

# DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE

Memoria de proyecto final de carrera

**Autora:** Jill Tiffany Acosta Montañez

**Director:** Juan Vicente Capella Hernández



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE

**Agradecimientos:**

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que han colaborado de manera directa o indirecta en este proyecto.

Quiero dar las gracias a mi familia, por confiar en mí y apoyarme en todo momento. Ellos me han inculcado los valores y principios que han servido de cimiento, para que hoy se vea realizado uno de mis objetivos.

**Tabla de contenido**

<b>1. Introducción</b> .....	10
1.2 Justificación del proyecto.....	10
<b>2. Especificación de requisitos</b> .....	12
2.1. Introducción.....	12
2.1.1. Propósito .....	12
2.1.2. Ámbito.....	12
2.1.3. Definiciones, acrónimos y abreviaturas .....	12
2.1.4. Referencias .....	14
2.1.5. Visión general del documento .....	14
2.2. Descripción general.....	15
2.2.1. Perspectiva el producto .....	15
2.2.2 Funciones del producto .....	15
2.2.3. Características del usuario .....	16
2.2.4. Restricciones .....	16
2.2.5. Suposiciones y dependencias.....	17
2.3. Requisitos de especificación.....	17
2.3.1. Interfaces extras .....	17
2.3.1.1. Interfaces de usuarios .....	17
2.3.1.2. Interfaces Hardware .....	17
2.3.1.3. Interfaces Software .....	17
2.3.1.4. Interfaces de comunicaciones .....	18
2.3.2. Funciones .....	18

2.3.2.1. Cliente .....	18
2.3.2.2. Administrador .....	19
2.3.3 Requerimientos de Rendimiento .....	20
2.3.4. Restricciones de diseño .....	20
2.3.5. Atributos del sistema .....	20
2.3.5.1. integridad .....	20
2.3.5.2. Facilidades de mantenimiento .....	20
2.3.5.3. Seguridad .....	21
2.3.6. Otros requerimientos .....	21
2.3.6.1. Base de datos .....	21
<b>3. Análisis.....</b>	<b>22</b>
3.1. Descripción del proyecto a desarrollar .....	22
3.2. Seguridad del sistema .....	22
3.3. Diagrama de casos de uso .....	22
3.3.1. Casos de uso del usuario .....	23
3.3.2. Casos de uso del administrador .....	24
3.4. Diagrama de clases .....	24
3.4. Escenario de la aplicación .....	26
<b>4. Diseño.....</b>	<b>28</b>
4.1. Introducción.....	28
4.2. Arquitectura del sistema.....	28
4.2.1. Capa de presentación.....	28
4.2.1. Capa de negocio .....	30
4.2.1. Capa de persistencia .....	30
<b>5. Implementación.....</b>	<b>32</b>
5.1. Tecnologías .....	30
5.1.1. MySql.....	32

## DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE

5.1.2. PHP( PHP Hypertext Pre-processor ) .....	33
5.1.3. HTML (HyperText Markup Language) .....	34
5.1.4. CSS (Cascading Style Sheets) .....	34
5.1.5. Paypal .....	35
5.1.6. JavaScript .....	36
5.2. Herramientas .....	36
5.2.1 AppServ .....	36
5.2.2. Apache.....	37
5.2.3 PhpMyAdmin .....	37
5.2.4 ArgoUML.....	38
5.3. Detalles de la implementación .....	39
5.3.1. Tienda para el administrador .....	39
5.3.1.1. Catalogo .....	39
5.3.1.2. Clientes.....	40
5.3.1.3. Control stock .....	40
5.3.1.4. Ventas .....	41
5.3.1.5. Configuración .....	41
5.3.2. Tienda para el cliente .....	42
5.3.2.1. Modo de acceso .....	42
5.3.2.2. Catalogo .....	43
5.3.2.3. Carrito de la compra .....	43
5.3.2.4. Forma de pago .....	44
5.3.2.5. Compra del cliente .....	46
5.3.2.6. Email compra .....	48
<b>6. Pruebas .....</b>	<b>49</b>
6.1. Evaluación .....	48
6.1.1. Pruebas de validación unitarias .....	50

6.1.2. Pruebas de compatibilidad en navegadores .....	62
<b>7. Conclusiones .....</b>	<b>64</b>
7.1. Trabajo realizado .....	64
7.2. Valoración personal .....	65
<b>8. Referencias .....</b>	<b>66</b>
<b>9. Anexo: Manual de usuario .....</b>	<b>67</b>
<b>10. Anexo: Crear una cuenta de pruebas en Paypal .....</b>	<b>91</b>

**Tabla de imágenes**

Figura 1. Caso de estudio cliente .....	23
Figura 2. Caso de estudio administrador .....	24
Figura 3. Diagrama de clases .....	25
Figura 4. Arquitectura tres capas .....	28
Figura 5. Interfaz usuario .....	29
Figura 6. Instalador Mysql Server .....	32
Figura 7. Petición de una página en PHP .....	33
Figura 8. Ejemplo HTML .....	34
Figura 9. Ejemplo hojas de estilo en cascada .....	34
Figura 10. Funcionamiento de Paypal .....	35
Figura 11. Instalador de Appserv .....	36
Figura 12. Apariencia del phpMyAdmin .....	37
Figura 13. Interfaz de usuario de ArgoUML .....	38
Figura 14. Productos categorías.....	50
Figura 15. Características producto .....	51
Figura 16. Alta usuario.....	52
Figura 17. Acceso usuario.....	52
Figura 18. Bienvenida usuario .....	53
Figura 19. Selección carrito .....	54
Figura 20. Selección medio de pago.....	54
Figura 21. Acceso paypal .....	55
Figura 22. Pago paypal .....	55
Figura 23. Volver a tienda.....	56
Figura 24. Compra por transferencia.....	56
Figura 25. Email compra .....	57



<b>Figura 26. Acceso admin .....</b>	<b>58</b>
<b>Figura 27. Página principal administración.....</b>	<b>58</b>
<b>Figura 28. Listado de categorías .....</b>	<b>59</b>
<b>Figura 29. Ejemplo de añadir producto.....</b>	<b>59</b>
<b>Figura 30. Ejemplo de editar proveedor .....</b>	<b>60</b>
<b>Figura 31. Listado de pedidos de compra .....</b>	<b>60</b>
<b>Figura 32 Listado de pedidos de compra.....</b>	<b>61</b>
<b>Figura 33. Cuentas usuario.....</b>	<b>61</b>
<b>Figura 34. Ejemplo de la tienda con la visualización en navegador de Explorer .....</b>	<b>62</b>
<b>Figura 35. Ejemplo de la tienda con la visualización en navegador de Chrome .....</b>	<b>63</b>
<b>Figura 36. Ejemplo de la tienda con la visualización en navegador de FireFox .....</b>	<b>63</b>

## 1. INTRODUCCIÓN

La razón principal por la que he decidido realizar mi trabajo fin de carrera basado en las tecnologías PHP, ha venido determinada por el deseo de aprender y profundizar en el conocimiento del desarrollo de aplicaciones en una plataforma moderna en constante actualización y que considero que tiene un gran futuro. Además considero que en concreto este trabajo permite repasar los conceptos fundamentales adquiridos durante la carrera aplicándolos a una solución que aunque teórica tiene muchos componentes que se aplican en el campo profesional de la programación.

### 1.2 JUSTIFICACIÓN DEL PROYECTO

La desaparición de las distancias en las comunicaciones, desde la popularización de internet, justifica la aparición de sitios web no ya sólo para la obtención de información sino que también para la realización de transacciones de compra-venta o lo que técnicamente se denomina e-commerce.

Hay que pensar en las ventajas que internet implica para la empresa que le permite ofrecer sus productos de forma inmediata en cualquier parte del mundo, con unos costes mínimos ya que no necesita tener un local físico y sin unos horarios de tienda ya que cualquiera se puede visitar la tienda las 24 horas día, lo que hace imprescindible para una tienda moderna disponer de este tipo de herramientas para realizar su actividad. Pero no sólo la justificación hay que buscarla en las ventajas que supone para la empresa; los potenciales clientes también pueden optar, por una gran variedad de ofertas repartidas a lo largo y ancho de toda la red pudiendo elegir la opción que consideren más interesante.

Desde esta perspectiva, surge la idea de diseñar e implementar el proyecto de una Tienda Virtual, como una tienda que ofrece diversos productos a los usuarios a través de internet en cualquier parte del país. Se va a utilizar el navegador lo cual supone que el período de aprendizaje es mínimo tanto para los empleados de la empresa como para los usuarios que se conecten a través de la red.

## 2. ESPECIFICACIÓN DE REQUISITOS

### 2.1. INTRODUCCIÓN

Esta Especificación de Requisitos Software para el diseño de una plataforma de comercio electrónico ha sido elaborada tomando en cuenta las características del sistema utilizado en la actualidad y la posibilidad de mejorarlo, de acuerdo a la experiencia de sus usuarios y los beneficios obtenidos.

---

#### 2.1.1. PROPÓSITO

El objetivo de esta especificación es definir de manera clara y exacta las funcionalidades y restricciones que presentara la aplicación que se pretende construir.

Esta especificación de requisitos va a ser destinada tanto a usuarios, como a equipo de desarrollo.

---

#### 2.1.2. ÁMBITO DEL SISTEMA

El producto que vamos a desarrollar es una plataforma que permita adquirir productos o servicios a través de la web (tienda virtual). La aplicación permitirá la gestión y seguimiento de pedidos, catálogos, etc.

---

#### 2.1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

- *Apache, Servidor.* Apache es un software servidor HTTP de código abierto para plataformas Unix, GNU/Linux, Windows, Macintosh, y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, que se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

- *IEEE (The Institute of Electrical and Electronics Engineers)*. Es una asociación de profesionales con sede en EEUU que fue fundada en 1884, que actualmente cuenta con miembros de más de 140 países. Investiga en campos como el aeroespacial, computacional, comunicaciones, etc. Y es muy conocida por ser un gran promotor de estándares.
- *MySQL*. Es un sistema de gestión de bases de datos multiusuario, multiplataforma y de código abierto que pertenece a la compañía sueca MySQL AB.
- *Página Web*. Fuente de información compatible con la WWW y que puede ser accedida a través de un navegador en Internet.
- *Password*. Palabra de paso o contraseña, se utiliza junto con el login para identificarse en el sistema.
- *PHP (PHP Hypertext Pre-processor)*. Es un lenguaje de programación interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Este lenguaje es usado normalmente para la creación de páginas Web dinámicas.
- *CSS: Las hojas de estilo en cascada (Cascading Style Sheets, CSS)* son un lenguaje formal de ordenador usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión XHTML). La W3C (World Wide Web Consortium) es la encargada de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.
- *HTML: Acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto)*, es un lenguaje informático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

- *HTTP*: Es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan Hyper Text Transfer Protocol, es decir, protocolo de transferencia hipertexto.
- *Sitio Web*. Conjunto de páginas Web que están relacionadas entre sí, por lo general porque se accede desde un mismo. Un sitio Web puede estar constituido de una o más páginas Web.
- *Usuario*. así es como se designa a cualquier persona que accede al sistema, un usuario puede desempeñar varios roles, Administrador, Empleado y Usuario Registrado.
- *Administrador*. rol asociado a la persona o personas que están capacitadas para controlar la aplicación. Los usuarios con este perfil pueden decidir sobre los contenidos y funcionamiento de la aplicación, por ello deben conocer el uso de la aplicación con detalle.

---

### 2.1.4. REFERENCIAS

IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE std. 830, 1998.

---

### 2.1.5. VISIÓN GENERAL DEL DOCUMENTO

Este documento está conformado de tres secciones que son la Introducción, la Descripción Global 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, en la tercera sección se definen los pormenores de los

requisitos que el usuario ha externado que el sistema actual cumple y por lo tanto el nuevo sistema debe satisfacer.

## 2.2. DESCRIPCIÓN GENERAL

En los siguientes puntos se tratarán los factores que afectaran al desarrollo de la aplicación así como sus requisitos.

### 2.2.1. PERSPECTIVA DEL PRODUCTO

El lenguaje utilizado para implementar la tienda virtual es PHP que nos permitirá la creación de una página Web dinámica. Por otra parte, el sistema de gestión de bases de datos que contendrá toda la información necesaria para ser mostrada en la página Web es MySQL.

Este producto (sistema de información) deberá funcionar en cualquier computador PC que tenga instalado un navegador web.

Se pretende que el producto utilice tecnologías con las que va a ser indispensable contar con un alojamiento de servicios que ofrezca posibilidad de utilizar PHP, bases de datos MySQL y que ejecute un servidor Apache.

### 2.2.2. FUNCIONES DEL PRODUCTO

La aplicación de comercio electrónico debe satisfacer las siguientes funcionalidades:

#### Usuario

- Identificarse como cliente.
- Añadir producto a la cesta de la compra.
- Ver cesta de la compra.
- Ver detalle del producto.

#### Administrador

- Añadir/ editar/ eliminar administrador.

- Añadir/ editar/ eliminar categoría.
- Añadir/ editar/ eliminar proveedor.
- Añadir/ editar/ eliminar producto.
- Añadir/ editar/ eliminar pedido proveedor

### Usuario No Registrado

- Navegar por catálogo.
- Crear cuenta cliente.
- Opciones de visualización de productos.

---

### 2.2.3. CARACTERÍSTICAS DE LOS USUARIOS

Nuestra aplicación tendrá tres tipos de usuarios:

- *Visitantes*: son aquellos que entran a nuestro sitio web, navegan por donde los dejemos, pero no pasan a autenticarse. Mientras se sigue siendo visitante no se podrá comprar ni recibir publicidad.
- *Clientes*: son aquellos usuarios que se autentican para poder acceder a todas las funcionalidades de la tienda. Los clientes no tienen privilegios para realizar tareas de administración.
- *Administradores*: son las personas autorizadas que realizan la administración de la aplicación. Por ejemplo, en la tienda virtual son los que actualizan el catálogo, los precios, las promociones, etc.

Tanto los clientes como los administradores estarán registrados en la base de datos de cuentas, con identificador y password.

---

### 2.2.4. RESTRICCIONES

Toda aquella persona que quiera acceder a los servicios ofrecidos por la aplicación deberá disponer del hardware necesario para conectarse a Internet a través de una conexión estándar.



Los usuarios potenciales de la aplicación la percibirán como una página Web por lo que los requisitos mínimos necesarios para trabajar con la aplicación serán una conexión a Internet y un navegador Web actualizado.

---

#### 2.2.5. SUPOSICIONES Y DEPENDENCIAS

Deberá cumplir con los siguientes requisitos:

- La aplicación requiere que los usuarios utilizaran un navegador Web para acceder a ella.
- Permitir la ejecución de código en lenguaje PHP.
- Trabajar con bases de datos MySQL.

### 2.3. REQUISITOS ESPECIFICACIÓN

---

#### 2.3.1. INTERFACES EXTERNAS

##### 2.3.1.1. INTERFACES DE USUARIOS:

Nuestra tienda se llamara Buy-online y la interfaz de la pagina se encuentra estructurada en 3 aéreas principales: una parte superior a la que llamaremos cabecera y donde se mostrara el logotipo o imagen corporativa de la empresa, además de un menú donde podemos elegir las diferentes acciones a realizar en la tienda como autenticarse, ver catalogo, registrarse, ver cesta, etc. La parte izquierda (columna izquierda) aparecerá zona de novedades, la zona de compra y zona de foro. En la zona central, aparecerán los listados y los resultados de la opción buscada o que desee consultar/ revisar el cliente.

##### 2.3.1.2. INTERFACES HARDWARE:

Los usuarios deberán disponer de un ordenador con una tarjeta de red o tarjeta de red inalámbrica y un punto de acceso que les permita una conexión a internet para poder acceder a la aplicación.

### 2.3.1.3. INTERFACES SOFTWARE:

Para que la aplicación pueda ser visualizada tanto por el cliente como por el administrador necesitaran la utilización de un navegador Web. La aplicación está preparada para que funcione en la gran mayoría de los navegadores disponibles.

### 2.3.1.4. INTERFACES DE COMUNICACIONES:

La comunicación entre el cliente y el servidor, consiste en una comunicación de petición y respuesta, estas se efectuaran con el protocolo HTTP y serán enviadas del cliente/servidor o del servidor/cliente con el protocolo TCP/IP.

---

## 2.3.2. FUNCIONES

### 2.3.2.1 CLIENTE:

- Registro de usuarios
  - Introducción: Registro de un nuevo cliente.
  - Entrada: Datos del usuario.
  - Proceso: Almacenar al cliente en la base de datos.
  - Salida: Actualizar la aplicación.
  
- Listado de productos
  - Introducción: Listado de productos para la página principal.
  - Entrada: El cliente elige una opción de menú.
  - Proceso: Se realiza una consulta de los productos en la base de datos según la opción del cliente.
  - Salida: Se muestra el resultado de la consulta por pantalla.
  
- Añadir productos a la cesta de la compra.
  - Introducción: Cualquier usuario puede añadirlo a su cesta de la compra sin adquirir ningún compromiso de compra por ello.
  - Entrada: Identificador del producto.

- Proceso: Añadir a la lista cesta de la compra el identificador de producto, en el caso de ser un usuario anónimo la cesta de la compra se perderá una vez cerrada la sesión del navegador, en caso de un usuario registrado el contenido de la cesta será almacenado en la base de datos.
- Salida: Carro de la compra actualizado o mensaje de error.

#### 2.3.2.2. ADMINISTRADOR:

- Añadir/Editar/Eliminar categorías
  - Introducción: Gestión de la zona de categorías.
  - Entrada: El administrador puede crear, modificar las categorías y eliminarlas.
  - Proceso: Si la categoría es nueva, se añaden los datos para la creación, si no es nueva se localiza la categoría a modificar o eliminar y se le aplica la acción.
  - Salida: Se actualiza la base de datos.
  
- Añadir/Editar/Eliminar productos
  - Introducción: Gestión de la zona de productos.
  - Entrada: El administrador puede crear, modificar los productos y eliminarlos.
  - Proceso: Si el producto es nuevo, se añaden los datos para la creación, si no es nuevo, se localiza el producto a modificar o eliminar y se le aplica la acción.
  - Salida: Se actualiza la base de datos.
  
- Opciones de visualización de productos
  - Introducción: Cambio de la visualización del producto.
  - Entrada: Diferentes modalidades de visualización.
  - Proceso: La visualización seleccionada se le aplica a la manera de listar los productos en la aplicación.
  - Salida: Actualización de los listados de productos.

- Generar factura pedido.
  - Introducción: A través de esta función se generará la factura asociada a una venta.
  - Entrada: Identificador de cliente, identificador de pedido.
  - Proceso: Se generará un documento con toda la información del pedido realizado por parte del cliente a partir de los datos contenidos en la tabla pedidos de la base de datos. Se deberá incluir en el documento los datos de facturación.
  - Salida: Formulario factura o mensaje de error.

---

### 2.3.3 RESTRICCIONES DE RENDIMIENTO:

La respuesta que dará el sistema con respecto a la petición del usuario deberá ser en tiempo real.

También debemos tener en cuenta el espacio que ocupara almacenar la información correspondiente a cada producto

---

### 2.3.4. RESTRICCIONES DE DISEÑO:

Para el diseño podemos utilizar un ciclo de vida en cascada ya que sus etapas nos ayudaran a simplificar planificación de las actividades.

El diseño de la aplicación facilitara el uso y una rápida elección de las opciones a los usuarios de la misma, ya que esta aplicación será utilizada por usuarios de cualquier edad y, con pocos o muchos conocimientos de Internet a nivel usuario.

---

### 2.3.5. ATRIBUTOS DEL SISTEMA:

#### 2.3.5.1. INTEGRIDAD:

Para salvaguardar la integridad de los datos, se ha optado por almacenar los mismos en una base de datos.

#### 2.3.5.2. MANTENIMIENTO:

El mantenimiento será llevado a cabo por el administrador del sistema, a quien se le facilita un módulo de administración para realizar todas las tareas necesarias.

#### 2.3.5.1. SEGURIDAD:

Los datos confidenciales de la cuenta de un cliente de de nuestra aplicación solo serán accesibles por el propietario de la cuenta y los administradores. Un usuario solo podrá acceder a sus datos tras realizar el proceso de autenticación.

---

#### 2.3.6. OTROS REQUISITOS:

##### 2.3.6.1. BASE DE DATOS:

La aplicación trabajará sobre una base de datos donde se almacenará toda la información necesaria para el funcionamiento de la tienda virtual.

### 3. ANÁLISIS

En la fase de análisis del proyecto se estudia toda la funcionalidad de la aplicación a desarrollar en la implantación. A lo largo de esta fase se describe la estructura y funcionalidad de la aplicación mediante el uso de diagramas de diversas clases.

#### 3.1 DESCRIPCIÓN DEL PROYECTO A DESARROLLAR

El software resultante del proyecto se encontrará ubicado en un servidor perteneciente a la tienda, dicho software permitirá el mantenimiento de los datos de los artículos, catálogos, usuario, etc.

#### 3.2 SEGURIDAD DEL SISTEMA

Para poder proteger los datos utilizados en esta aplicación se toma en cuenta las siguientes medidas:

3. Identificar a los usuarios que acceden a nuestra aplicación mediante la introducción de su nombre y contraseña.
4. Los usuarios que ya no utilizarán el sistema obtienen una baja que evita su acceso del sistema pero manteniendo su información en las bases de datos.
5. Solicitud de confirmación antes de completar cualquier acción delicada para el sistema o de carácter irreversible.

#### 3.3 DIAGRAMA DE CASOS DE USO

En este apartado mostraremos las funcionalidades y los comportamientos del sistema mediante su interacción con algún agente externo, ya sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

3.3.1. CASOS DE USO DEL USUARIO:

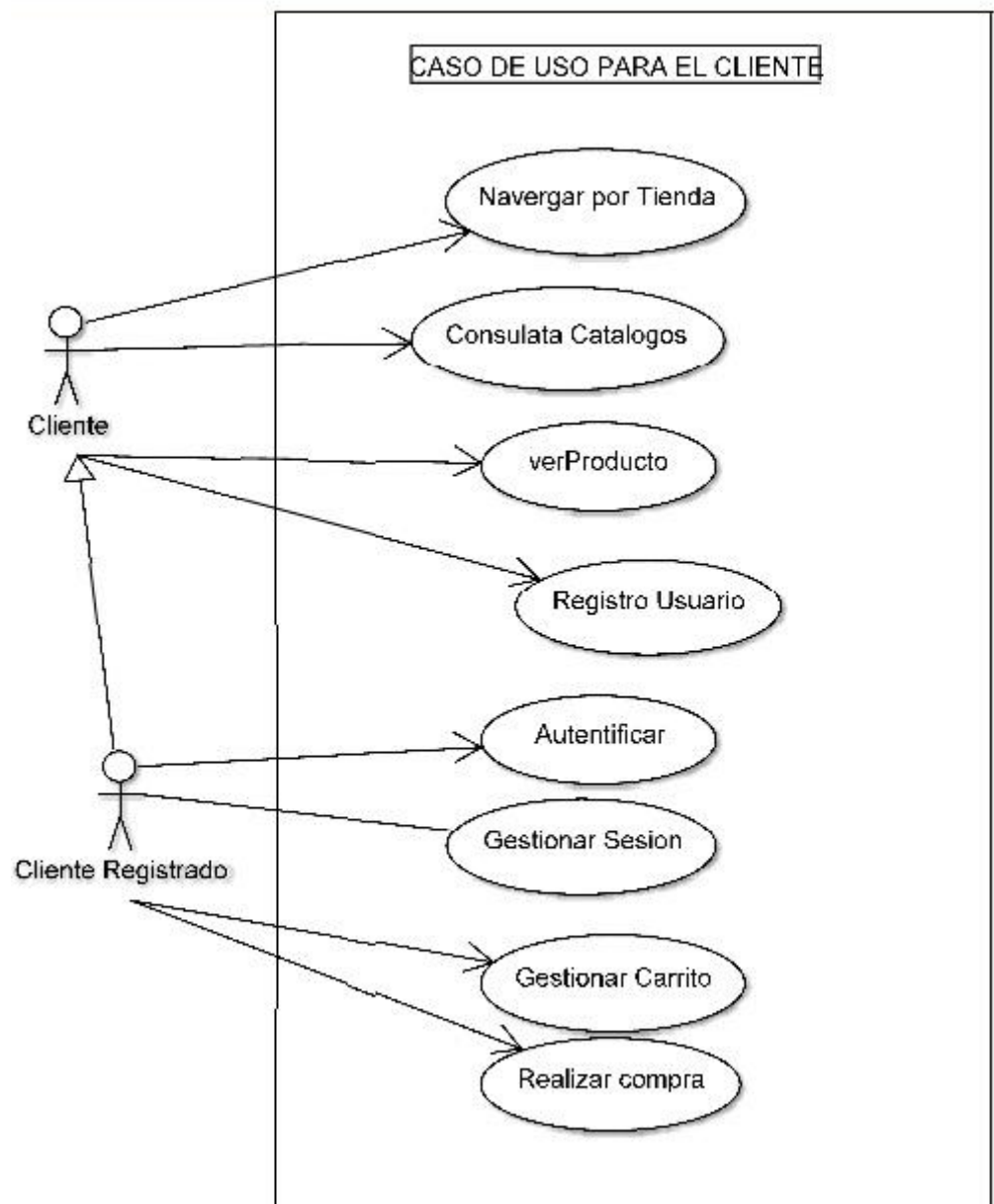


Figura 1. Caso de estudio cliente

3.3.2. CASOS DE USO DEL ADMINISTRADOR:

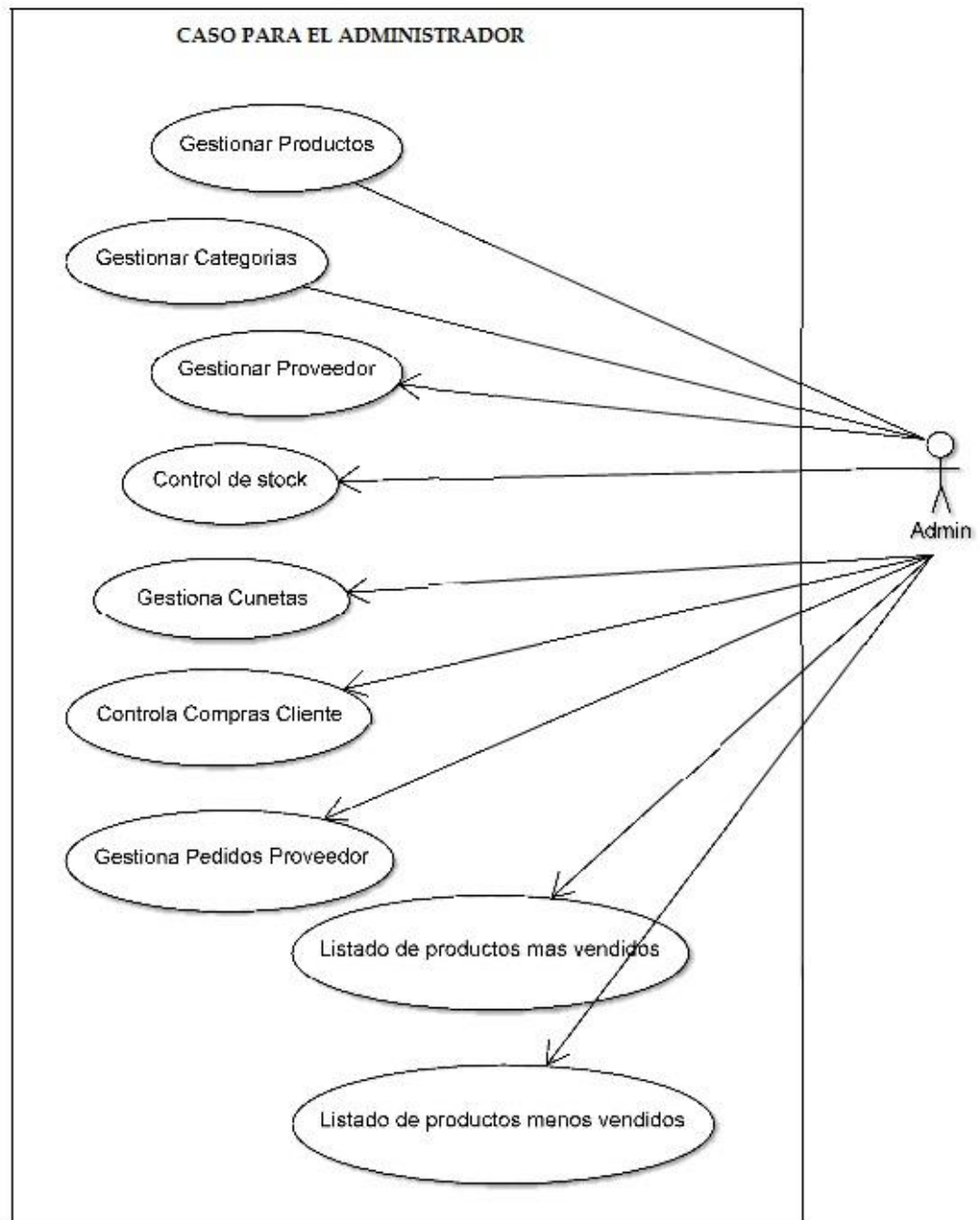


Figura 2. Caso de estudio del administrador

3.4 DIAGRAMA DE CLASES

El diagrama de clases describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos.



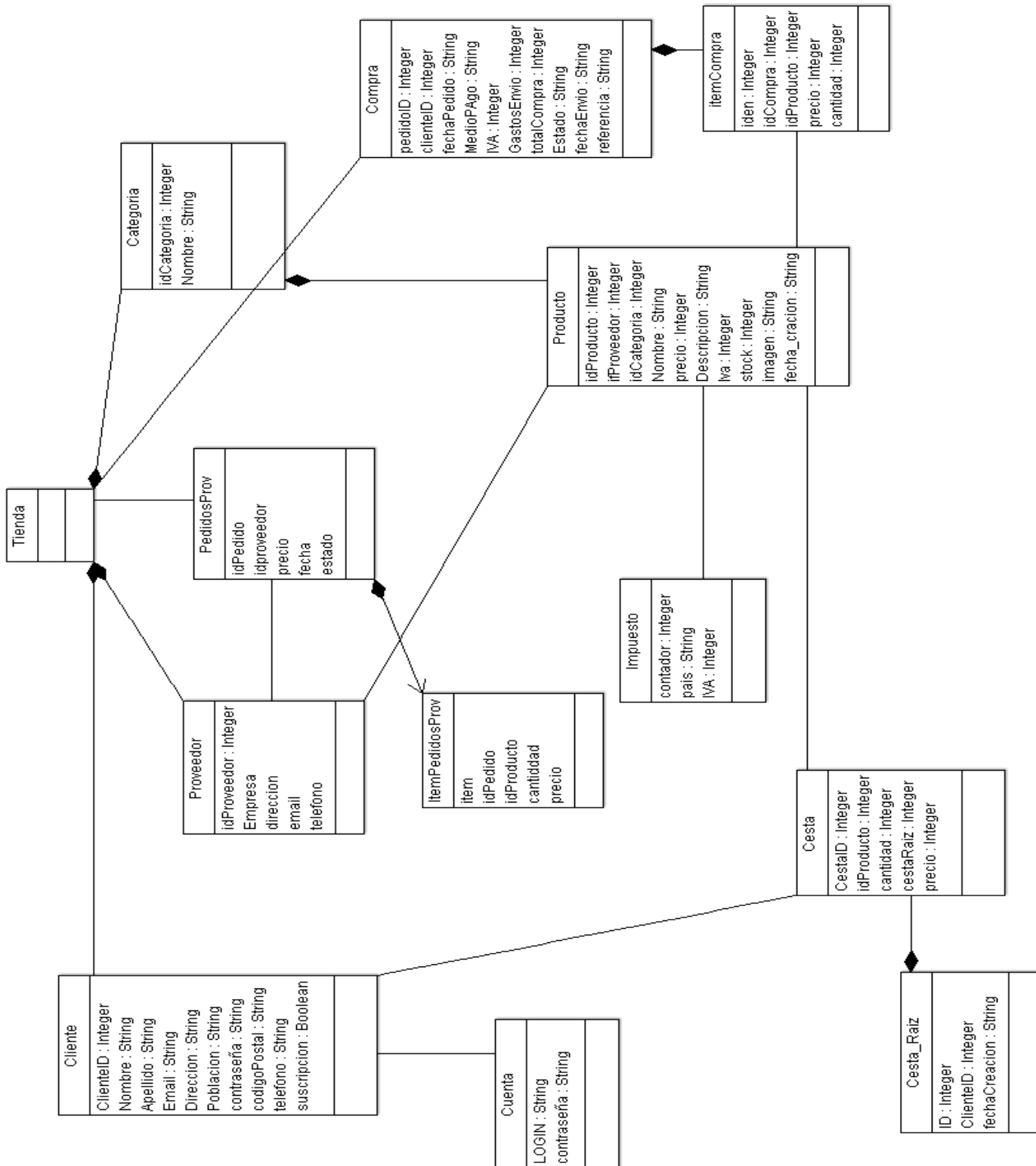


Figura 3. Diagrama de clase

### 3.5 ESCENARIOS DE LA APLICACIÓN

*Escenario: Registrar usuario*

*Descripción:* El usuario no registrado podrá darse de alta en nuestra aplicación.

1. El usuario introduce sus datos personales, incluyendo un identificador y contraseña.
2. Se comprueba que el identificador elegido no esté dado de alta, si esta dado de alta mostraremos un mensaje informándonos, en caso contrario se efectuara el alta correctamente.

*Escenario: Identificación usuario*

*Descripción:* En la web nos aparecerá un una opción de identificación donde introducimos el identificador y contraseña.

1. Comprobamos que el identificador este dado de alta, si no está mostramos un mensaje informando de ello, de lo contrario la identificación será correcta.

*Escenario: Comprar producto*

*Descripción:* 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*

*Descripción:* 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

*Descripción:* 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

*Descripción.* 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...

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

Escenario: Baja producto

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

Escenario: Listar usuarios registrados

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

Escenario: listar pedidos

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

Escenario: Cambio estado pedido

*Descripción.* 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 apartado describimos nuestra aplicación con mayor detalle, teniendo en cuenta las características de un entorno web.

### 4.2 ARQUITECTURA DEL SISTEMA

El diseño de la Tienda Virtual se basará en una arquitectura multicapas de tres capas lógicas

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

El desarrollo en capas permite separar claramente el acceso a los datos, la lógica del negocio y la interfaz del usuario. El modelo de desarrollo en capas implica que cada capa se pueda comunicar solamente con la capa que tiene a su lado.

Dicho en otros términos, desde el código de la capa de presentación no se puede realizar accesos directos a la base de datos sin pasar por la capa de negocio.

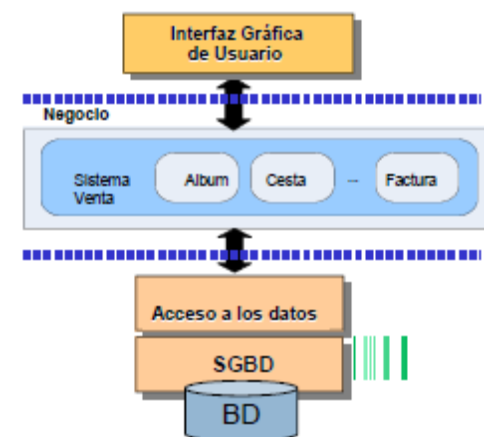


Figura 4. Arquitectura tres capas

#### 4.2.1 CAPA DE PRESENTACIÓN:

La capa de presentación es la parte de la aplicación que se encarga de las tareas de visualización en los clientes y la entrada de datos que recibe la aplicación en el cliente; resumiendo, es la propia interfaz de usuario.

Se debe tener en cuenta que una parte fundamental de este entorno de trabajo es el propio navegador del cliente que no sólo hace de receptor del código HTML que recibe, también se encarga de controlar y actuar ante los eventos que se producen en la pantalla. Si el usuario hace clic en un botón de una página PHP, será el navegador quien intervendrá en primera instancia.

La capa de presentación es responsable de:

- Obtener información del usuario.
- Enviar la información del usuario a los servicios de negocio para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.



Figura 5. Interfaz usuario

La interfaz de usuario ha sido diseñada de tal manera que resulte ameno, intuitivo y fácil de utilizar. De este modo, todas las páginas de la Tienda Virtual están compuestas por cuatro fases distinguibles fácilmente:

- Una cabecera en la que se muestra el logo y se deja un espacio para añadir los nuevos productos.
- Un menú horizontal desde el que se puede acceder a algunas de las funciones de la página Web desde el que en determinadas funciones.
- La columna izquierda con la ventana de catálogo con la que puedes acceder a los productos de cada una de ellas, también puedes acceder y registrarte a la tienda.
- La ventana principal en la que se muestra toda la información al usuario y se recogen los datos que este introduzca

---

### 4.2.2 CAPA DE NEGOCIO:

La capa de negocio está formada por un conjunto de librerías que implementan las clases del dominio. Este nivel es el encargado de realizar todas las operaciones a nivel de aplicación.

Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos (persistencia), para solicitar al gestor de base de datos el almacenaje o la recuperación de estos.

La capa de negocio es responsable de:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de almacenaje o recuperación de estos.
- Enviar el resultado procesado al nivel de presentación.

---

### 4.2.3 CAPA DE PERSISTENCIA

La capa de negocio es la que suministra los datos de la aplicación. Lo forman la base de datos y el SGBD, los encargados de almacenar toda la información del portal y

permitir el acceso a la misma de forma controlada y segura. En realidad, recibe datos suministra datos a la aplicación. Solamente se comunica con la capa de negocio; esto quiere decir que solo una clase de la capa de negocio debería poder acceder a los datos de esta capa.

La capa de persistencia es responsable de:

- Almacenar los datos.
- Recuperar los datos.
- Mantener los datos.
- La integridad de los datos.

## 5. IMPLEMENTACIÓN

En este apartado describiremos los aspectos más significativos de nuestra implementación

### 5.1 TECNOLOGÍAS

En este apartado hablaremos de forma resumida sobre las tecnologías empleadas en este proyecto.

#### 5.1.1 MYSQL

MySQL es un sistema de gestión de bases de datos relacional (inglés RDBMS - Relational DataBase Management System) multi-hilo(multithread) y multiusuario, que usa el lenguaje SQL (Structured Query Language).

MySQL se ha convertido muy popular gracias a su velocidad al ejecutar consultas y su apoyo de forma nativa por parte del lenguaje PHP(hasta la versión 4.X de este lenguaje ya que a partir de la versión 5 deja de estar lo), en la elaboración de aplicaciones web, en el entorno del software libre .

Se puede hacer uso de MySQL en aplicaciones de todo tipo (web, de escritorio u otros) de forma libre y gratuita bajo las condiciones de la licencia GPL . Si se quiere integrar MySQL como parte de un producto privativo es necesario adquirir una licencia de uso específica para este propósito.



Figura 6. Instalador de MySQL Server 5.0



### 5.1.2 PHP (PHP HYPERTEXT PROCESSOR)

PHP es un lenguaje de programación de uso general de script del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

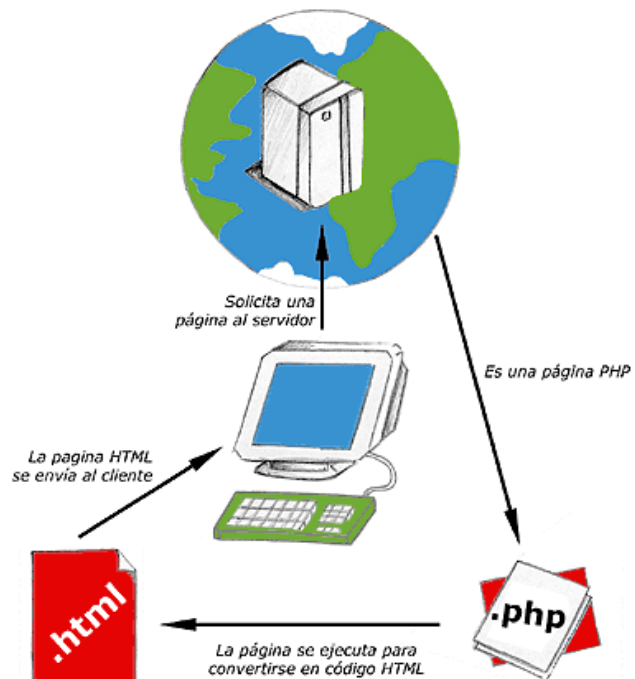


Figura 7. Petición de una página en PHP

### 5.1.3 HTML (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.

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

HTML

Figura 8. Ejemplo HTML

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).

### 5.1.4 CSS (CASCADING STYLE SHEETS «HOJAS DE ESTILO EN CASCA-DA»)

El nombre viene del inglés, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

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

```
body {
  margin: 4px;
  border: 3px dotted #;
  font-family: sans-serif;
  color: #000000;
  background-color: #FFFFFF;
}

h1 {
  padding: 5px;
  margin: 10px;
  border: 1px solid #COCOCO;
  color: #FF0000;
  background-color: #0000FF;
}
```

CSS

Figura 9. Ejemplo Hojas De Estilo En Cascada

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".

#### 5.1.5 PAYPAL

PayPal es la forma rápida y segura de pagar en Internet. Puede pagar fácilmente sus compras con la tarjeta de débito, la tarjeta de crédito o la cuenta bancaria a través de PayPal.

¿Por qué utilizar PayPal?

PayPal es seguro: PayPal almacena con seguridad su información financiera y la protege con sistemas de seguridad y de prevención de fraude líderes del mercado. Es más, sus detalles financieros nunca se comparten con el vendedor.

PayPal es **rápido y sencillo**: no tendrá que introducir los detalles de su tarjeta o su cuenta bancaria para pagar. Sólo necesitará su dirección de correo electrónico y una contraseña.

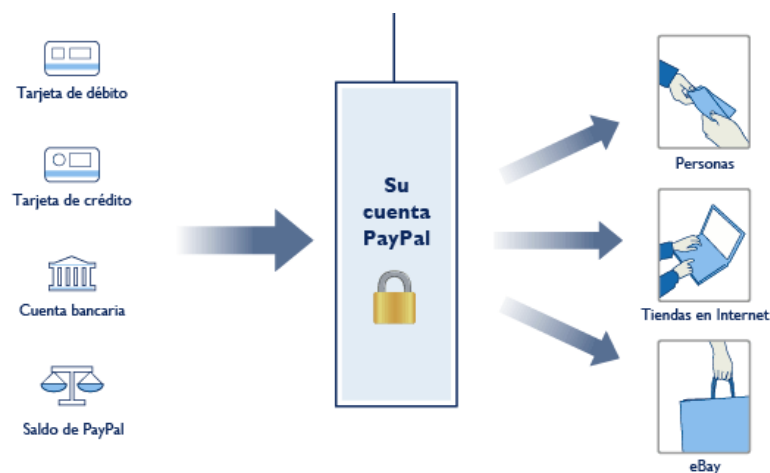


Figura 10. Funcionamiento de Paypal

### 5.1.6 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.

## 5.2 HERRAMIENTAS

### 5.2.1 APPSERV

Es una herramienta OpenSource para Windows con Apache, MySQL, PHP y otras adicciones, en la cual estas aplicaciones se configuran en forma automática, lo que permite ejecutar un servidor web completo.

Como extra incorpora phpMyAdmin para el manejo de MySQL.

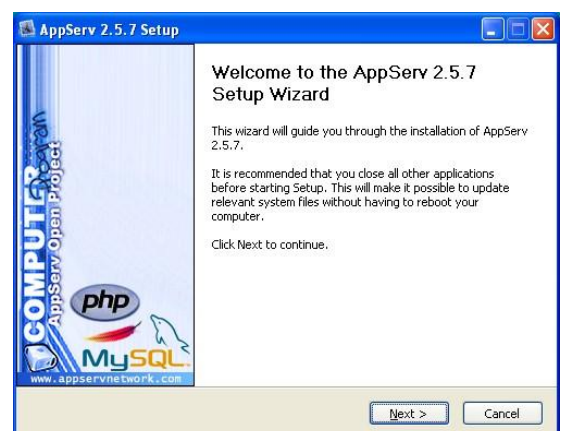


Figura 11. Instalador de Appserv

### 5.2.2. APACHE

El servidor Apache es un servidor HTTP de código abierto para plataformas Unix, Windows, Macintosh, etc. que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPD 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Ahora Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

### 5.2.3 PHPMYADMIN

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra disponible bajo la licencia GPL

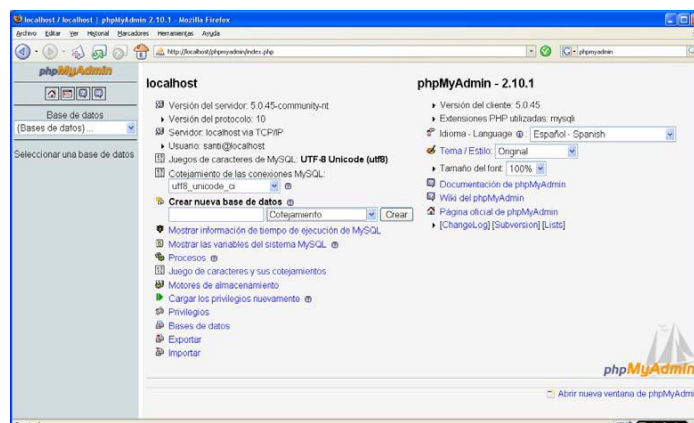


Figura 12. Apariencia del phpMyAdmin

#### 5.2.4 ARGOUML

Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java.

El Magazine de Desarrollo de Software entrega premios anuales a herramientas de desarrollo de software populares en varias categorías. En 2003 ArgoUML fue una de las finalistas en la categoría "Design and Analysis Tools". ArgoUML recibió un premio "runner-up"(revelación), derrotando a muchas herramientas comerciales.

Sin embargo, desde la versión 0.20, ArgoUML está incompleto. No es conforme completamente a los estándares UML y carece de soporte completo para algunos tipos de diagramas incluyendo los Diagrama de secuencia y los de colaboración.

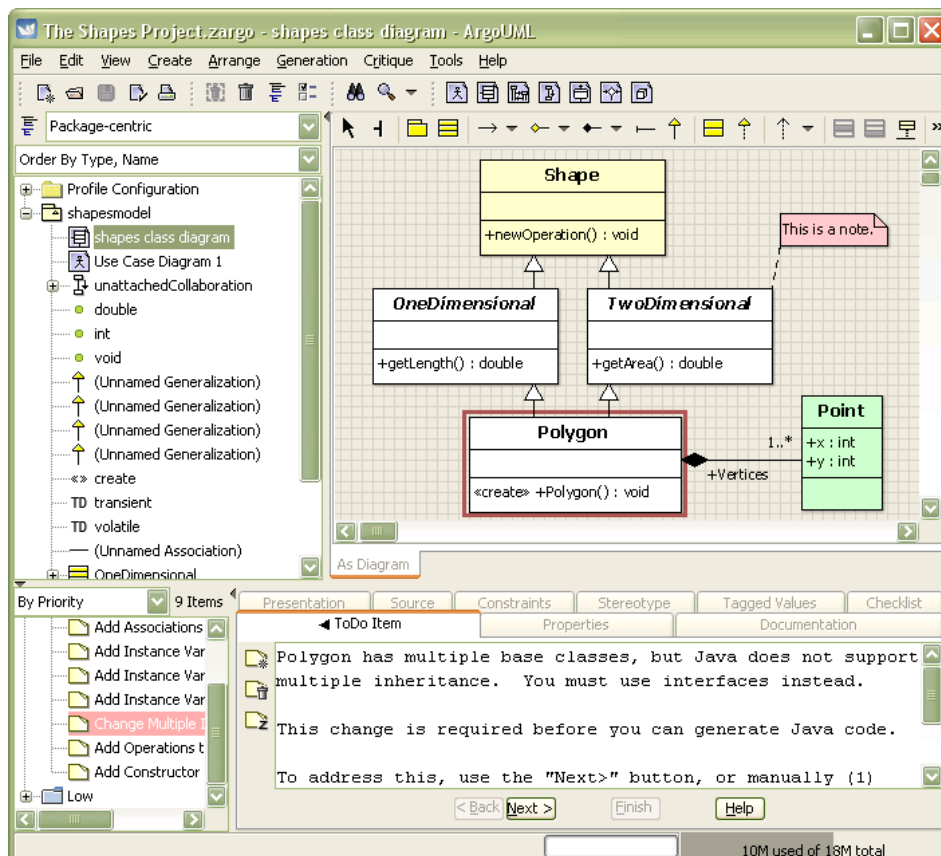


Figura 13. Interfaz de usuario de ArgoUML

## 5.3 DETALLES DE IMPLEMENTACIÓN.

### 5.3.1. TIENDA PARA EL ADMINISTRADOR

En la administración mostramos el control y monitorización de la tienda como por ejemplo.

En la página principal aparecen los productos que han sido introducidos en el último en la tienda. Esto lo identifico por la variable fecha, de igual manera estos mismos productos me aparecen en la página principal de la tienda.

#### 5.3.1.1. CATALOGO:

- **Categoría:** Mostramos un listado de las categorías de la tienda ordenado por el número de identificación.
  - En esta página también podremos:
    - ✓ Añadir: accediendo a la página addCategoria.php en la mayor complejidad ha sido añadir la imagen creando una ventana externa (gestionImagen.php), en la que cargamos la imagen pero nos guardamos la referencia de la imagen, es decir, el nombre de la imagen, en nuestra base de datos.
    - ✓ Editar y eliminar.- En estas páginas pasamos por método GET los identificadores de los productos.
- **Productos:** Al igual que en categorías, mostramos un listado de los productos de nuestra tienda ordenado por el número de identificación.
  - En esta página mostramos todos los atributos del producto (id, nombre, stock, precio e imagen)
  - En esta página también podremos:
    - ✓ Añadir: accediendo a la página addProductos.php en la mayor complejidad ha sido añadir la imagen creando una ventana externa (gestionImagen.php), en la que cargamos la imagen pero nos guardamos la

referencia de la imagen, es decir, el nombre de la imagen, en nuestra base de datos.

- ✓ Editar y eliminar.- En estas páginas pasamos por método GET los identificadores de los productos.

- **Proveedores:** Mostramos un listado de los proveedores de la tienda y al igual que los anteriores listado podemos añadir, editar y eliminar.

### 5.3.1.2. CLIENTES:

- **Cuentas:** En cuentas de usuario mostrábamos el login y el password del usuario y haciendo click en ver podemos visualizar con más detalle los datos del nuestro usuario seleccionado (el identificador es pasado por el método GET a la página datosUsuario.php).

- **Compras/Clientes:** Mostramos las compras realizadas en nuestra tienda, indicando el numero de pedido, el identificador del cliente, fecha de pedido, estado del pedido, fecha de entrega, total de compra, impuestos, gastos de envío y referencia de la compra.

- ✓ Detalle de compra: También encontramos un enlace al detalle de la compra en el que podemos visualizar los productos de la compra seleccionada.

- **Mejores clientes:** En este apartado mostramos los clientes que tienen mayor número de compras (el porcentaje de mayor compra seria mayor o igual al 20% del total de la compra). Los datos que se muestran seria: los identificadores, nombres del cliente, teléfono y números de compras.

### 5.3.1.3. CONTROL STOCK:

- **Monitorización:** en este apartado simplemente mostramos las categorías que no tiene relación con algún producto y los productos que no tienen stock.



- **Productos con poco stock:** definimos como elementos con poco stock a los productos que tienen un stock inferior o igual a 20, por lo que cuando vamos hacer pedidos a nuestros proveedores visualizamos de que productos debemos hacer pedidos y a quienes debemos hacerlos.
  
- **Pedidos a proveedores:** Mostramos los pedidos realizadas en nuestros proveedores, indicando el numero de pedido, el identificador del proveedor, fecha de pedido, estado del pedido, fecha en la que se ha realizado y total de compra.
  - ✓ Detalle de pedido: También encontramos un enlace al detalle del pedido en el que podemos visualizar los productos relacionados con el pedido.

#### 5.3.1.4. VENTAS.

- **Productos más vendidos:** En este apartado mostramos un listado de los productos más solicitados por nuestros clientes. Mostramos el identificador del producto, el nombre del mismo y el total de compras que ha tenido.
  
- **Productos menos vendidos:** En este apartado mostramos un listado de los productos menos comprados por nuestros clientes. Mostramos el identificador del producto, el nombre del mismo y el total de compras que ha tenido.

#### 5.3.1.5. CONFIGURACIÓN:

- **Administrador:** Mostramos un listado de los administradores de nuestra tienda ordenado por el número de identificación.
  - En esta página mostramos el login, el password del usuario y el perfil del administrador.
  - En esta página también podremos:

- ✓ **Añadir:** accediendo a la página addAdmin.php introducimos los datos básicos de nuestro nuevo administrador (login, password y perfil).
- ✓ **Editar y eliminar.-** En estas páginas pasamos por método GET los identificadores del administrador.

---

### 5.3.2. TIENDA PARA EL CLIENTE

En la nuestra tienda tenemos distintas funcionalidades para el cliente. En la página principal encontramos con las categorías y el acceso de los usuarios.

#### 5.3.2.1. MODO DE ACCESO:

- **Dar de alta:** les pediremos al nuevo usuario que nos facilite datos básicos como son: nombre y apellido, dirección, postal, teléfono, etc. Una vez dado de alta le pediremos que acceda con esta nueva cuenta a nuestra página.
- **Acceso a cuenta:** una vez introducido su correo (login) y su password crearemos unas variables de sesión tanto para identificarlos con el login, password y el identificador del cliente (esta variable sesión nos facilita el proceso de creación de cesta de compra y creación del pedido).

*Un detalle a tener en cuenta es que, al acceder con nuestro login y password automáticamente las opciones de acceso desaparecen, con lo que no aparecería el correo del usuario conectado el cual contaría con dos opciones*

- ✓ **Modificar:** en esta página le mostrara al cliente los datos que ha introducido a la hora de crearse su cuenta y le facilitara la modificación de cualquiera de sus datos.
- ✓ **Salir de la sesión:** Con lo que respecta a el cierre de la sesión de nuestro usuario simplemente destruyo la sesión (ses-

sion\_destroy() ) y automáticamente me redirecciono a la página principal (header("Location: index.php")).

#### 5.3.2.2. CATALOGO:

Solo mostrara las categorías que tienen productos en ella. Al seleccionar una categoría nos encontramos con todos los productos disponibles de la categoría, el cual visualizaremos con más detalle si accedemos a él.

- **Producto:** Mirando con más detalle las características del producto nos encontraremos con dos alternativas, dependiendo de si se ha iniciado sesión o no.
  - ✓ **Sesión no iniciada:** le aparecerá un mensaje pidiéndole que se dé de alta o que inicie la sesión.
  - ✓ **Sesión iniciada:** le aparecerá la opción añadir a cesta.

#### 5.3.2.3. CARRITO DE LA COMPRA:

El cliente podrá visualizar en cualquier momento su carrito de la compra. En el podemos ver con detalle los productos agregados, también nos mostrara detalles como el IVA de la compra, los gastos de envío y el total final; en cualquier momento el cliente podrá anular un producto no deseado del carrito.

En este apartado crearemos una nueva variable de sesión “MM\_CARRITO” la cual nos facilitara el recuperar los valores de la cesta del cliente.

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.

### 5.3.2.4. FORMA DE PAGO:

Al visualizar nuestra cesta nos encontraremos con la opción “seleccionar forma de pago” la cual nos re direccionará a la página formaPago.php, en la cual seleccionaremos una de las dos opciones de pago.

- **Paypal:** seleccionando este medio de pago nos redirecionamos a la pagina '<https://www.sandbox.paypal.com/cgi-bin/webscr>', pasando por la pagina finalizacionPagos.php, a la cual le enviaremos una serie de datos como son:
  - *finalizacionPagos.php*: esta página pasara los siguientes argumentos o atributos:
    - ✓ **cmd:** Indica el tipo de fichero que va a recoger PayPal, que pueden ser: `_cart` (varios items), `_donations` (donaciones), `_xclick` (boton de compra).
    - ✓ **business:** Indica el identificador del negocio registrado en paypal. Aquí vendrá la dirección e-mail de registro en Paypal del vendedor.
    - ✓ **shopping\_url:** La dirección de nuestra tienda online.
    - ✓ **currency\_code:** El tipo de moneda (USD , EUR ...)
    - ✓ **return:** Sera el enlace de vuelta a nuestro negocio que ofrece paypal

- ✓ **notify\_url:** En esta página es donde recogeremos el estado del pago y un gran número de variables con información adicional en nuestro caso lo hemos llamado paypalipn.php
- ✓ **rm:** Indica el método que se va a utilizar para enviar la información de vuelta a nuestro sitio. RM=1 (información enviada por GET) , RM=2 (información enviada por POST)(En este caso usamos este método porque es un script php el que recoge los datos).
- Recuperaremos de una página datosCesta.php por la llamada a la función datosCesta(SESION\_CARRITO); los siguientes atributos
  - ✓ **item\_number\_X:** Identificador del producto. Aquí tendremos que definir tantos productos como haya en el carro de compra. Ejemplo item\_number\_1, item\_number\_2, etc ..
  - ✓ **item\_name\_X:** Nombre del producto. Aquí tendremos que definir tantos nombres de productos como haya en el carro de compra. Ejemplo item\_name\_1, item\_name\_2, etc ..
  - ✓ **amount\_X:** Precio del producto. Aquí tendremos que definir tantos precios de producto como haya en el carro de compras. Ejemplo amount\_1, amount\_2, etc ..
  - ✓ **quantity\_X:** Cantidad del producto. Aquí tendremos que definir tantas cantidades de producto como haya en el carro de compras. Ejemplo quantity\_1, quantity\_2, etc ..

- ✓ **shipping\_X:** Es el costo de envío de cada producto. Aquí tendremos que definir tantas cantidades de producto como haya en el carro de compras. Ejemplo shipping \_1, shipping \_2, etc ..
  
- ✓ **tax\_X:** Es el impuesto de cada producto. Aquí tendremos que definir tantas cantidades de producto como haya en el carro de compras. Ejemplo tax \_1, tax \_2, etc ..

Una vez terminada el proceso de pago, este nos devolverá (por método POST) todos los datos necesarios para registrar nuestro pedido.

- **Transferencia:** Mostraremos un mensaje indicando nuestro número de cuenta bancaria y que la confirmación de su compra se enviara una vez haya sido cancelada la compra.

### 5.3.2.5. COMPRA DEL CLIENTE

A continuación pasaremos a describir como creamos nuestro registro de pedidos.

- **Por PayPal:** Como hemos mencionado anteriormente, PayPal nos envía por método post los datos necesarios para registrar nuestros pedidos.
  - *Creación de compra:* de los métodos recibidos utilizaremos solo los esenciales para la creación del pedido de compra como son:
    - ✓ **mc\_gross:** que nos indica el total total de compra, pero sin sumarle el coste del gasto de envío.
  
    - ✓ **Tax:** nos indica el impuesto del toda la compra
  
    - ✓ **payment\_status:** simplemente nos indica si el pago ha sido completado o no.

- ✓ **mc\_shipping:** el total del gasto de envío.
- ✓ **txn\_id:** el cual utilizaremos como referencia de la compra.

Con lo recibido de PayPal le añadiremos datos como:

- ✓ **El identificador del pedido:** recuperado de la variable de sesión.
- ✓ **Fecha del pedido:** este dato se crea en nuestro código ya que el recibido por PayPal es el de su servidor.
- ✓ **Medio de pago:** aquí le indicamos la si el pago ha sido efectuado por transacción o por PayPal,
- ✓ **Total cargo:** que es la suma del total de la compra de los productos y los gastos de envío.
- ✓ **Fecha de envío:** donde nos creamos una variable fecha envío añadiéndole 5 días a la fecha de compra actual.

Cada compra tendrá una lista de productos relacionado

- Creación de *itemCompra*: como en la clase compra, de los métodos recibidos utilizaremos solo los esenciales para la creación del pedido de compra como son:
  - ✓ **item\_nameX** : nos indica el nombre del artículo de la compra.
  - ✓ **quantityX**]: pasamos la cantidad de artículos
  - ✓ **mc\_gross\_X**: indicamos el precio del pedido.

- ✓ **taxX:** indicamos el impuesto relacionado a el producto.

A esta relación le agregamos:

- ✓ **idCompra:** recuperamos el identificador de la compra y lo compráramos con la hora actual del sistema, si estos coinciden se agregara el ítem en relación a la compra, si la fecha y hora son igual (ya que el cliente puede realizar muchas compras en un día) se agrega el producto.

### 5.3.2.6. EMAIL COMPRA

En esta página mostramos los datos del cliente, los cuales han sido recuperados de paypal y serán los siguientes.

- **first\_name:** nombre del usuario en paypal
- **last\_name:** apellido del cliente
- **address\_name:** dirección del cliente
- **business:** email de nuestra cuenta en paypal
- **address\_country:** país de origen.
- **address\_city:** ciudad de origen
- **payer\_email:** Email del cliente recuperado de paypal
- **txn\_id:** numero que hemos utilizado como referencia



Mostraremos en este apartado lo que se le enviaría al cliente por correo lo que ha sido su compra.

## 6. PRUEBAS

En este capítulo, terminada toda la parte de implementación, va a estar dedicado a las pruebas de funcionamiento y evaluación de resultados de la aplicación terminada

### 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:

- La facilidad de comprensión de la estructuración de la aplicación, así como de las funcionalidades y contenidos que en esta pueden observarse.
- La sencillez de uso de la aplicación en las etapas iniciales.
- La rapidez con la que el usuario encuentra aquello que busca.
- La facilidad percibida en la navegación en términos de tiempo requeridos y tareas precisadas para obtener los resultados perseguidos.
- La capacidad del usuario del sistema para controlar aquello que hace y el lugar en el que se encuentra en cada momento.

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.

---

#### 6.1.1. PRUEBAS DE VALIDACIÓN UNITARIAS

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, probaremos toda la aplicación web tanto la parte del cliente (catálogo) como el apartado de administración.

- ❖ Por parte de cliente se comprobó:
  - **Navegación del menú categorías:** Comprobación de la navegación en el menú categorías, productos asignados y visualización.



Figura 14. Productos categorías

Como podemos observar, nos aparecen los productos que se encuentran asignado en la categoría Música, y de cada uno de ellos podemos observar una imagen, el nombre del producto y su precio.

- **Especificación del producto:** Visualización de las características específicas de un producto determinado.

## GIBSON ES335 PLAIN CHERRY (ESDPCHNH1)



2149.00 €

Pastillas 57 classics Clavijero Grover Puente tune-omatic Incluye estuche y certificado de autenticidad. Cuerpo de arce M

[Añadir a cesta](#) (Importante: Al total de la compra se le añadira el 5% por los gastos de envío)

### Productos relacionados

YAMAHA YAS-280 SAXO ALTO



849.00 €

Figura 15. Características producto

Una vez seleccionado un producto determinado, podemos ver su descripción y también podemos observar los productos relacionados.

- **Registro:** Registro de varios clientes.

## Alta de usuario

formulario de alta en la tienda:

Nombre:

Apellidos:

Email:

PalabraPaso:

Direccion:

Provincia:

CodigoPostal:

Telefono:

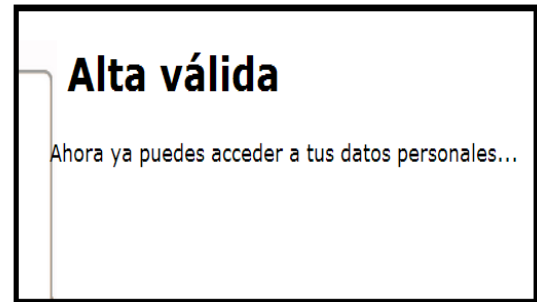


Figura 16. Alta usuario

Hemos introducido distintos clientes y todos se han creado correctamente.

## Accede a tu cuenta

Accede al contenido de tu cuenta

Email:

Contraseña:

Figura 17. Acceso usuario

Accedemos a la tienda correctamente con el cliente creado y la tienda nos da la bienvenida.

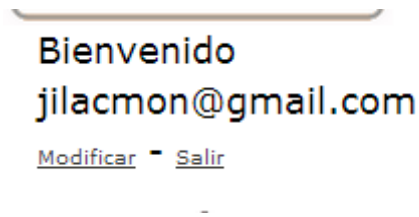


Figura 18. Bienvenida usuario

- **Proceso de compra:** Seguimiento del proceso de todo el proceso de compra de varios productos. Visualización y gestión del carrito.

Una vez añadido productos a mi carrito nos dirigimos a un listado de productos de nuestro carrito, en el podemos actualizar las unidades y eliminar algún producto no deseado.

También hemos comprobado que cada vez que añadamos el mismo producto recalcula las unidades y el precio que tenía en la cesta.




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.

**lista carrito**

contenido

Producto	Precio/Unidad	Unidades	Precio/Total	Acciones
TS-L462 IDE CD-RW	19.51€	<input type="text" value="1"/> 	19.51 €	<a href="#">Eliminar</a>
YAMAHA YAS-280 SAXO ALTO	849.00€	<input type="text" value="1"/> 	849.00 €	<a href="#">Eliminar</a>
Trangoworld Zimal	289.00€	<input type="text" value="1"/> 	289.00 €	<a href="#">Eliminar</a>
Subtotal sin IVA:			1,157.51€	
IVA: (21%)			243.08€	
Subtotal con IVA:			1400.59 €	
Cargo de Envío: (5%)			57.88€	
Total:			1,458.47 €	

[Seleccionamos la forma de pago](#)

Figura 19. Selección carrito

- ✓ En el listado de carrito de compra también nos encontramos con un enlace que no redirecciona a la selección del medio de pago que se va a realizar por la compra.

He este apartada podremos elegir entre pagar a través de la plataforma paypal o por transferencia.

## Forma de pago

Escoge tu forma de pago.

- Paypal
- Transferencia

Selección del medio de pago

Pagar

Figura 20. Selección medio de pago



- Si elegimos pagar por paypal, al dar en el botón de pagar nos enviara a la página de paypal en el cual debemos introducir nuestros datos como usuario paypal.

### Resumen de su pedido

Descripciones	Importe
Barbacoa para mesa Brisbane Número de artículo: 13 Precio del artículo: €71,50 Cantidad:1	€71,50
<b>Importe total</b>	<b>€71,50</b>
<b>Total €71,50 EUR</b>	

### Seleccione una forma de pago

PayPal procesa de forma segura los pagos para fanny montañez's Test Store.

#### ¿Tiene una cuenta PayPal?

Inicie sesión en su cuenta para pagar

Correo electrónico

Contraseña de PayPal

[¿Ha olvidado su contraseña o dirección de correo electrónico?](#)

#### ¿No dispone de una cuenta PayPal?

Pagar con tarjeta de débito o crédito como usuario invitado de PayPal

[Cancelar y volver a fanny montañez's Test Store.](#)

Figura 21. Acceso paypal

Una vez introducido nuestros datos como usuario paypal, procederemos a pagar nuestra compra.

57

### Resumen de su pedido

Descripciones	Importe
Barbacoa para mesa Brisbane Número de artículo: 13 Precio del artículo: €71,50 Cantidad: 1	€71,50
<b>Importe total</b>	<b>€71,50</b>
<b>Total €71,50 EUR</b>	

### Revise su información

**Pagar ahora**

**Dirección de envío** [Cambiar](#)

Jill ACosta  
calle Vilamar 76993- 17469  
02001 Albacete  
España

Nota para el vendedor: [Añadir](#)

---

**Formas de pago**

Saldo de PayPal €71,50 EUR

Tarjeta de regalo o cupón de descuento de PayPal [Cambiar](#)

---

**Información de contacto**  
lij\_yn\_1351185817\_per@hotmail.com

**Pagar ahora**

Figura 22. Pago paypal

Al terminar el pago el cliente debe volver a nuestra tienda para que el pago sea completado.

### fanny montañez's Test Store

Acaba de realizar un pago de

**€71,50 EUR**

[Imprimir recibo](#)

**Pagado a**  
fanny montañez's Test Store  
lij\_yn\_1351185527\_biz@hotmail.com

**Enviar a**  
Jill ACosta  
calle Vilamar 76993- 17469  
02001 Albacete  
España

**Se ha completado el pago correctamente**

✓ Acaba de completar el pago.

El Id. de transacción de este pago es: 4GG84578HU316145G.

Enviaremos una confirmación por correo electrónico a  
lij\_yn\_1351185817\_per@hotmail.com.

[Volver a fanny montañez's Test Store](#)  
[Ir a Descripción general de PayPal](#)

Figura 23. Volver a tienda

- Si el pago es realizado por transferencia, en esta página nos aparecerá un número de cuenta para que efectuemos el ingreso del importe total del pedido.

## fin compra

pago por transferencia

debes emitirnos un email cuando el pago por transferencia se halla realizado al siguiente numero de cuenta

ISBN: 16494626848648941648416

finalizar compra

Figura 24. Compra por transferencia

Al terminar cualquier proceso de compra, 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. Luego nos dirigiremos emailCompra.php, donde les informaremos al cliente sus datos y el total de su compra. También le indicaremos al cliente que le será enviado un correo electrónico indicándole el detalle de su pedido.

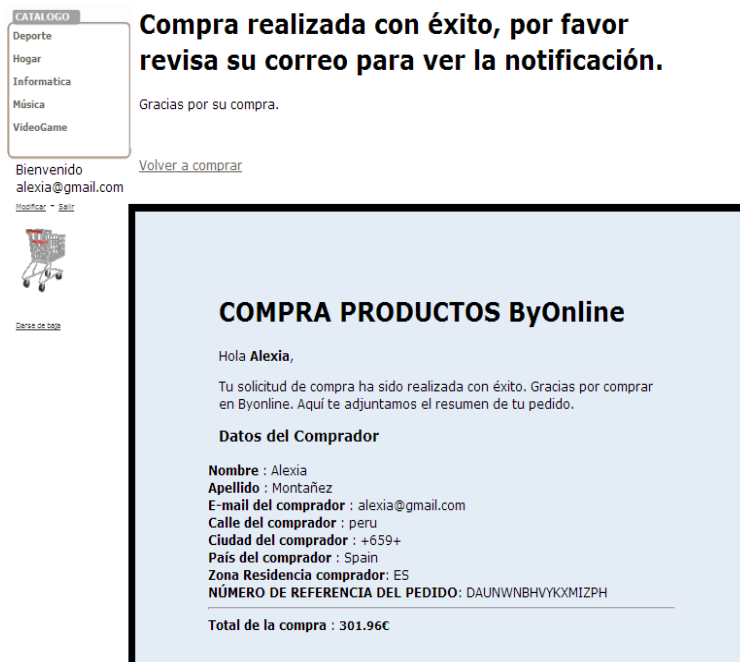


Figura 25. Email compra

- ❖ En cuanto a las comprobaciones por parte del administrador fueron:
  - **Seguridad y autenticación:** Comprobación de la seguridad y la correcta autenticación del administrador registrado, para un correcto y seguro acceso a la zona de administración del catálogo.



Figura 26. Acceso admin

Una vez haya accedido con mi cuenta de usuario admin me direccionare a principal.php donde ya puedo encontrarme con las distintas funcionalidades de mi tienda administrador



Figura 27. Página principal administración

- **Gestión del catálogo:** En este apartado contamos con tres diferentes listados, listado de categorías, listado de productos y listados de proveedores.

ID	NOMBRE	ACCIONES
1	Deporte	<a href="#">Editar</a> - <a href="#">Eliminar</a>
2	VideoGame	<a href="#">Editar</a> - <a href="#">Eliminar</a>
3	Hogar	<a href="#">Editar</a> - <a href="#">Eliminar</a>
4	Informatica	<a href="#">Editar</a> - <a href="#">Eliminar</a>
5	Música	<a href="#">Editar</a> - <a href="#">Eliminar</a>
10	Prensa	<a href="#">Editar</a> - <a href="#">Eliminar</a>

Administración Tienda Virtual  
Hora: 11:17:27 Fecha: 12/12/2012

Figura 28. Listado de categorías

Como podemos observar en el listado de las categorías () nos encontramos con las opciones de añadir, editar y eliminar.

Añadiendo ...

Categoria:

Proveedor:

Nombre:

Precio:

DescripGrande:

DescripPequena:

IVA:

Stock:

Etiqueta:

Imagen:

Figura 29. Ejemplo de añadir producto

## Editar Proveedor

modificando datos del proveedor ...

Empresa:

Dirección:

Email:

telefono:

Figura 30. Ejemplo de editar proveedor

- **Gestión de pedidos:** Gestión de los pedidos realizados por los clientes.

Buy-Online

Administración

- Catálogo
- Cientes
- Control Stock
- Ventas
- Configuración

### Lista de pedidos

ID	CLIENTE	FECHA PEDIDO	IVA	GASTO DE ENVIO	TOTAL CARGO	ESTADO	FECHA ENVIO	REFERENCIA	ACCIONES
1	1	2012-11-19 19:01:34	22.89	5.45	142.00	Completed	2012-11-24	7RF56226XR1311717	<a href="#">Ver</a>
2	1	2012-11-19 19:05:18	60.69	14.45	378.00	Completed	2012-11-24	1F855411XA7997534	<a href="#">Ver</a>
3	1	2012-11-20 00:41:24	22.89	5.45	114.00	EnProceso	2012-11-25	6JS237101M7665946	<a href="#">Ver</a>
4	1	2012-11-22 13:23:19	22.89	5.45	114.00	EnProceso	2012-11-25	6JS237101M7665946	<a href="#">Ver</a>
5	1	2012-11-22 13:23:19	22.89	5.45	114.00	EnProceso	2012-11-27	Gm8mK0dD36	<a href="#">Ver</a>
6	1	2012-11-22 13:43:12	22.89	5.45	5.00	EnProceso	2012-11-27	BSV6A6ERUM	<a href="#">Ver</a>
7	1	2012-11-22 13:46:00	22.89	5.45	5.00	EnProceso	2012-11-27	URNOD00WLJ	<a href="#">Ver</a>
8	1	2012-11-22 13:51:05	22.89	5.45	114.00	EnProceso	2012-11-27	KUHMZL09F3	<a href="#">Ver</a>
9	1	2012-11-22 13:51:59	22.89	5.45	114.00	EnProceso	2012-11-27	O968TWLOE1	<a href="#">Ver</a>
10	1	2012-11-22 13:52:49	22.89	5.45	142.00	EnProceso	2012-11-27	3CDSQ88DCK	<a href="#">Ver</a>
11	1	2012-11-22 13:53:52	22.89	5.45	137.00	EnProceso	2012-11-27	53BTL0WCY7	<a href="#">Ver</a>
12	1	2012-11-22 13:54:37	22.89	5.45	137.00	EnProceso	2012-11-27	9CDZB2GI18	<a href="#">Ver</a>
13	1	2012-11-22 13:55:10	22.89	5.45	137.00	EnProceso	2012-11-27	D6YQ0VRFUHRSYJJOQ	<a href="#">Ver</a>
14	1	2012-11-22 13:59:31	22.89	5.45	137.00	EnProceso	2012-11-27	UQCQAEV6GEMES52R	<a href="#">Ver</a>
15	1	2012-11-22 14:01:25	22.89	5.45	137.00	EnProceso	2012-11-27	KLTX0Q6SMKZQZG3P	<a href="#">Ver</a>
16	1	2012-11-22 14:02:13	22.89	5.45	137.00	EnProceso	2012-11-27	R1NWWWFZ582F7X9EN5	<a href="#">Ver</a>
17	1	2012-11-22 14:02:45	22.89	5.45	137.00	EnProceso	2012-11-27	YPOQLXGWAU1713U02	<a href="#">Ver</a>

Figura 31. Listado de pedidos de compra

El administrador puede mirar un listado de los pedidos realizados por nuestros clientes, al mismo tiempo también podrá editar el estado y la fecha de entrega del mismo así como mirar el detalle del pedido.

Detalle del pedido 51				
Lista de artículos				
PRODUCTO	CANTIDAD	PRECIO	IVA	TOTAL
Ferrino Alpha 3	2	218.00€	45.78€	263.78€

Figura 32. Detalle pedido cliente

- **Cientes:** Podemos visualizar un listado de las cuentas tanto si están activas como si no lo están, también podemos mirar los datos de cada cuenta (como a quien pertenece junto a sus datos personales).

Cuentas de Usuarios			
Listando cuentas ...			
LOGIN	CONTRASEÑA	ESTADO	ACCIONES
1234@hotmail.com	147	deshabilitado	<a href="#">Ver</a>
alexia@gmail.com	alexia	deshabilitado	<a href="#">Ver</a>
dani@gmail.com	123456	habilitado	<a href="#">Ver</a>
fanny@hotmail.com	123456	deshabilitado	<a href="#">Ver</a>
jilacmon@gmail.com	jilacmon	habilitado	<a href="#">Ver</a>
vefrjb@gmail.com	1254	deshabilitado	<a href="#">Ver</a>

*Administración Tienda Virtual*

Figura 33. Cuentas usuario

- **Monitorización:** en este apartado podemos observar que productos no tienen stock y a que categorías no se les ha asignado productos.
- **Productos stock:** en este apartado visualizamos a los productos que tienen menos de 20 cantidades en stock para así tener controlado su pedido a nuestros proveedores.
- **Ventas:** en ventas observamos correctamente los productos más y menos vendidos de nuestra tienda.



### 6.1.2. PRUEBAS DE COMPATIBILIDAD EN NAVEGADORES

Se han realizado pruebas en los siguientes navegadores:

- Microsoft Internet Explorer 9

En este navegador no se ha encontrado ningún problema de funcionamiento pero sí unos pequeños errores en el posicionamiento de algunos iconos tanto al inicio como dentro de la aplicación.



Figura 34. Ejemplo de la tienda con la visualización en navegador de Explore

- Google Chrome

En este navegador, desde el enfoque visual se ve todo correcto. La única pega, es que la posición del widget de nuevos productos. Por lo demás, funciona correctamente.

## DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE



Figura 35. Ejemplo de la tienda con la visualización en navegador de Chrome

- Mozilla Firefox

Este es el navegador predeterminado para la aplicación. En él se ha desarrollado y diseñado toda la aplicación. Aparece todo correcto



Figura 36. Ejemplo de la tienda con la visualización en navegador de Firefox

## 7. CONCLUSIONES

En este punto del Proyecto, se me pasan muchas conclusiones por la cabeza. La finalización de este proyecto me aporta una gran satisfacción ya que, hace unos meses, cuando este Proyecto solo era una idea en mi cabeza, parecía como algo inalcanzable y difícil de conseguir al mismo tiempo.

Con esta aplicación he aprendido a enfrentarme a un trabajo que podría considerarse como una tarea real en cualquier empresa. He sido capaz de utilizar parte de los conocimientos adquiridos durante toda la carrera y plasmarlos en un trabajo del que me siento orgulloso de cómo ha quedado. Un trabajo que no se aleja mucho de la idea que en un principio tenía en mente.

Además me ha servido para darme cuenta de que tareas triviales que a simple vista parecían sencillas, se han convertido en un quebradero de cabeza y tareas que parecían imposibles han sido mucho más fáciles de llevar a cabo. También decir, que a medida que el Proyecto iba tomando forma, han surgido muchas dudas que al resolverlas me servían de ejemplo para aprender nuevas cosas dentro de la programación Web.

Pese a estar finalizado el proyecto, la motivación que iba naciendo en mi paso a paso de ir descubriendo nuevas posibilidades me hará seguir investigando en nuevas opciones a añadir dentro de la Tienda Virtual.

### 7.1. TRABAJO REALIZADO

La aplicación desarrollada en este proyecto de final de carrera cumple en si con todos sus requisitos.

La aplicación está preparada para realizar todas las funciones expuestas en el capítulo de especificación de requisitos sin problemas, además de que han sido probadas.

### 7.2. VALORACIÓN PERSONAL

En primer lugar, la motivación para elegir este proyecto de final de carrera ha sido la temática del comercio electrónico, ya que es un tema bastante actual y que ahora mismo está en auge. Por ello, para la creación de este proyecto de final de carrera he tenido el aliciente de crear y desarrollar una aplicación que este enfocada a estos temas.

Este proyecto de final de carrera ha sido una experiencia muy enriquecedora a nivel didáctico y sobre todo a nivel profesional y formativo.

Hemos aprendido lenguajes nuevos que nunca se dieron en la carrera como el PHP y java Script. El lenguaje PHP, me ha gustado mucho. He visto que ha sido una tecnología de desarrollo web muy útil y muy fácil de manejar, además de me ha servido para proyectos personales. También me sorprendido la semejanza que tiene con otras tecnologías que nos han enseñado como Java o C#.

La parte del diseño, más bien el CSS,s ha costado un poco más, ya que ha costado la elección de que colores, imágenes, etc, a utilizar, en cambio la parte de desarrollar ha sido más fluida, pero con ayuda de material que se encuentra expuesto por la red esto ha facilitado mi trabajo.

Finalmente, lo que más nos ha gustado, es que conforme iba pasando el tiempo, la aplicación iba cogiendo forma y se iba haciendo realidad.

## 8. REFERENCIAS

La bibliografía utilizada es la siguiente

[1] <http://sofasoutlet.info/>

[2] <http://wowslider.com/>

[3] <http://appserv.softonic.com/>

PHP

[4] <http://php.net/manual/en/index.php>

[5] <http://php.net/manual/es/book.mysql.php>

[6] <http://www.desarrolloweb.com/articulos/conectar-con-paypal.html>

Html

[7] <http://www.w3schools.com/html/default.asp>

CSS

[8] [http://www.araudi.net/ejemplos/menus\\_desplegables\\_CSS.html](http://www.araudi.net/ejemplos/menus_desplegables_CSS.html)

[9] <http://vagabundia.blogspot.com/2010/09/menu-vertical-desplegable-solo-con-css.html>

PAYPAL

[10] <https://www.sandbox.paypal.com/cgi-bin/webscr>

[11] <https://www.paypal.com/es/webapps/mpp/home>

Creación de carrito de compras con PayPal:

[12] <http://www.colordeu.es/BLOG/carrito-de-compra-en-php-mysql-y-forma-de-pago-paypal-parte-6-anadir-paypal-como-metodo-de-pago-a-nuestro-carro-de-compra>

## **Anexo: manual de usuario**

### Tabla de contenido

<b>9. Manual de usuario</b> .....	70
9.1. Descripción de la base de datos .....	70
9.1.1. Creación de la base de datos y comprobación de la conexión .....	70
9.1.2. Creación de las tablas de la base de datos.....	75
9.1.3. Acceso a la web .....	90

### Tabla de esquemas

<b>Esquema 1. Creación de base de datos</b> .....	70
<b>Esquema 2. Comprobar conexión (visualización)</b> .....	72
<b>Esquema 3. Conexión para tablas</b> .....	73
<b>Esquema 4. Creación e inserción de tablas</b> .....	76

### Tabla de imágenes

<b>Figura 1. Creación correcta</b> .....	70
<b>Figura 2. Conexión ok</b> .....	72

## 9. Anexo: Manual de usuario

### 9.1 – DESCRIPCIÓN DE LA BASE DE DATOS:

#### 9.1.1. CREACIÓN DE LA BASE DE DATOS Y COMPROBACIÓN DE LA CONEXIÓN

Para el almacenamiento de los datos de esta aplicación se ha creado una base de datos llamada “TiendaVirtual”, en la que se ha almacenado todos los datos del proyecto

Existe un fichero un fichero “CrearBD.txt” con el que creamos nuestra base de datos TiendaVirtual:

```
<?php

print "<br><b><u> Creación de una base de datos con MYSQL </b></br>";

//creamos la conexión
$con = mysql_pconnect("localhost", "root", "jill")

or die ("No se puedo conectar");

// creamos la base de datos

$sql = 'CREATE DATABASE TiendaVirtual';

if(mysql_query($sql, $con))
{
echo "se creó la base de datos Tienda Virutal \n";
}
else {
echo "Error al crear la base de datos Base: ".mysql_error(). "\n";
}
?>
```

Esquema 1 Creación de base de datos



Este es mensaje que nos aparece al crear la base de datos:

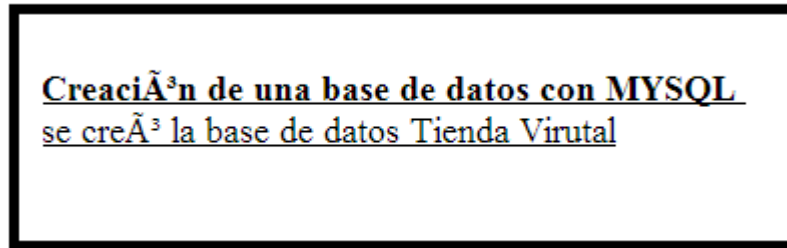


Figura 1. Creación correcta

También existe un fichero que comprueba la conexión de la base de datos:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin tem-
plate="/Templates/ConexionServidor.dwt.php" codeOutsideHTMLOIsLocked="false" -->

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<!-- InstanceBeginEditable name="doctitle" -->

<title>Tienda Virtual</title>

<!-- InstanceEndEditable -->

<!-- InstanceBeginEditable name="head" -->

<!-- InstanceEndEditable -->

<link href="CSS/principal.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="container">

<div class="header"><div class="headerinterior">
```

```
</div></div>

</div>
</br>
<div class="subcontenedor">

<div class="content">

<h1><!-- InstanceBeginEditable name="EditRegion3" -->CConectando a la base de da-
tos<!-- InstanceEndEditable --></h1>

<!-- InstanceBeginEditable name="contenido" -->

<?php

print "<br><b><u> Seleccionamos una base de datos con MYSQL </b></br>";

<?php

print "<br><b><u> Seleccionamos una base de datos con MYSQL </b></br>";

# FileName="Connection_php_mysql.htm"
# Type="MYSQL"
# HTTP="true"
$hostname_conexion = "localhost";
$database_conexion = "TiendaVirtual";
$username_conexion = "root";
$password_conexion = "jill";

$conexion = mysql_pconnect($hostname_conexion, $username_conexion, $pass-
word_conexion) or trigger_error(mysql_error(),E_USER_ERROR);
```

```
echo 'La conexión a la base de datos TiendaVirtual se ha establecido';

?>
        <!-- InstanceEndEditable -->
    </div>
</div>

<div class="footer">
    <center>
        <address>
            <small>Tienda de Virtual ByOnline.</small>
            <br>
            <small>Hora: <? echo date("H:i:s"); ?></small>
            <small>Fecha: <? echo date("j/m/Y"); ?></small>
            <br>
            <small>Webmaster: <a
href="mailto:ByOnline@TiendaVirtual.com">ByOnline@TiendaVirtual.com</a></small>
        </address>
    </center>
</div>
</div>
</body>
<!-- InstanceEnd --></html>
```

Esquema 2. Comprobar conexión (visualización)

Ahora podemos observar que la conexión se establece correctamente



Figura 2. Conexión ok

Pero utilizamos el ficher “conexión.php” del directorio Connections con una conexión mas practica para nuestras tablas.

```
<?php if (!isset($_SESSION)) {  
    session_start();  
}>  
  
<?php  
# FileName="Connection_php_mysql.htm"  
# Type="MYSQL"  
# HTTP="true"  
$hostname_conexion = "localhost";  
$database_conexion = "TiendaVirtual";  
$username_conexion = "root";  
$password_conexion = "jill";  
  
$conexion = mysql_pconnect($hostname_conexion, $username_conexion, $password_conexion) or trigger_error(mysql_error(),E_USER_ERROR);
```

```
//echo $conexion. "Conexion establecida a base de datos ".$database_conexion;  
?>
```

Esquema 3. Conexión para tablas

### 9.1.2. CREACIÓN DE LAS TABLAS DE LA BASE DE DATOS

La creación de las tablas fue a través de un fichero llamado crearTablasBD.txt el cual fue importado a phpMyAdmin. En el eliminamos las tablas existentes, creamos las nuevas tablas y les insertamos sus atributos.

No se encuentran todos los atributos de las tablas porque muchas de ellas han sido insertadas a través de la tienda.

Esta base de datos cuenta con las siguientes tablas:

- ✓ Admin: Almacena información de los administradores de la tienda.
- ✓ Categorías: Mantiene información de todas las categorías existentes.
- ✓ Cesta: En esta tabla tenemos almacenados los productos que se encuentran en el carrito de compras, también guardamos una relación con carritoRaiz.
- ✓ Cestaraiz: contiene la relación del carrito con el usuario.
- ✓ Clientes. Esta tabla guarda información de los clientes datos de alta en el sistema.
- ✓ Compra: En esta tabla creamos el pedido del cliente, así que guardamos su identificador.
- ✓ Cuentas: Almacenamos las cuentas tanto activas como inactivas de nuestra tienda.
- ✓ Impuesto: Se creó esta tabla para facilitarnos la modificación del impuesto por si este cambiase
- ✓ Itemcompra: Almacenamos los productos relacionados a la compra del cliente
- ✓ Itempedidoproveedor: Guardamos los productos que van a ser pedidos a los proveedores de nuestra tienda
- ✓ Pedidoproveedor: Almacena información de los pedidos realizados a nuestros proveedores.
- ✓ Productos: Almacenamos los productos de nuestra tienda

## DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE

- ✓ Proveedores: Mantenemos información de los proveedores de la tienda

Podemos observar los siguientes datos de la creación:

```
DROP TABLE IF EXISTS `admin`;
CREATE TABLE `admin` (
  `id_login` varchar(50) NOT NULL,
  `contraseña` varchar(50) NOT NULL,
  `perfil` varchar(50) NOT NULL,
  PRIMARY KEY (`id_login`),
  UNIQUE KEY `perfil` (`perfil`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `admin`
--

INSERT INTO `admin` VALUES ('admin', 'admin', 'Administrador');

-----

--
-- Estructura de tabla para la tabla `categorias`
--

DROP TABLE IF EXISTS `categorias`;
CREATE TABLE `categorias` (
```

```
`idCategoria` int(5) unsigned NOT NULL auto_increment,
`Nombre` varchar(250) NOT NULL,
PRIMARY KEY (`idCategoria`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=11;

--
-- Volcar la base de datos para la tabla `categorias`
--

INSERT INTO `categorias` VALUES (1, 'Deporte');
INSERT INTO `categorias` VALUES (2, 'VideoGame');
INSERT INTO `categorias` VALUES (3, 'Hogar');
INSERT INTO `categorias` VALUES (4, 'Informatica');

-----

--
-- Estructura de tabla para la tabla `cesta`
--

DROP TABLE IF EXISTS `cesta`;
CREATE TABLE `cesta` (
  `CestaID` int(5) unsigned NOT NULL auto_increment,
  `idProducto` int(4) NOT NULL,
  `cestaRAIZ` int(5) NOT NULL,
  `cantidad` int(5) NOT NULL default '1',
  `precio` decimal(12,2) NOT NULL,
  PRIMARY KEY (`CestaID`),
  KEY `idProducto` (`idProducto`),
```

```
KEY `cestaRAIZ` (`cestaRAIZ`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;

--
-- Volcar la base de datos para la tabla `cesta`
--
-----

--
-- Estructura de tabla para la tabla `cestaraiz`
--

DROP TABLE IF EXISTS `cestaraiz`;
CREATE TABLE `cestaraiz` (
  `ID` int(5) unsigned NOT NULL auto_increment,
  `ClienteID` int(5) NOT NULL,
  `fechaCreacion` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
  PRIMARY KEY (`ID`),
  KEY `ClienteID` (`ClienteID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;

--
-- Volcar la base de datos para la tabla `cestaraiz`
--
-----
```



```
--  
-- Estructura de tabla para la tabla `clientes`  
--  
  
DROP TABLE IF EXISTS `clientes`;  
CREATE TABLE `clientes` (  
  `ClienteID` int(5) unsigned NOT NULL auto_increment,  
  `Nombre` varchar(20) NOT NULL,  
  `Apellidos` varchar(25) NOT NULL,  
  `EMail` varchar(50) NOT NULL,  
  `PalabraPaso` varchar(16) default NULL,  
  `Direccion` varchar(50) NOT NULL,  
  `Provincia` varchar(50) default NULL,  
  `CodigoPostal` varchar(10) default NULL,  
  `Telefono` varchar(20) default NULL,  
  PRIMARY KEY (`ClienteID`),  
  KEY `EMail` (`EMail`,`PalabraPaso`),  
  KEY `PalabraPaso` (`PalabraPaso`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=7 ;  
  
--  
-- Volcar la base de datos para la tabla `clientes`  
--  
  
INSERT INTO `clientes` VALUES (1, 'jill', 'Acosta', 'jilacmon@gmail.com', 'jilacmon',  
'carretera', 'valencia', '46100', '96451874511');  
  
INSERT INTO `clientes` VALUES (2, 'daniel', 'vilar', 'dani@gmail.com', '123456', '18cev',  
'venrverb', '62329', '69856289652');
```

```
--  
-- Estructura de tabla para la tabla `compra`  
--  
  
DROP TABLE IF EXISTS `compra`;  
CREATE TABLE `compra` (  
  `PedidoID` mediumint(8) unsigned NOT NULL auto_increment,  
  `ClienteID` smallint(5) unsigned NOT NULL,  
  `FechaPedido` timestamp NULL default NULL,  
  `MedioPago` varchar(100) default NULL,  
  `IVA` decimal(12,2) unsigned NOT NULL default '0.00',  
  `GastosEnvio` decimal(12,2) NOT NULL,  
  `TotalCargo` decimal(12,2) unsigned NOT NULL default '0.00',  
  `Estado` varchar(100) default NULL,  
  `FechaEnvio` date default NULL,  
  `RefEnvio` varchar(255) default NULL,  
  PRIMARY KEY (`PedidoID`),  
  KEY `ClienteID` (`ClienteID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=71;  
  
-----  
  
--  
-- Estructura de tabla para la tabla `cuentas`  
--
```

```
DROP TABLE IF EXISTS `cuentas`;  
  
CREATE TABLE `cuentas` (  
  `LOGIN` varchar(20) NOT NULL,  
  `PalabraPaso` varchar(20) NOT NULL,  
  `estado` int(1) NOT NULL,  
  PRIMARY KEY (`LOGIN`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Volcar la base de datos para la tabla `cuentas`  
--  
  
INSERT INTO `cuentas` VALUES ('dani@gmail.com', '123456', 1);  
INSERT INTO `cuentas` VALUES ('jilacmon@gmail.com', 'jilacmon', 1);  
  
-----  
  
--  
-- Estructura de tabla para la tabla `impuesto`  
--  
  
DROP TABLE IF EXISTS `impuesto`;  
  
CREATE TABLE `impuesto` (  
  `Contador` int(11) unsigned NOT NULL auto_increment,  
  `pais` varchar(50) NOT NULL,  
  `IVA` int(11) NOT NULL,  
  PRIMARY KEY (`Contador`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
```

```
--  
-- Volcar la base de datos para la tabla `impuesto`  
--  
  
INSERT INTO `impuesto` VALUES (1, 'España', 21);  
  
-----  
  
--  
-- Estructura de tabla para la tabla `itemcompra`  
--  
  
DROP TABLE IF EXISTS `itemcompra`;  
CREATE TABLE `itemcompra` (  
  `itemCompra` int(11) unsigned NOT NULL auto_increment,  
  `Compra` int(11) NOT NULL,  
  `idProducto` varchar(50) NOT NULL,  
  `precio` decimal(12,2) NOT NULL,  
  `cantidad` int(11) NOT NULL,  
  `iva` decimal(10,2) NOT NULL,  
  PRIMARY KEY (`itemCompra`),  
  KEY `Compra` (`Compra`,`idProducto`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=48 ;  
  
-----  
  
--
```

```
-- Estructura de tabla para la tabla `itempedidoproveedor`
--
DROP TABLE IF EXISTS `itempedidoproveedor`;
CREATE TABLE `itempedidoproveedor` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `idPedido` int(11) NOT NULL,
  `idProducto` int(11) NOT NULL,
  `cantidad` int(11) NOT NULL,
  `precio` decimal(12,2) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `idProducto` (`idProducto`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=11 ;

--
-- Volcar la base de datos para la tabla `itempedidoproveedor`
--

INSERT INTO `itempedidoproveedor` VALUES (1, 1, 2, 5, 89.50);
INSERT INTO `itempedidoproveedor` VALUES (2, 1, 1, 7, 136.57);
INSERT INTO `itempedidoproveedor` VALUES (3, 2, 9, 10, 619.90);
INSERT INTO `itempedidoproveedor` VALUES (4, 2, 11, 3, 614.97);
INSERT INTO `itempedidoproveedor` VALUES (5, 3, 5, 2, 125.80);
INSERT INTO `itempedidoproveedor` VALUES (6, 3, 5, 2, 125.80);
INSERT INTO `itempedidoproveedor` VALUES (7, 3, 5, 2, 125.80);
INSERT INTO `itempedidoproveedor` VALUES (8, 3, 5, 2, 125.80);
INSERT INTO `itempedidoproveedor` VALUES (9, 3, 5, 2, 125.80);
INSERT INTO `itempedidoproveedor` VALUES (10, 3, 5, 2, 125.80);
```

```
--  
-- Estructura de tabla para la tabla `pedidoproveedor`  
--  
  
DROP TABLE IF EXISTS `pedidoproveedor`;  
CREATE TABLE `pedidoproveedor` (  
  `ID` int(10) unsigned NOT NULL auto_increment,  
  `idProveedor` int(11) NOT NULL,  
  `precio` decimal(12,2) NOT NULL,  
  `fecha` date NOT NULL,  
  `estado` varchar(50) NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `idProveedor` (`idProveedor`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=4;  
  
--  
-- Volcar la base de datos para la tabla `pedidoproveedor`  
--  
  
INSERT INTO `pedidoproveedor` VALUES (1, 3, 226.07, '2012-12-03', 'enEspera');  
INSERT INTO `pedidoproveedor` VALUES (2, 2, 1234.00, '2012-12-03', 'enEspera');  
INSERT INTO `pedidoproveedor` VALUES (3, 1, 16.00, '0000-00-00', 'completado');  
  
-----  
  
--  
-- Estructura de tabla para la tabla `productos`  
--
```

```
DROP TABLE IF EXISTS `productos`;

CREATE TABLE `productos` (
  `idProducto` int(5) unsigned NOT NULL auto_increment,
  `idCategoria` int(5) NOT NULL,
  `idProveedor` int(5) NOT NULL,
  `Nombre` varchar(250) NOT NULL,
  `precio` decimal(15,2) default NULL,
  `descripGrande` varchar(200) default NULL,
  `descripPequena` varchar(200) default NULL,
  `IVA` decimal(15,2) default NULL,
  `stock` int(250) default NULL,
  `imagen` varchar(10000) default NULL,
  `fecha` datetime NOT NULL default '0000-00-00 00:00:00',
  `etiqueta` varchar(50) NOT NULL,
  PRIMARY KEY (`idProducto`),
  KEY `idProveedor` (`idProveedor`),
  KEY `idCategoria` (`idCategoria`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=20 ;

--
-- Volcar la base de datos para la tabla `productos`
--

INSERT INTO `productos` VALUES (1, 4, 3, 'TS-L462 IDE CD-RW', 19.51, 'Aplicaci', 'SE PRO-
VEE: 1 Disco combinada de X TS-L462 CD-RW DVD-ROM IDE', 3.51, 140, 'TS-L462 IDE CD-
RW.JPG', '2012-11-03 00:00:00', 'disco');

INSERT INTO `productos` VALUES (2, 4, 3, 'Cartucho hp 300 negro hp deskjet d2560 y
f4280', 17.90, 'Descripci', 'Cartucho de impresi', 3.22, 108, 'cartuchoHP300.jpg', '2012-11-03
00:00:00', 'impresora');

INSERT INTO `productos` VALUES (3, 4, 3, 'hp 301 combo tricolor y negro', 24.30, 'Color
de impresi', 'Los cartuchos del paquete combinado de cartuchos de tinta tricolor y negra
HP 301 est', 4.37, 212, 'cartuchoHP301.jpg', '2012-11-03 00:00:00', 'impresora');
```

## DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE E-COMMERCE

```
INSERT INTO `productos` VALUES (4, 4, 3, 'Disco Seagate', 100.00, 'Especificaciones\r\n
N', 'Unidad de disco duro externa Expansion de Seagate de 2 TB ',
18.00, 94, 'disco-Seagate3.5.jpg', '2012-11-03 00:00:00', 'ordenadores');

INSERT INTO `productos` VALUES (5, 1, 1, 'Maillot Saxo Bank Tinkoff Bank BodyFit Pro
Team Jersey', 62.90, 'Maillot Saxo Bank Tinkoff Bank BodyFit Pro Team Jersey con un
ajuste anat', 'Camiseta de deporte ', 11.32, 92, 'SF-04804001-201.jpg', '2012-11-03
00:00:00', 'viento');

INSERT INTO `productos` VALUES (6, 1, 1, 'Ferrino Alpha 3', 109.00, 'Tejidos\r\n
', 'Tienda 3 plazas compacta muy vers', 19.62, 100, 'ALPHA_3.jpg', '2012-11-03
00:00:00', 'campo');

INSERT INTO `productos` VALUES (7, 1, 1, 'Trangoworld Zimal', 289.00, 'Descripci', '
Magn', 52.02, 94, 'Chaqueta-ZIMAL_541_s.jpg', '2012-11-03 00:00:00', 'footing');

INSERT INTO `productos` VALUES (8, 1, 1, 'Petzl Alcanadre', 104.00, 'M', 'Saco com', 18.72,
84, 'Petzl_Alcanadre_k2_s.jpg', '2012-11-03 00:00:00', '');

INSERT INTO `productos` VALUES (9, 2, 2, 'Borderlands 2', 61.99, 'Borderlands 2, secuela
del aclamado juego de rol y acci', 'Juego XBOX300\r\n
Genero: Accion/Aventura',
18.00, 94, '2med.jpg', '2012-11-03 00:00:00', 'consola');

INSERT INTO `productos` VALUES (10, 2, 2, 'PlayStation 3, 320GB + Dancestar Party +
Move', 279.98, 'PlayStation 3, 320GB + Dancestar Party + Move\r\n
', 'Juego
PlayStation3\r\n
Genero: Consola', 50.40, 84, 'play3Move.jpg', '2012-11-03 00:00:00',
'consola');

INSERT INTO `productos` VALUES (11, 2, 2, 'Nintendo 3DS XL, color Azul y Negro', 204.99,
'', 'Juego 3DS-DS\r\n
Genero: Consola', 36.90, 76, '3DS-Ds.jpg', '2012-10-09
00:00:00', 'consola');

INSERT INTO `productos` VALUES (12, 3, 4, 'Lavavajillas Beko DFN1403', 224.14, 'Caracter',
'Lavavajillas Beko DFN1403. Clase A Blanco.', 40.34, 78, 'DFN1403.png', '2012-11-03
00:00:00', 'cocina');

INSERT INTO `productos` VALUES (13, 3, 4, 'Barbacoa para mesa Brisbane', 71.50, 'Carac-
ter', 'La barbacoa de mesa Brisbane es la barbacoa ideal para el verano', 12.87, 84,
'tabbbq-base.jpg', '2012-11-03 00:00:00', 'campo');

INSERT INTO `productos` VALUES (14, 3, 4, 'Enfriador de vino Cooper Cooler', 99.00,
'Caracter', 'Ideal para fiestas o eventos donde se necesiten grandes cantidades de bebi-
das fr', 17.82, 84, 'cooler-4.jpg', '2012-12-03 00:00:00', 'campo');

INSERT INTO `productos` VALUES (15, 5, 5, 'GIBSON ES335 PLAIN CHERRY (ESDPCHNH1)',
2149.00, '
Pastillas 57 classics\r\n
Clavijero Grover\r\n
Puente tune-o-matic\r\n
Incluye estuche y certificado de autentici-
dad.\r\n
Cuerpo de arce\r\n
M', 'Quiz', 386.82, 75, 'Articu-
lo0002583.png', '2012-12-03 00:00:00', 'instrumento');

INSERT INTO `productos` VALUES (16, 5, 5, 'YAMAHA YAS-280 SAXO ALTO', 849.00, 'El
YAS280 es el sustituto del gran cl', 'El YAS280 es el sustituto del gran cl', 152.82, 84,
'Articulo0004757.png', '2012-12-03 00:00:00', 'instrumento');

INSERT INTO `productos` VALUES (17, 5, 5, 'Atlantic IV 120 Negro', 2785.00, 'ACORDEON
```



```
DE PIANO CROMATICO Nueva reedici', 'CORDEON ATLANTIC IV 120M NEG', 501.00, 84,
'Acordeon120-Negro.jpg', '2012-12-03 00:00:00', '');

INSERT INTO `productos` VALUES (18, 5, 5, 'LD Systems DEEP2 DP1600', 359.00, ' Model
Name: DP1600\r\n          ', 'Amplificador de Potencia 2x800W 2 OHM Dinamico y
eficiente. ', 64.62, 89, 'amplificador-dp600.jpg', '2012-12-03 00:00:00', '');

-----

--
-- Estructura de tabla para la tabla `proveedores`
--

DROP TABLE IF EXISTS `proveedores`;
CREATE TABLE `proveedores` (
  `idProveedor` int(11) NOT NULL auto_increment,
  `Empresa` varchar(200) NOT NULL,
  `Direccion` varchar(200) NOT NULL,
  `email` varchar(200) NOT NULL,
  `telefono` varchar(50) NOT NULL,
  PRIMARY KEY (`idProveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;

--
-- Volcar la base de datos para la tabla `proveedores`
--

INSERT INTO `proveedores` VALUES (1, 'Herrera Compusistem', 'Doblas 1578S', 'Herre-
ra@Compusistem.com', '964585412');

INSERT INTO `proveedores` VALUES (2, 'InfoHouse', 'Av. Rivadavia 3558', 'Servi-
cios@infoCad.com', '964785125');

INSERT INTO `proveedores` VALUES (3, 'Music Master', 'Av. La Plata 365', 'distribu-
cion@musicmaster.com', '948512364');

INSERT INTO `proveedores` VALUES (4, 'Joan', 'Salas Marcos', 'salas@distrisur.com',
```

```
'948512457');  
INSERT INTO `proveedores` VALUES (5, 'Enrique', 'Castelli Parera',  
'castelli@districentro.com', '9854712548');
```

Esquema 4. Creación e inserción de tablas

### 9.1.3. ACCESO A LA WEB

El proyecto ha sido subido a un servidor online (<http://ooowebhost.com/>), el cual puede ser consulta a través del url: <http://byonline.host22.com/>



## **10. Anexo: Crear cuenta paypal de prueba**

## 10. Crear una cuenta de pruebas en Paypal

Para realizar las pruebas de nuestro script podríamos usar nuestras propias cuentas de PayPal pero lo cierto es que no os lo aconsejo, a no ser que queráis perder dinero xD.

Para hacer **pruebas con api de Paypal** antes de lanzar nuestra tienda virtual al público existe una plataforma llamada [Paypal Sandbox](#) para desarrolladores con la que podremos hacer pruebas con cuentas ficticias.

Así que lo primero que haremos es registrarnos, donde nos pedirán algunos datos personales como Nombre, Apellidos, email y password (recomiendo no usar vuestros propios nombres).



The screenshot shows the PayPal Sandbox website interface. At the top left is the 'PayPal Sandbox' logo. Below it is a 'Member Log In' section with input fields for 'Email Address' and 'Password', a 'Keep me logged in' checkbox, a 'Log In' button, and a 'Trouble Logging In?' link. To the right is a banner with the text 'PayPal Sandbox Put your code to the test'. Below the banner is a section titled 'Need an account? Sign up now to access PayPal Environment.' with a 'Sign Up Now' button. A red arrow points to the 'Sign Up Now' button. Below the banner is a section titled 'Additional resources' with links for 'Documentation', 'PayPal Developer Network', and 'Customer Support'. Below this is a text block: 'Use the Sandbox Test Environment to create and manage test accounts and API credentials. You can also access valuable developer resources.' and another 'Sign Up Now' button.

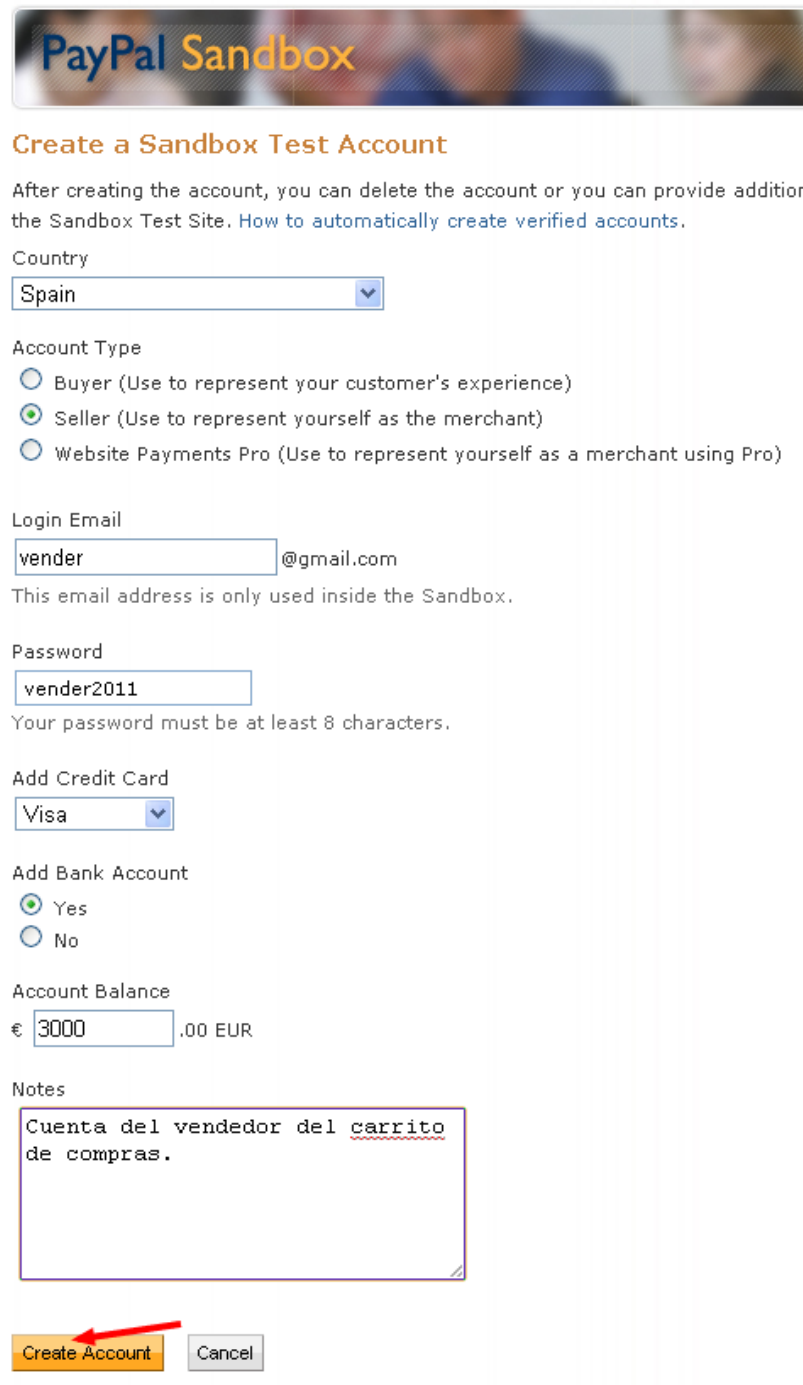
Una vez tenemos registrado nuestro usuario de pruebas de Paypal es hora de crear las 2 partes "humanas" que interactúan en una tienda virtual, el comprador y el vendedor, para así tener constancia de que se efectúa la compra y que el vendedor recibe el pago.

### CREAR CUENTA DE PRUEBAS PAYPAL SANDBOX PARA VENDER

Una vez hemos activado nuestro usuario de Paypal Sandbox a través del e-mail y nos hemos logueado correctamente, entramos en el panel de administración.

Para crear una cuenta Paypal SandBox para vender con datos que ellos mismo nos proporcionan (es lo más fácil ya que no tenemos que validar la cuenta), seguimos los siguientes pasos:

1. Ir a Test Accounts
2. Clic en Preconfigured y rellenar los datos que nos piden tal y como veis en la siguiente imagen.



**PayPal Sandbox**

### Create a Sandbox Test Account

After creating the account, you can delete the account or you can provide additional information to the Sandbox Test Site. [How to automatically create verified accounts.](#)

Country  
Spain

Account Type

- Buyer (Use to represent your customer's experience)
- Seller (Use to represent yourself as the merchant)
- Website Payments Pro (Use to represent yourself as a merchant using Pro)

Login Email  
vender@gmail.com  
This email address is only used inside the Sandbox.

Password  
vender2011  
Your password must be at least 8 characters.

Add Credit Card  
Visa

Add Bank Account

- Yes
- No

Account Balance  
€ 3000.00 EUR

Notes  
Cuenta del vendedor del carrito de compras.

Create Account Cancel

El mail que se va a utilizar aquí, tal y como indican en el formulario, es una dirección simulada, así que no hace falta registraros en gmail

#### CREAR CUENTA DE PRUEBAS PAYPAL SANDBOX PARA COMPRAR

El sistema es muy parecido, solo cambia en el tipo de cuenta (Account Type) que lo seleccionaremos como **Buyer** en lugar de Seller.

**PayPal Sandbox**

### Create a Sandbox Test Account

After creating the account, you can delete the account or you can provide additional information to the Sandbox Test Site. [How to automatically create verified accounts.](#)

Country  
Spain

Account Type  
 Buyer (Use to represent your customer's experience)  
 Seller (Use to represent yourself as the merchant)  
 Website Payments Pro (Use to represent yourself as a merchant using Pro)

Login Email  
compra@gmail.com  
This email address is only used inside the Sandbox.

Password  
comprar2011  
Your password must be at least 8 characters.

Add Credit Card  
Visa

Add Bank Account  
 Yes  
 No

Account Balance  
€ 4500.00 EUR

Notes  
Cuenta para comprar en el carrito de compras

Create Account Cancel

Ahora si que estamos preparados para hacer pruebas comprando en nuestra tienda virtual.

