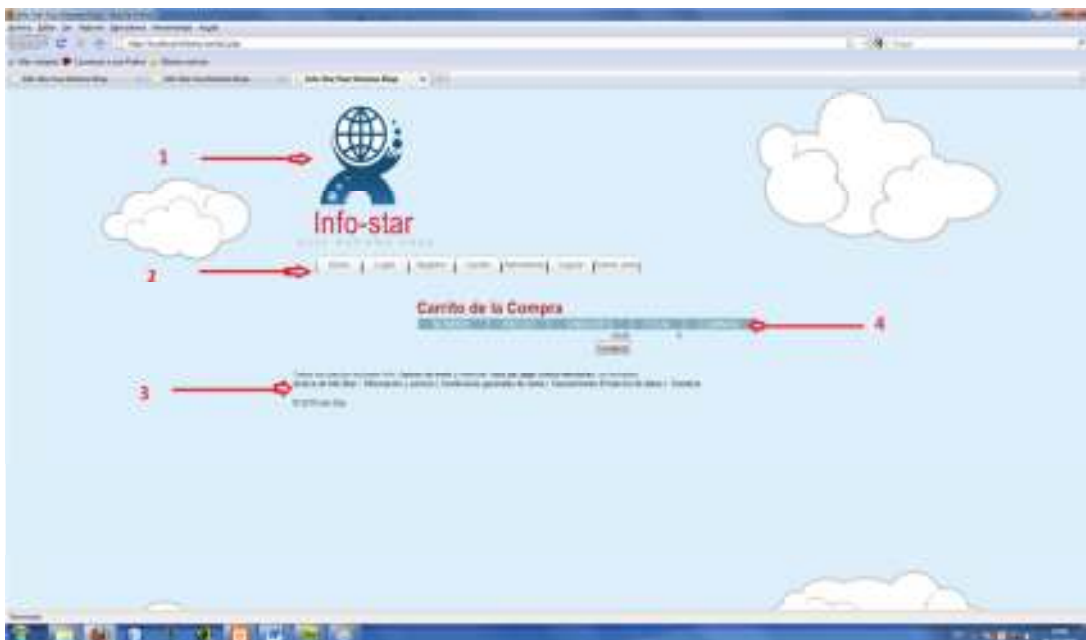
## Diseño e implementación de una tienda virtual

En el pie de página hemos ubicado todo lo referente a gastos, envíos, información de la empresa, normativa de compra, protección de datos y contacto, en la captura zona 4.

La zona superior derecha hemos añadido una pequeña aplicación que nos permite realizar una búsqueda, introduciendo una palabra o frase, nos dará como resultado todos los productos que contengan dicha clave, la hemos numerado con el 5.

Para terminar la zona número 6, corresponde con la zona central de nuestra pantalla, donde nos irán apareciendo los productos, sus características, los formularios de registro y acceso, los mensajes de errores o éxito ...

La siguiente captura corresponde con la zona de carrito de la compra, la cual se puede acceder con cualquiera de la usuarios a excepción de la última parte de confirmación de compra, no accesible para anónimos.



Plantilla2

La zona 1 y 3 no varían, disponemos de las mismas opciones anteriormente explicadas.

En la zona de navegación hemos añadido un botón que nos permitirá volver a la página anteriormente visitada.

En el centro de la pantalla aparecerá el material que vamos añadiendo al carro para efectuar una posible compra, que como ya hemos comentado anteriormente, para poder finalizarse el usuario anónimo deberá identificarse o registrarse.

La última de nuestras capturas corresponde a la zona de gestión, solo accesible por nuestro usuario administrador.





Plantilla3

Las zonas 1,2 y 3 se corresponden con las anteriormente explicadas, no han variado nada a nivel funcional, se ha modificado la accesibilidad disponible solo a usuarios administradores.

En la parte central, zona 4, nos aparecerán las opciones de gestión del administrador, una vez elegidas accederemos a los formularios que nos permitirán realizar listados, cambios, altas y bajas de los distintos productos, usuarios, categorías y pedidos.

#### 4.2.2. Nivel de lógica de la aplicación.

Este nivel estará formado por un conjunto de funciones que agruparemos en librerías, estas funciones sustituyen a los métodos que tendrían las clases correspondientes a los diagramas de clases de UML.

Su funcionalidad hará posible las operaciones descritas en apartados anteriores, conexión con la base de datos, manipulación de la base de datos, validación de usuarios y datos, registro socios, alta productos, listar usuarios, búsqueda de productos...

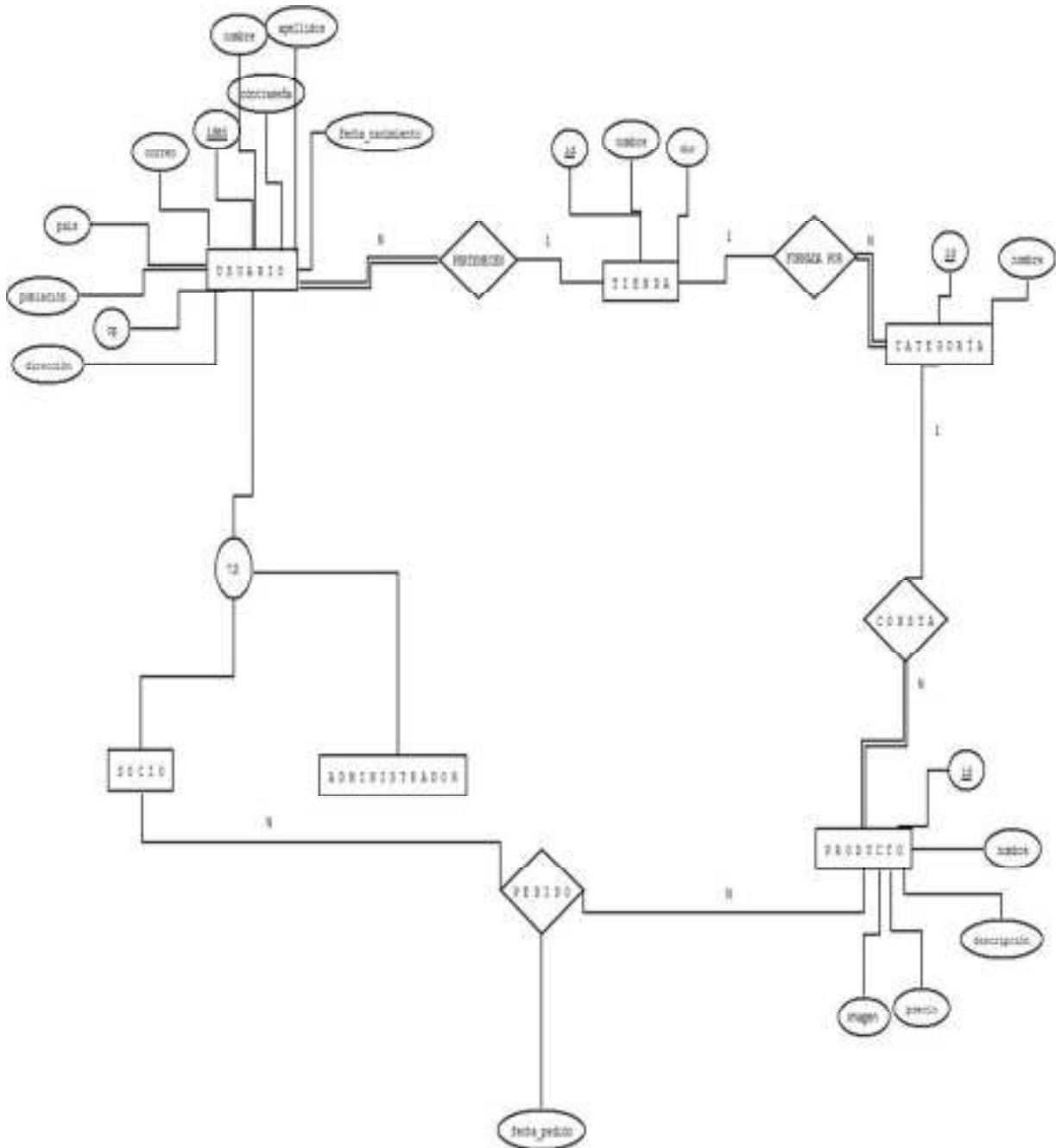
La separación por capas nos permite tener una independencia entre niveles, de esta manera el mantenimiento y las posibles ampliaciones de la aplicación se realizarán de una forma sencilla, rápida y limpia.

#### 4.2.3. Nivel de datos o persistencia.

Este nivel estará formado por una base de datos relacional, la cual, almacenará todos los datos relacionados con nuestra aplicación Web, tanto usuarios registrados, como los datos de todos los productos de nuestro catálogo, las compras...

### 4.3. Diagrama entidad relación.

A continuación tenemos el diagrama entidad relación que vamos a utilizar para la creación de nuestra base de datos.



## 4.4. Diseño lógico.

**TIENDA** (id: int(10), nombre: varchar(25), dir: varchar(50))

CP: {id}

VNN: {nombre, dir}

Restricción de existencia de TIENDA a CATEGORÍA.

**CATEGORÍA** (id: int(10), nombre: varchar(25), id\_tienda: int(10))

CP: {id}

VNN: {nombre, id\_tienda}

CAJ: {id\_tienda} → TIENDA

Restricción de existencia de CATEGORÍA a PRODUCTO.

**PRODUCTO** (id: int(10), nombre: varchar(50), descripción: varchar(300), id\_categoria: int(10), id\_subcategoria: int(10), imagen: varchar(15), precio: decimal(7))

CP: {id}

VNN: {nombre, descripción, id\_categoria, imagen, precio}

CAJ: {id\_categoria} → CATEGORÍA.

Restricción de existencia de PRODUCTO a CATEGORÍA.

**USUARIO** (iden: varchar(20), contraseña: varchar(30), id\_tienda: int(10), tipo: varchar(20), nombre: varchar(50), apellidos: varchar(100), fecha\_nacimiento: datetime, dirección: varchar(100), cp: int(10), población: varchar(30), país: varchar(50), correo: varchar(100))

CP: {iden}

VNN: {contraseña, tipo, nombre, apellidos, fecha\_nacimiento, dirección, cp, población, país, correo}

CAJ: {id\_tienda} → TIENDA

Restricción Total y Disjunta.

## Diseño e implementación de una tienda virtual

**PEDIDO** (id: int(10),idem\_usuario: varchar(50), fecha\_pedido: datetime, estado:varchar(15),total: double(10,2), prod\_nombre: varchar(50),prod\_precio: double(10,2), prod\_unidades: int(10), id\_compra: int(10))

CP: {id}

VNN: {idem\_usuario, fecha\_pedido, estado, total, prod\_nombre, prod\_precio, prod\_unidades}

CAJ:{id\_compra}→COMPRA

**COMPRA** (id: int(10), id\_pedido: int(10))

CP: {id,id\_pedido}

CAJ: {id\_pedido}→ PEDIDO

## 5. Implementación.

### 5.1. Tecnologías.

Para poder realizar toda nuestra aplicación hemos utilizado distintas tecnologías y lenguajes de programación.

Inicialmente hemos necesitado simular un servidor Web para poder utilizar nuestra aplicación y ver las respuestas que luego se darán en un servidor real, dentro de la red.

Para ello hemos optado por instalar XAMPP 1.7.3, que incorpora en un solo programa:

- Apache versión 2.2.14, es el servidor Web.
- PHP versión 5.3.1, lenguaje de programación interpretado, para webs dinámicas.
- MySQL, junto con PhpMyADMIN que nos permite la creación y gestión de nuestras bases de datos.

En nuestro diseño de interfaces hemos utilizado la programación con el lenguaje HTML y CSS, para darle una apariencia correcta a todas las páginas y que nos permita con el mínimo trabajo posible realizar modificaciones.

Con JavaScript hemos realizado las comprobaciones de nuestros formularios, para así evitar que se dejen en blanco o se rellenen con espacios en blanco, además de comprobar el correo y nombre identificativo de los usuarios que se van a registrar.

PHP ha sido el lenguaje elegido para desarrollar la lógica de la aplicación, ya que es uno de los lenguajes más utilizados en la red, el cual nos permite la realización de páginas dinámicas de una forma rápida y potente.

Para el nivel de persistencia, hemos utilizado MySQL, que conjuntamente con PhpMyADMIN hemos podido crear y administrar nuestra base de datos de una forma cómoda y visual.

#### 5.1.1. HTML.

Corresponde a las siglas de *HyperText Markup Language* (lenguaje de marcado de hipertexto), es el lenguaje predominante en la elaboración de páginas Web que nos permite tener documentos estructurados y vinculados entre si.

Es usado para describir la estructura y el contenido en forma de texto, así como para completar el texto con objetos tales como imágenes, tablas...

### 5.1.2. CSS.

Las hojas de estilo en cascada o CSS, *Cascading Style Sheets*, es un lenguaje utilizado para definir la presentación de un documento estructurado, HTML o XHTML.

Este tipo de lenguaje nos permite de una forma sencilla, definir todo tipo de formatos, colores... en definitiva la apariencia que le queremos dar a nuestra aplicación Web, inclusive podemos disponer de varias hojas de estilo, para que según que visitante o que usuario, según el navegador con el que accedemos, si queremos imprimir una página...

Los cambios se realizan en una única página, no tenemos que ir página por página definiendo un nuevo estilo, con cambiar nuestro CSS es suficiente, es una forma muy efectiva y potente.

### 5.1.3. JavaScript.

Javascript es un lenguaje de programación interpretado. Se define como orientado a objetos, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones del lenguaje de programación Java.

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

En nuestra aplicación, el JavaScript utilizado es básicamente para realizar operaciones de comprobación lanzando mensajes de advertencia y ubicando el foco en la parte del formulario que debemos rellenar de forma correcta.

También lo hemos utilizado en nuestra barra de navegación, en el botón de volver atrás.

### 5.1.4. PHP.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Es usado principalmente en la interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando bibliotecas específicas.

Nos permite la construcción de páginas Web con independencia del servidor y de la base de datos que estemos utilizando, con una compatibilidad completa en cualquier plataforma, además de integrarse de una forma completa con el código HTML.

### 5.1.5. MySQL.

MySQL, *Structured Query Language*, es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Una base de datos relacional almacena los datos en tablas separadas, lo cual hace que la velocidad y flexibilidad de trabajo sean muy grandes.

Cada tabla esta enlazada mediante relaciones, con lo cual es posible combinar datos de varias tablas cuando realizamos una consulta.

Este *software* es de código abierto, *open source*, con lo cual cualquier persona puede utilizarlo y modificarlo sin necesidad de realizar la compra del mismo.

## 5.2. Herramientas.

Las herramientas utilizadas en el desarrollo de nuestra aplicación has sido:

- XAMPP versión 1.7.3., el cual ya hemos comentado anteriormente, funcionando bajo el entorno de *Windows 7* versión *Ultimate*.

- *DreamWeaver CS4*, esta herramienta nos ha ayudado en la programación de HTML, aunque también la hemos utilizado de ayuda con la programación en PHP y pos supuesto para realizar nuestra hoja de estilos.

- Con *NetBeans* hemos programado parte del código PHP, es un software gratuito y muy completo ya que aparte del lenguaje PHP te permite la programación en multitud de lenguajes orientados a objetos.

- Para la creación del logotipo de la empresa hemos utilizado *AAA.Logo 2008*, con imágenes prediseñadas, es un software muy intuitivo que te permite la creación sencilla de logos.

- ArgoUML, nos ha permitido la creación de todos los diagramas UML y su exportación en imágenes jpg.

- Para el retoque y captura de imágenes hemos utilizado un clásico, el *paint*, con características más que suficientes para los pequeños arreglos que hemos necesitado.

## 5.3. Detalles de implementación.

El índice o página principal es accesible por cualquier tipo de usuario, desde aquí si el visitante ya estaba registrado podrá acceder a la opción de identificación, de lo contrario deberá efectuar el alta en el sistema para poder proceder a la finalización de los posibles productos que se ofrecen en nuestra aplicación.

Una vez el usuario identificado, si este es administrador, podrá acceder a las opciones de gestión de la aplicación.

Vamos a proceder a detallar algunas de las funcionalidades más significativas que hemos implementado en la aplicación de nuestra tienda virtual.

### 5.3.1. Identificación.

Desde registro.php realizaremos la comprobación de que el usuario existe y que corresponde con su contraseña, además de saber si el usuario es administrador o simplemente un usuario registrado.

## Diseño e implementación de una tienda virtual

Si alguno de los datos no son correctos nos redireccionará otra página donde se nos dará la opción de volver a introducir los datos o registrarnos.

En el caso contrario la redirección nos dará las opciones de seguir navegando, seguir comprando o en el caso de ser administrador gestionar productos.

```
<?php
```

```
// *** Validate request to login to this site.
```

```
if (!isset($_SESSION)) {
```

```
    session_start();
```

```
}
```

```
$loginFormAction = $_SERVER['PHP_SELF'];
```

```
if (isset($_GET['accesscheck'])) {
```

```
    $_SESSION['PrevUrl'] = $_GET['accesscheck'];
```

```
}
```

```
if (isset($_POST['usuario'])) {
```

```
    $loginUsername=$_POST['usuario'];
```

```
    $password=$_POST['contrasena'];
```

```
    $MM_fldUserAuthorization = "tipo";
```

```
    $MM_redirectLoginSuccess = "registro_correcto.php";
```

```
    $MM_redirectLoginFailed = "error_registro.php";
```

```
    $MM_redirecttoReferrer = false;
```

```
mysql_select_db($database_conexion_bd, $conexion_bd);
```

```
$LoginRS__query=sprintf("SELECT iden, contrasena, tipo FROM usuario WHERE iden=%s AND contrasena=%s",
```

```
GetSQLValueString($loginUsername, "text"), GetSQLValueString($password, "text"));
```

```
$LoginRS = mysql_query($LoginRS__query, $conexion_bd) or die(mysql_error());
```

```
$loginFoundUser = mysql_num_rows($LoginRS);
```

```
if ($loginFoundUser) {
```

```
    $loginStrGroup = mysql_result($LoginRS,0,'tipo');
```

```
    //declare two session variables and assign them
```

```
    $_SESSION['MM_Username'] = $loginUsername;
```

```
    $_SESSION['MM_UserGroup'] = $loginStrGroup;
```

```
if (isset($_SESSION['PrevUrl']) && false) {
```



## Diseño e implementación de una tienda virtual

```
$MM_redirectLoginSuccess = $_SESSION['PrevUrl'];  
}  
header("Location: " . $MM_redirectLoginSuccess );  
}  
else {  
    header("Location: " . $MM_redirectLoginFailed );  
}  
}  
?>
```

### 5.3.2. Acceso restringido.

El siguiente fragmento de código PHP, lo hemos incluido en la plantilla utilizada para crear todas las páginas correspondientes a la parte de gestión de nuestra aplicación.

Desde este código se comprueba que el usuario registrado sea administrador, en el caso de no estar registrado lo redirigimos a la página `acceso_restringido.php`, donde aparece un mensaje con el aviso de que no está registrado y le damos la opción con un link de poder ir a la página de identificación.

Si está registrado pero, no es administrador el mensaje le avisa que no tiene privilegios de administración.

Por último si se comprueba que el usuario está registrado y es administrador no enviará al índice de administración, en el fichero `index-admin.php`.

```
<?php  
if (!isset($_SESSION)) {  
    session_start();  
}  
$MM_authorizedUsers = "administrador";  
$MM_donotCheckaccess = "false";  
// *** Restrict Access To Page: Grant or deny access to this page  
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {  
    // For security, start by assuming the visitor is NOT authorized.  
    $isValid = False;  
    // When a visitor has logged into this site, the Session variable MM_Username set equal to their username.  
    // Therefore, we know that a user is NOT logged in if that Session variable is blank.
```

## Diseño e implementación de una tienda virtual

```
if (!empty($UserName)) {  
  
    // Besides being logged in, you may restrict access to only certain users based on an ID established when they  
    login.  
  
    // Parse the strings into arrays.  
  
    $arrUsers = Explode(",", $strUsers);  
  
    $arrGroups = Explode(",", $strGroups);  
  
    if (in_array($UserName, $arrUsers)) {  
  
        $isValid = true;  
  
    }  
  
    // Or, you may restrict access to only certain users based on their username.  
  
    if (in_array($UserGroup, $arrGroups)) {  
  
        $isValid = true;  
  
    }  
  
    if (($strUsers == "") && false) {  
  
        $isValid = true;  
  
    }  
  
    }  
  
    return $isValid;  
  
}  
  
$MM_restrictGoTo = "../Acceso_restringido.php";  
  
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers, $_SESSION['MM_Username'],  
$_SESSION['MM_UserGroup'])))) {  
  
    $MM_qsChar = "?";  
  
    $MM_referrer = $_SERVER['PHP_SELF'];  
  
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";  
  
    if (isset($_QUERY_STRING) && strlen($_QUERY_STRING) > 0)  
  
    $MM_referrer .= "?" . $_QUERY_STRING;  
  
    $MM_restrictGoTo = $MM_restrictGoTo . $MM_qsChar . "accesscheck=" . urlencode($MM_referrer);  
  
    header("Location: " . $MM_restrictGoTo);  
  
    exit;  
  
}  
  
?>
```

### 5.3.3. Logout.

Desde el fichero `logout.php` destruimos las variables de sesión que hemos creado, y la sesión, enviamos al usuario a la página inicial, `index.php`.

```
<?php
session_start();
session_unset();
session_destroy();
header("location: index.php");
?>
```

### 5.3.4. Comprobación de campos.

Este pequeño *script* nos permite comprobar que el campo del formulario `iden` no se encuentra vacío, ni contiene espacios en blanco, si es así continua con la siguiente parte del *script*, hasta llegar a la finalización del mismo.

En caso contrario lanzara un mensaje advirtiendo que no se han introducido los datos de forma correcta y dejando el foco en el campo correspondiente.

```
function valida_envia(){

//valido el producto
nombre1=document.fvalida.iden.value

if (nombre1==null || nombre1.length==0 || /^s+$/ .test(nombre1))
{
alert("Tiene que escribir el nombre del usuario")
document.fvalida.iden.focus()
return 0;
}
...
}
```

### 5.3.5. Validación del correo electrónico.

La parte de este *script* va a comprobar que la dirección de correo introducida sea correcta, es decir que contenga un *string* antes del @ después otro *string* un punto y después del punto otro *string*.

Si no es así nos lanzará el mensaje de alerta, dejando el foco en el campo del correo.

Si los datos introducidos cumplen con estos parámetros continuaremos con la siguiente parte del *script* verificará que el siguiente campo y el del correo sean iguales, si es correcto continuará con el programa.

En caso negativo lanzaremos el mensaje de alerta avisando del error y dejando el foco en el campo erróneo.

```
valor=document.getElementById("correo").value;
var b=/^[^\s]+@[^\s]+\.[^\s]+\.[^\s]+$/

if(!(b.test(valor))){

    alert("El correo no es correcto")
    document.fvalida.contrasenya.focus()
    return 0;

}

if (document.fvalida.correo.value!=document.fvalida.correo2.value){

    alert("Los correos no son iguales")
    document.fvalida.contrasenya2.focus()
    return 0;
}
```

### 5.3.6. Validación del código postal.

Una vez comprobado que los campos no están vacíos, ni con espacios en blanco, este script verificará que el número introducido no contiene decimales.

En caso que detecte alguna de estas anomalías procederemos igual que en los casos anteriores, lanzamos el mensaje de advertencia y dejamos el foco en el campo a corregir. Si todo va bien continuamos con las comprobaciones.

```
decalloved = 0;
cp1=document.fvalida.cp.value
```

## Diseño e implementación de una tienda virtual

```
if (cp1==null || cp1.length==0 || /^s+$/ .test(cp1) || isNaN(cp1)){

alert("Tiene que escribir el C.P. o no ha introducido un número válido")

document.fvalida.cp.focus()
return 0;
}

if (cp1.indexOf('.') == -1) cp1 += ".";

dectext = cp1.substring(cp1.indexOf('.')+1, cp1.length);

if (dectext.length > decallowed)
{
    alert (" Introduce un numero con sin decimales. Intentalo de nuevo.");
    document.fvalida.cp.focus()
    return 0;
}

}
```

### 5.3.7. Modificar estado pedido.

Con este código se realiza la modificación de estado en que se encuentra el pedido, nos permite seleccionar entre pendiente, preparado, agotado y servido.

Una vez seleccionado el estado nuevo, accedemos a la tabla correspondiente, pedido, donde el nuevo valor para el estado es introducido, una vez realizada la operación nos envía a la página exito.php donde se nos informa que la operación ha sido realizada de forma correcta.

```
$editFormAction = $_SERVER['PHP_SELF'];

if (isset($_SERVER['QUERY_STRING'])) {

    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);

}

if ((isset($_POST["MM_update"])) && ($_POST["MM_update"] == "form1")) {

    $updateSQL = sprintf("UPDATE pedido SET estado=%s WHERE id=%s",

        GetSQLValueString($_POST['estado'], "text"),

        GetSQLValueString($_POST['id'], "int"));

    mysql_select_db($database_conexion_bd, $conexion_bd);

    $Result1 = mysql_query($updateSQL, $conexion_bd) or die(mysql_error());

    $updateGoTo = "exito.php";
```

## Diseño e implementación de una tienda virtual

```
if (isset($_SERVER['QUERY_STRING'])) {  
    $updateGoTo .= (strpos($updateGoTo, '?') ? "&" : "?");  
    $updateGoTo .= $_SERVER['QUERY_STRING'];  
}  
header(sprintf("Location: %s", $updateGoTo));  
}  
  
$colname_listar_pedido = "-1";  
if (isset($_GET['id'])) {  
    $colname_listar_pedido = $_GET['id'];  
}  
mysql_select_db($database_conexion_bd, $conexion_bd);  
$query_listar_pedido = sprintf("SELECT * FROM pedido WHERE id = %s",  
GetSQLValueString($colname_listar_pedido, "int"));  
$listar_pedido = mysql_query($query_listar_pedido, $conexion_bd) or die(mysql_error());  
$row_listar_pedido = mysql_fetch_assoc($listar_pedido);  
$totalRows_listar_pedido = mysql_num_rows($listar_pedido);  
?>
```

## 6. Evaluación y pruebas.

### 6.1. Evaluación.

En esta fase de desarrollo de nuestra aplicación vamos a evaluar tanto el funcionamiento como la usabilidad, pasando por la compatibilidad con los distintos navegadores.

Existen una serie de pautas que se deben cumplir para que el nivel de usabilidad de nuestra aplicación sea alto:

- Debemos ponernos en lugar del usuario y tener en cuenta en que modo desearíamos encontrar la información.
- Pensar en las posibles limitaciones que el usuario a nivel de procesamiento, ancho de banda o resolución pudiera tener.
- Diseñar el grado de profundidad que daremos a nuestros contenidos y definir un árbol de navegación atractivo sin excesivos enlaces.
- Obtener un diseño visual con importante carga semántica, que transmita lo que nos proponemos y que cree un impacto visual sin que produzca confusión.

Con estas recomendaciones he tratado de realizar mi aplicación, existen infinidad de recomendaciones para crear un nivel alto de usabilidad pero, para mi parecer, estas son las más importantes.

Al seguir estas recomendaciones desde el principio del proyecto la usabilidad se ha ido incorporando de un modo interactivo, iterativo e incremental. En el resultado final se nota que hemos buscado una interface sencilla, agradable y cómoda para que el usuario final se desenvuelva con soltura.

### 6.2 Pruebas.

A medida que hemos ido creando la aplicación se ha ido realizando pruebas para comprobar la navegabilidad, para ello hemos utilizado un programa llamado Xenu, que nos permite comprobar si existen enlaces rotos.

Otras pruebas a realizar son la validación de los estándares desde la página del W3C.

#### 6.2.1. Pruebas de validación.

Las pruebas de validación de formato se pueden realizar directamente desde la página de la organización W3C, nos permite validar el código XHTML y las hojas de estilo en cascada CSS.

## Diseño e implementación de una tienda virtual

Para realizar la validación de nuestro código lo podremos realizar desde el servidor en que resida nuestra web, copiando directamente el código en la página o mediante un *upload* de la misma.

La mayoría de errores que nos aparecen en las páginas de nuestra aplicación son debidos a la mezcla de código HTML con PHP, aunque son muy pocos. Lo mismo pasa con los *warnings*.

La hoja de estilos nos da un par de errores, pero tras modificar las líneas correspondientes a los avisos, pasamos sin ningún problema la validación.



Validación css

### 6.2.2. Comprobación de enlaces.

Para realizar las pruebas de enlaces rotos hemos utilizado un programa libre llamado Xenu, el cual revisa todos los enlaces de la página seleccionada y muestra una lista indicando si existe algún problema con estos.



Nombre	Tamaño	Fecha	Extensión
index.php	46 KB	2011-07-07 10:00:00	.php
css	1 KB	2011-07-07 10:00:00	css
js	1 KB	2011-07-07 10:00:00	js
img	1 KB	2011-07-07 10:00:00	img
librerias	1 KB	2011-07-07 10:00:00	librerias
... (many more files)	...	...	...

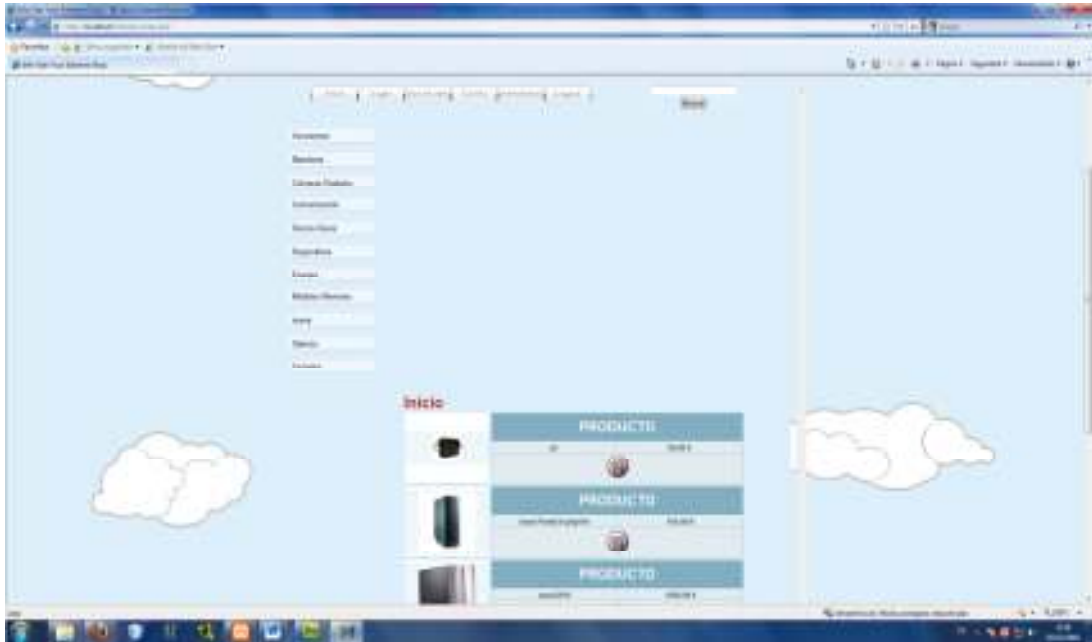
Comprobación enlaces

### 6.2.3. Comprobación con otros navegadores.

Es conveniente realizar pruebas con otros navegadores para saber si nuestra aplicación se visualizará de la forma que realmente queremos.

Hemos realizado las pruebas en los siguientes navegadores:

- Firefox versión 3.6.12, con el cual no hemos tenido ningún problema de visualización.
- Google Chrome, hemos descargado e instalado la última versión, 7.0.517.44, nuestra aplicación se visualiza de forma correcta.
- Safari 5.0.3, la última versión del mercado, nos visualiza nuestra aplicación a la perfección, al igual que los anteriores navegadores.
- Internet Explorer versión 8, nos descuadra las fichas de los productos, tanto en index.php, como en alguna de las categorías, categorias.php, era de esperar, no nos ha sorprendido.



I.Explorer8

#### 6.2.4. Otras pruebas.

Lógicamente a la vez que íbamos implementando nuestra aplicación, realizábamos pruebas unitarias y pruebas globales para verificar que el código hacía lo que creíamos y queríamos que tenía que realizar.

Ahora que la aplicación está finalizada, vamos a capturar algunos ejemplos de los casos de uso para que a la vez que plasmamos esas pruebas que habíamos hecho con anterioridad, nos sirva como pequeño manual de uso de nuestra aplicación.

Vamos a comenzar con la simulación de una compra, añadiendo productos a nuestro carrito, lo haremos inicialmente como un usuario anónimo el cual todavía no ha realizado el proceso de registro, de esta forma veremos la reacción de nuestra aplicación ante un usuario no registrado y como cuando se registra e identifica la aplicación actúa de manera distinta.

#### **Proceso de compra.**

Una vez accedemos a nuestra página principal, podemos ir visualizando productos en las distintas categorías. Nos aparecerán los productos clasificados por categorías, con su imagen, nombre, precio y un botón con un carro para añadirlos a nuestra compra.

## Diseño e implementación de una tienda virtual



Compra 1

Los detalles del producto, los podemos ver simplemente pulsando en la imagen del material que estemos interesados, entonces se abre otra página donde aparece el producto seleccionado, donde aparecerá la imagen, el nombre del producto, su descripción y el carrito para que se nos añada a nuestra compra.



Compra 2

## Diseño e implementación de una tienda virtual

Una vez pulsado el botón de añadir, el producto pasa a nuestro carrito, se añade una unidad del producto seleccionado, junto con su descripción y precio.

Como podemos ver en la siguiente captura, se introduce la nueva unidad del material y se recalcula el total de la compra.



### Compra 3

Dentro de la columna unidades tenemos la opción de modificar el número de unidades, una vez introducida la nueva cantidad solo le tenemos que dar al pequeño icono que tenemos al lado, automáticamente se actualizarán las nuevas cantidades y se recalculara el precio del total de productos iguales y del total de la compra.

Hemos añadido una columna llamada eliminar, que como indica su nombre borra de nuestro carrito los productos correspondientes a la línea donde hemos pulsado su icono.

Una vez realizada esta operación se nos recalcula el importe total de la compra.

Cuando hemos terminado de añadir los productos que necesitábamos, podemos pasar al proceso de realización de la compra, entonces dentro de carrito pulsamos el botón de comprar.

Nos llevará al cuadro de resumen de productos añadidos a nuestro carro.

Como muestra la imagen siguiente.

## Diseño e implementación de una tienda virtual



Compra 4

Es ahora cuando se nos muestra un resumen con los productos que hemos seleccionado, con su precio, unidades y total tanto unitario como el del conjunto de productos.

Si alguno de los productos que aparecen lo hemos añadido por error o bien hemos decidido que no lo queremos, simplemente pulsamos volver atrás, para así poder modificar la compra.

Cuando hemos repasado los productos de nuestro carrito pulsamos el botón que nos finaliza la compra.



Compra 5

## Diseño e implementación de una tienda virtual

Nos aparece este mensaje, debemos identificarnos si estamos registrados o en caso contrario tenemos que darnos de alta.

Como no estamos registrados vamos a proceder a darnos de alta rellenando el siguiente formulario.



### Compra 6

Si alguno de los campos, lo dejamos en blanco, o solo introducimos espacios o bien cuando nos piden repetir alguno de los campos no lo hacemos correctamente, se lanzará un mensaje de advertencia para que se solucione el problema.

Después de rellenar correctamente el formulario deberemos de enviarlo.

Somos llevados a una pantalla donde nos aparece un resumen de los datos que se van a proceder a insertar en la base de datos.

Toda la información del formulario es enviada a esta nueva página para que el usuario se cerciore que todos los productos introducidos son los que efectivamente estaba dispuesto a comprar.

Si no son correctos deberemos volver atrás para rectificarlos.

## Diseño e implementación de una tienda virtual



Compra 7

Si todos los datos son correctos procederemos a insertar el registro, antes de introducir los datos en la tabla el sistema verificará que no existan ningún usuario con el mismo nombre que el nuestro.

Si ya tenemos registrado un usuario con nuestro mismo nombre aparecerá esta pantalla, informándonos de la duplicidad del usuario.

Con un vínculo a la página donde se encuentra el formulario.



Compra 8



## Diseño e implementación de una tienda virtual

Volvemos a introducir los datos del formulario con otro identificador distinto. Si el sistema verifica que ese usuario no existe entonces procederá a insertar la tupla en la base de datos, avisándonos de la correcta inserción del nuevo usuario.



Compra 9

Ahora procederemos a identificarnos para así finalizar la compra.



Compra 10



## Diseño e implementación de una tienda virtual

Si alguno de los datos que hemos introducido son incorrectos, nos mostrará un mensaje para que lo volvamos a intentar o en caso de que no estemos dados de alta que lo hagamos.



Compra 11

Volvemos a introducir nuestro usuario y contraseña. Ahora de forma correcta para proceder a la finalización de nuestra compra.



Compra 12

## Diseño e implementación de una tienda virtual

Ahora tenemos distintas opciones, podemos continuar navegando, finalizar nuestra compra o si somos administradores ir al apartado de gestión.

Vamos a terminar nuestra compra.

Nos envía de nuevo al carrito de la compra, confirmamos la finalización de la compra y después del resumen concluiremos nuestro pedido, en esta última página nos aparecerá un número de cuenta para que efectuemos el ingreso del importe total del pedido.

Se generará un pedido con los datos de los productos comprados y una compra, que se almacenarán en sus tablas dentro de nuestra base de datos.

Cuando este realizada la transferencia Info-Star, nos preparará el pedido y nos lo enviará a nuestro domicilio.

Después de esto podemos seguir navegando por la Web.



Compra 13

### **Introducir nuevo producto.**

En este apartado vamos a realizar la traza que nos permitirá dar de alta un nuevo producto.

Lo primero que hacemos es ir al apartado de administración, como no estamos identificados nos aparecerá la siguiente pantalla.

## Diseño e implementación de una tienda virtual



### Nuevo producto 1

Nos identificamos con nuestro usuario administrador, omitiremos estas capturas ya que son iguales que las de identificación de un usuario registrado sin privilegios.

Una vez ya identificados podemos acceder a la página central de administración. Donde nos aparecerán las distintas acciones para la gestión de nuestra aplicación.



### Nuevo producto 2

Seleccionaremos nuevo producto, nos lanzará la siguiente pantalla, donde nos aparecerá un formulario para introducir los datos del nuevo producto, nombre, descripción, precio y categorías.

## Diseño e implementación de una tienda virtual

En cada campo se realizará la pertinente comprobación para asegurarnos que no están vacíos, ni llenos de espacios y en el caso del precio se verificará que sea un número, que sea mayor de 0 y que si tiene decimales que sean como máximo dos.

En el caso de la categoría simplemente seleccionaremos la más adecuada dentro del listado del que se dispone.



Nuevo producto 3

Cuando se introduce en algún campo un dato erróneo el sistema lo detectará y lanzara una advertencia para que se modifique.



Nuevo producto 4

## Diseño e implementación de una tienda virtual

Si ya se ha rellenado correctamente el formulario este será enviado. Nos aparecerá un resumen del nuevo producto, si estamos conformes entonces pulsaremos la opción de insertar la tupla en su correspondiente tabla.



Nuevo producto 5

Una vez insertado el nuevo producto, nos parecerá un mensaje del éxito de la operación. Donde podremos seguir realizando tareas de administración.



Nuevo producto 6

## 7. Conclusión.

En mi caso la primera toma de contacto con el lenguaje PHP ha sido durante el desarrollo de nuestro sistema Web.

La verdad es que inicialmente tenía algún temor sobre si la elección de este lenguaje había sido lo apropiado, más que nada por mi desconocimiento, pero poco a poco me he dado cuenta que es un lenguaje que se integra perfectamente con el código HTML, su potencia nos hace mucho más fácil realizar procesos que de otra forma serían muy rudimentarios y costosos.

PHP es una herramienta que en un corto espacio de tiempo me ha permitido un manejo con soltura, hasta yo mismo me he sorprendido de los avances que he logrado.

Ayudado por mi director de proyecto, libros, manuales, foros y video tutoriales los problemas y dudas que me iban surgiendo durante el desarrollo de la aplicación los iba resolviendo.

La forma de trabajar con PHP que he aprendido seguro que no es la más purista, pero lo realmente importante son los conocimientos adquiridos.

La mayor satisfacción de este trabajo ha sido ir superando los objetivos que se me han ido planteando. Los problemas iban surgiendo a medida que avanzaba.

Primero fueron problemas técnicos con el funcionamiento de XAMPP, una vez superados empezamos la lucha con las bases de datos y MyPHPADMIN. Después vino el enfrentamiento con el carrito de la compra y la finalización de la compra y por último todo lo relacionado con la administración, listados, altas, inserciones.

Después de cada lucha salía un poco más reforzado, ya que miraba atrás y hasta casi me reía de los problemas que había tenido, parecía que habían sido minimeces.

Aparte de haber aprendido el lenguaje PHP, he practicado y reforzado los conocimientos del lenguaje SQL, he puesto de nuevo en práctica lo aprendido sobre HTML, XHTML y CSS.

También he profundizado en el manejo de herramientas, como *Dreamweaver* y he hecho mis "pinitos" con la plataforma *Netbeans*, que además de poder trabajar con PHP permite la implementación en un gran repertorio de lenguajes de programación.

La aplicación desarrollada en este proyecto final de carrera, cumple con los requisitos acordados, aunque personalmente me han ido viniendo a la cabeza mejoras que se podrían realizar para facilitar su manejo y que las paso a detallar:

- Cuando se formalice un pedido, enviar un correo electrónico al usuario que ha efectuado la compra, detallando los productos, importes, plazo de entrega y factura de compra así como un plazo aproximado de entrega.
- Una opción accesible a los usuarios registrados, que permita consultar el estado de su pedido introduciendo el número de factura.

## Diseño e implementación de una tienda virtual

- Otra opción donde nos pida el número de pedido y aparezca la factura para su impresión.
- Una zona donde el usuario que ha efectuado alguna compra pueda dar su opinión sobre cómo se ha recibido el material, cuanto ha tardado o cualquier tipo de incidencia.
- Desarrollar el pago con PayPal o tarjeta de crédito.
- En el formulario de alta, poner una dirección de envío.

En resumen la experiencia ha sido muy grata, el aprendizaje muy completo y satisfactorio mejorando las expectativas que tenía. Pienso que todo este esfuerzo es una inversión de cara a trabajos futuros y que estos años en la universidad me han ayudado en cuanto a la adquisición de conocimientos y lo más importante, ya que estos conocimientos no serán suficientes de cara a mi futuro laboral, es adoptar la capacidad de aprendizaje y la pérdida del miedo a aprender de forma autónoma.

Por ello quiero agradecer a todo el personal docente, las capacidades que he podido adquirir en estos tres increíbles y fructíferos años.

## 8. Bibliografía.

D'andrea, Edgar. Cómo crear una tienda virtual con php 6 y mysql. Barcelona : Inforbook's , 2010. ISBN 8496897702.

D'andrea, Edgar. Cómo crear una tienda virtual con php 5 y mysql. Barcelona : Inforbook's , 2005. ISBN 8496097471.

Boronczyk , Timothy. Desarrollo web con php6, apache y mysql. Barcelona : Anaya multimedia Wrox, 2009. ISBN 9788441526228.

Choi, Wanky. Beginning php4. Birmingan: worx press, 2000. ISBN 1861003730.

Gilmore, W. Jason. Beginning PHP 5 and MySQL from novice to profesional. New York : Apress, cop. 2004. ISBN 1893115518.

Paterson, Douglas. Construyendo sitios web con PHP-Nuke. Olton, Birmingham, UK : Packt, 2005. ISBN 9781904811428.

Ratschiller, Tobias. Creación de aplicaciones Web con PHP 4. Madrid : Prentice Hall, D.L. 2000. ISBN 8420531081.

Pavón Huertas, Jacobo. Creación de un portal con PHP y MySQL. Madrid : Ra-ma, D.L. 2007. ISBN 9788478977543.

Kent, Allan. Desarrollo web con PHP y Dreamweaver MX 2004. Madrid : Anaya Multimedia, D.L. 2005. ISBN 8441518025.

Welling, Luke. Desarrollo Web con PHP y MySQL. Madrid : Anaya Multimedia, 2005. ISBN 8441518181.

Powers, David. Desarrollo web dinámico con Dreamweaver 8 y PHP. Madrid : Anaya Multimedia, 2006. ISBN 844152047X.

Duckett, Jon. Programación Web con HTML, XHTML y CSS. Madrid : Anaya Multimedia, D.L. 2008. ISBN 9788441524934.

Orós, Juan Carlos. Diseño de páginas Web con XHTML, JavaScript y CSS. Madrid : Ra-ma, D.L. 2008. ISBN 8478978569.

Duckett, Jon. Accessible XHTML and CSS web sites problem: *problem - design -solution*. Indianapolis : Wiley, 2005. ISBN 0764583069.

Meyer, Eric A. CSS : *Pocket reference*. New York : O'Reilly, 2007. ISBN 0596515057.

Beauchemin, Bob. A developer's guide to SQL server 2005. Upper Saddle River, NJ : Addison-Wesley, cop. 2006. ISBN 0321382188.



## Diseño e implementación de una tienda virtual

World Wide Web Consortium(W3C). Documentos, guías y estándares.[Consulta: 02 de noviembre de 2010]. Disponible en: <http://www.w3c.es>

Wikipedia la enciclopedia libre. Definiciones. [Consulta: 05 de noviembre de 2010].Disponible en: <http://es.wikipedia.org>

REAL ACADEMIA ESPAÑOLA. Ortografía .[Consultas: 08 de noviembre de 2010]. Disponible en: <http://rae.es>

Programación por capas. Imagen1. [Consultado: 03 de diciembre de 20010]. Disponible en : <http://mtycoders.com/programacion-por-capas/>

Maestros del Web. Registro de usuarios utilizando a PHP. [Consultado: 15 de noviembre de 20010]. Disponible en: <http://www.maestrosdelweb.com/editorial/phppreusr/>

El web master. Procesar un formulario. [Consultado: 16 de noviembre de 20010]. Disponible en: <http://www.elwebmaster.com/editorial/taller-de-php-procesar-un-formulario-recibiendo-los-datos>

Sector web. Estilos CSS. [Consultado: 16 de noviembre de 20010]. Disponible en: <http://www.sectorweb.net/index.php>

Ricardo Tayar. Utilizar CSS. [Consultado: 16 de noviembre de 20010]. Disponible en: <http://www.ricardotayar.com/2007/11/10/utilizar-css-con-tablas-para-hacer-cosicas-chulas-asi-se-hace-otro-gran-secreto-de-css-desvelado/>

Desarrolloweb. Introducción a base de datos. [Consultado: 10 de noviembre de 20010]. Disponible en: <http://www.desarrolloweb.com/articulos/video-base-datos-php.html>

Libros Web. Introducción a JavaScript. [Consultado: 20 de noviembre de 20010]. Disponible en: <http://www.librosweb.es/javascript/capitulo7/validacion.html>

W3schools. Curso de PHP. [Consultado: 05 de noviembre de 20010]. Disponible en: <http://www.w3schools.com/php/default.asp>

Tutoriales scourdesign. CSS diseño menús. [Consultado: 25 de noviembre de 20010]. Disponible en: <http://www.scourdesign.com/articulos/tutoriales/css/css14.php>

Ignside. Variables en PHP. [Consultado: 10 de noviembre de 20010].Disponible en: <http://www.ignside.net/man/php/variables.php>

Buendía García, Félix. Guía para la realización y supervisión de proyectos de final de carrera (PFC) en el ámbito de la web. Valencia: Editorial UPV. ISBN 9788483633250.

López Moratilla, Javier. Catálogo de tienda virtual. PFC Valencia : Universidad Politécnica de Valencia, 2007.

Broch De Gracia, David. Comercio electrónico.PFC Valencia : Universidad Politécnica de Valencia, 2003.

Apuntes ISG. Esquemas y división proyecto. [Consulta: 02 de noviembre de 2010]. Disponible en: <http://poliformate.upv.es>

<<La imaginación es más importante que el conocimiento. El conocimiento es limitado, mientras que la imaginación no>>

Albert Einstein.