



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Portal web para una tienda online de horticultura

Proyecto Final de Carrera
Ingeniería Superior de Informática

Autor: Enrique Diego Oviedo

Director: Sergio Sáez Barona

16/09/2012

Tabla de contenidos

Contenido

1. Introduccion.....	6
Motivación.....	6
2. ESPECIFICACION DE REQUISITOS.....	8
2.1. Introducción	8
2.1.1. Propósito	8
2.1.2. Ámbito del Sistema	8
2.1.3. Definiciones, Acrónimos y Abreviaturas.	8
2.1.4 Referencias.	10
2.1.5. Visión global	10
2.2. Descripción general	11
2.2.1 Perspectiva del producto.....	11
2.2.2 Funciones del producto.....	11
2.2.3 Características de los usuarios.	11
2.2.4 Restricciones.....	12
2.2.5 Supuestos y dependencias.....	12
2.2.6 Requisitos futuros.	12
2.3. Requisitos específicos.	13
2.3.1 Requisitos de interfaces externas	13
2.3.2 Requisitos funcionales.....	13
2.3.3 Requisitos de Rendimiento	15
2.3.4 Restricciones de Diseño.....	15
2.3.5 Requisitos hardware	16
2.3.6 Atributos del sistema	16
2.3.7 Otros requisitos	17
3 Análisis	18
3.1 Diagrama de clases.	18
3.2 Diagrama de casos de uso.	19
3.3 Plantillas de casos de uso.....	20
3.3.1 Consultar categoría.....	20
3.3.2 Consultar producto	20
3.3.3 Búsqueda producto.....	21



3.3.4 Alta usuario	21
3.3.5 Contacto.....	22
3.3.6 Añadir al carro	23
3.3.7 Quitar del carro	23
3.3.8 Realizar compra	23
3.3.9 Modificar datos de usuario.....	24
3.3.10 Visualizar historial de pedidos	24
3.3.1 Cambiar contraseña	25
3.3.12 Cerrar sesión.....	26
3.3.13 Recuperar contraseña	26
3.3.14 Visualizar un pedido.....	27
3.4 Diagramas de secuencia	27
3.4.1 Añadir al carrito.	27
3.4.2 Listar el historial de los pedidos de un usuario	28
3.4.3 Registro de usuario nuevo.	28
4. Diseño	29
4.1. Arquitectura del Sistema.....	29
4.2 Nivel de interfaz.....	30
4.3 Nivel de negocio.	32
4. 4 Capa de persistencia.	32
4.4.1 Diseño lógico.	33
5. Implementación e integración	34
5.1. Tecnologías	34
5.1.1 MySQL.....	34
5.1.2 PHP	34
5.1.3 CSS:	35
5.1.4 AJAX:	35
5.1.5 HTML:.....	35
5.1.6 Arquitectura Cliente Servidor:.....	36
5.2. Herramientas.....	36
5.2.1 WAMP	36
5.2.2 PhpMyAdmin	36
5.2.3 Notepad++.....	36
5.3. Detalles de la implementación	37

5.3.1 Perfiles de usuario.....	37
5.3.2 Autenticación de usuarios.	37
5.3.3 Selección fichero PHP.....	38
5.3.4 Inserción de HTML en el DOM	38
5.4 Diagrama de navegabilidad.	39
6. Evaluación y pruebas.....	40
6.1 Evaluación.....	40
6.2 Pruebas.	40
6.2.1 Enlaces rotos.....	40
6.2.2. Pruebas de Validación HTML y CSS.....	41
6.2.2. Pruebas de compatibilidad en navegadores	42
6.2.3. Pruebas de compatibilidad de resolución	43
6.2.4 Casos de prueba.....	44
7. Conclusiones	49
8. Bibliografía.....	50



1. Introduccion

La realización de este proyecto llevara a cabo una tienda virtual de venta de productos hortícolas, donde los distintos usuarios, tanto registrados como no registrados podrán ver el catalogo de nuestros productos.

El catalogo de los productos lo podremos navegar o bien seleccionado el tipo de producto que queremos ver mediante los menús o bien utilizando la función de búsqueda.

Los productos tendrán una breve descripción de la especie y variedad de la que se esta hablando así como las características especiales de la misma.

Los usuarios de nuestra página podrán ir añadiendo productos a un carrito que al final podrán comprar pagando el importe mediante tarjeta de crédito. Los usuarios que no estén registrados, si quieren comprar lo que hayan depositado en el carrito tendrán que registrarse.

Un usuario registrado puede loguearse y entrar en su historial de pedidos así como para ver cómo van las compras que aun no han finalizado.

Los usuarios que quieran, podrán participar en un foro que pondremos a disposición de ellos para que formulen preguntas o den consejos a otros compradores de cómo tienen que cultivar sus variedades o que hacer ante una determinada plaga.

Los objetivos concretos consistirán en:

- Ofrecer un catalogo de productos a nuestros clientes así como la navegación y el filtrado de los productos.
- Permitir poder comprar los productos que se encuentran en nuestra página.
- Facilitar el mantenimiento de dicho catálogo.

Los resultados y productos que se piensan obtener:

- Una tienda virtual on-line.
- Una aplicación para el control de usuarios, gestión de productos y pedidos, así como el mantenimiento del catalogo.

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

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

La codificación de un prototipo de la aplicación.

- La realización de pruebas técnicas del sitio Web.

Los recursos disponibles consistirán en:

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

Motivación

La motivación para realizar este proyecto es el hecho de poder tener conocimiento sobre un campo que me puede dar trabajo yendo por libre sin tener que depender de una empresa.

Gracias a la realización de este tipo de proyecto me adentrare en un campo en el que cualquier persona desde su casa puede ver y en el caso de estar interesada comprar.

Con comercio online puedo montar una tienda sin ningún coste que no sea el del material que voy a vender.

Cada vez más personas montan su propia tienda on-line, con lo que saber cómo hacer una tienda on-line podrá llevarme a construir tiendas para otras personas o, una vez controlado el php, mysql y ajax, poder crear otro tipo de aplicaciones web.

Al realizar una tienda on-line entramos en la importantísima tarea de la gestión de usuarios y visibilidad de los mismos, por esto nuestra aplicación debe tener diferentes perfiles, eso es lo que diferencia el control sobre bases de datos y perfiles y lo que da calidad a una página web, el hacer que una persona se sienta especial por poder acceder a una parte de la aplicación que nadie mas puede acceder.

En resumen poder tener conocimientos de la creación de aplicaciones web, siempre te servirá para poder desenvolverte en la vida.

2. ESPECIFICACION DE REQUISITOS

2.1. Introducción

En esta sección explicaremos cual es la meta del proyecto, como lo vamos a hacer y las características y funcionalidades del mismo.

2.1.1. Propósito

El propósito de la ESR (especificación de requisitos) va dirigida para poder tener un conocimiento inicial global de la aplicación que vamos a abordar, y dando así una guía de trabajo sobre las características y condiciones de debe tener y cumplir nuestra aplicación. Esto va dirigido a cualquier persona que esté interesada en la aplicación web ya que se hablara en un lenguaje lo menos técnico posible (sin pérdida de información claro está) para poder llegar al máximo de personas.

2.1.2. Ámbito del Sistema

Nuestro futuro sistema será una tienda virtual on-line de venta de productos hortícolas.

El sistema permitirá la búsqueda de productos así como su compra. Se permitirá el registro de usuarios, así como visualizar el historial de los pedidos pasados y el seguimiento de los pedidos presentes. El sistema dará como soporte un foro donde los usuarios podrán exponer sus dudas, preguntas o afirmaciones.

Gracias al sistema una sola persona podrá gestionar la tienda virtual dedicándole muy pocas horas al día una vez ya este en producción, con el objetivo de poder exponer a la clientela nuestros productos de la forma mas clara posible y venderlas de la manera mas simple que tenemos ahora en el mercado de las tiendas on-line.

Con todo esto tendremos una aplicación de venta de productos on-line de última generación totalmente funcional y al alcance de cualquier persona que tenga internet.

2.1.3. Definiciones, Acrónimos y Abreviaturas.

2.1.3.1 Definiciones

Tienda online: Se refiere a un comercio convencional que usa como medio principal para realizar sus transacciones un sitio web de Internet.

Sitio web: Un sitio web es una colección de páginas web relacionadas y comunes a un dominio de Internet o subdominio en la www.

Dominio de Internet: Un dominio de Internet es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet.

Foro: es una aplicación web que da soporte a discusiones u opiniones en línea, permitiendo al usuario poder expresar su idea o comentario respecto al tema tratado.

Contenido dinámico: es aquél que se genera automáticamente en el momento que alguien solicita su visualización, por tanto, puede cambiar dependiendo de quién lo solicite o en que momento lo haga.

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Usuario anónimo: es un individuo que visitara nuestra página y que no se logeera contra el sistema.

Usuario registrado: Es un usuario que visitara nuestra página y que se logueara contra el sistema para acceder a una serie de características de la aplicación que no se podrían acceder

Loguearse: cuando un usuario pone su loguin y contraseña y se comprueba en la base de datos que existe ese usuario dándole paso a una parte privada de la aplicación.

Loguin: Es el nombre con el que se identifica a un usuario, que con anterioridad ha realizado un proceso de registro, suele formarse con un código alfanumérico de varios dígitos.

Dependiendo de la aplicación se le pueden asignar distintos roles y accesos.

Base de datos: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Contraseña: Palabra secreta perteneciente a un nombre de usuario que dan paso a la parte privada de la aplicación de ese usuario.

2.1.3.2. Acrónimos.

PHP: PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor.

MySQL: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

HTML: siglas de *HyperText Markup Language* («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.

CSS: es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML.

WAMP: es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- **Windows**, como sistema operativo;
- **Apache**, como servidor web;
- **MySQL**, como gestor de bases de datos;
- **PHP** (generalmente), **Perl**, o **Python**, como lenguajes de programación.

UML: Lenguaje Unificado de Modelado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

IEEE: en español **Instituto de Ingenieros Eléctricos y Electrónicos**, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

W3C: World WideWeb Consistorium es un consorcio internacional que produce recomendaciones para la World Wide Web.

RAM: La **memoria de acceso aleatorio** (en inglés: *random-access memory*), se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software.

MHZ: equivale a 10^6 hertzios (1 millón). Se utiliza muy frecuentemente como unidad de medida de la frecuencia de trabajo de un dispositivo de hardware, o bien como medida de ondas electromagnéticas en telecomunicaciones.

MB: El **megabyte (MB)** o megaocteto (Mo) es una unidad de medida de cantidad de datos informáticos. Es un múltiplo del byte u octeto, que equivale a 10^6 B (un millón de bytes).

HTTPS: es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hiper Texto, es decir, es la versión segura de HTTP.

2.1.4 Referencias.

Las referencias utilizadas para la realizar la especificación de requisitos son los siguientes:

- <http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- <http://es.wikipedia.org/wiki/Wikipedia:Portada>

2.1.5. Visión global

En la especificación de requisitos describiremos la aplicación, sus restricciones y en que ámbito trabaja. En resumen explicaremos como se debe comportar nuestro programa sobre unas determinadas circunstancias y que funcionalidades tiene que cumplir, para que podamos entender mejor el producto.

2.2. Descripción general

En esta sección vamos a explicar los factores que afectan al producto

2.2.1 Perspectiva del producto

El producto que presentamos es una aplicación que no pertenece a una aplicación mayor. Es una aplicación web que para poder funcionar necesita estar alojada en un servidor web o en un servidor local, ya que la parte del servidor está hecha en php. Además tendrá una base de datos que accederemos a ella mediante MySQL.

Para poder ejecutarla en la parte cliente necesitamos un sistema operativo y un navegador.

2.2.2 Funciones del producto

En esta sección vamos a explicar cuál es la funcionalidad que tiene nuestra aplicación a grandes rasgos.

En primer lugar se podrá visitar el catalogo de la tienda. Ya que es una tienda virtual este es una de las partes más importantes ya que expone los productos que queremos vender.

Tendremos un modulo de búsqueda de productos para que el usuario pueda encontrar más rápidamente el producto o tipo de producto que busca.

Tendrá un sistema para dar de alta a usuarios nuevos que quieran formar parte del sistema de la tienda y quieran hacer compras, así como recibir información.

Una vez que el usuario se ha dado de alta podrá, cada vez que entre a la pagina y se loguee podrá acceder a su cuenta, donde podrá cambiar sus datos, contraseña, acceder al historial de pedidos, así como a los propios pedidos y cerrar la sesión.

En la sección del historial de pedidos el usuario no puede modificar nada. La otra sección es la de la información referente al alta del usuario, lo que vendría a ser la información personal, como por ejemplo nombre, edad, dirección. En esta sección también podrá cambiar sus datos de envió.

El usuario anónimo como el usuario registrado pueden hacer uso del carrito de la compra que es donde van añadiendo productos para luego, si ellos quieren, hacer efectiva la compra. En este carrito pueden añadir y quitar producto en todo momento al estar en uno de los laterales además de visualizar su contenido.

También se dispondrá de un foro donde todos los usuarios podrán visitar y los usuarios registrados en el foro podrán postear.

2.2.3 Características de los usuarios.

Tendremos dos tipos de usuarios. El usuario anónimo y el registrado.

El usuario anónimo es el que menos cosas puede hacer en la aplicación. Es el usuario que entra en nuestra página para visitarla y ver los productos pero que no ha comprado ninguna vez ni se

ha registrado, y si lo ha hecho no esta logueado en el sistema. Podrán visitar las secciones del catalogo, añadir y quitar productos del carrito, contactar con la tienda mediante email, así como entrar en el foro y visitarlo, nunca escribir en el, solo puede ver el contenido del mismo.

El usuario registrado es un usuario que ya forma parte de nuestro sistema, que o bien se ha dado de alta porque ha querido sin comprar nada, o que ha comprado algo y necesariamente cuando hace esta acción, debe darse de alta en el sistema.

Este usuario tiene mas opciones a las que puede acceder como modificar los datos de su cuenta, contraseña y ver el historial de pedidos.

2.2.4 Restricciones.

Para poder ver nuestra página en un ordenador personal, las restricciones son tener acceso a internet y tener un sistema operativo que incluya un navegador web. En otro tiempo estas restricciones hubieran tenido sentido, pero ahora cualquier ordenador personal tiene estas características.

Ya que nuestra aplicación no va a constar de popup, no es necesario que este desactivado el bloqueo de popups. Y como la información entre paginas la guardaremos en sesiones tampoco hará falta que estén activadas las cookies.

Nuestra página va a estar alojada en un servidor web por lo que las restricciones que nos dé el proveedor del servidor de conexión y de atención a peticiones en paralelo serán las que tengamos.

2.2.5 Supuestos y dependencias.

El desarrollo de la aplicación se hará sobre Mozilla Firefox, por lo que el cambio de explorador como Explorer, google Chrome o Safari puede repercutir en el buen funcionamiento de la aplicación web. Por lo demás la parte servidor lo llevara la empresa contratada que aloje nuestra página que debe soportar PHP, MySQL.

El cambio de sistema operativo así como las características del ordenador no influirán en el buen funcionamiento, solo repercutirán en el tiempo de respuesta hacia el usuario.

2.2.6 Requisitos futuros.

En un futuro esta aplicación podría generalizarse y crear un portal web general en el que se podrá vender cualquier tipo de producto. Para ello solo tendrá que haber un administrador que gestione la página con la base de datos.

La pagina en si será la misma, lo único que cambiara son los productos que se venden en ellas así como el estilo pero la capa negocio será la misma.

Gracias a esto podremos comercializar nuestra aplicación como una plantilla y crear nuevos clientes que nos pidan a nosotros el mantenimiento de la página.

2.3. Requisitos específicos.

2.3.1 Requisitos de interfaces externas

Interfaces hardware

Para la parte del cliente nos basta con un ordenador personal con tarjeta de red y un contrato con un operador que nos proporcione internet. Cualquiera de los ordenadores personales de ahora debe soportarlo.

Para la parte servidor solo necesitamos una maquina que nos sirva y aloje nuestra página web con su base de datos.

Interfaces software

Para la realización de este proyecto utilizaremos un sistema operativo Windows 7 Enterprise con la aplicación wamp que contiene un servidor apache 2.2, MySQL X.X como gestor de base de datos y PHPMyAdmin como interfaz para la gestión de la misma. Y Php para la comunicación con el servidor. En la parte cliente se utilizara HTML, CSS y AJAX.

Interfaces de comunicaciones.

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

2.3.2 Requisitos funcionales

Para la especificación de requisitos funcionales lo haremos distinguiendo por tipo de usuario.

Usuario anónimo.

- ***Consulta según categoría.***

El usuario anónimo, al igual que los demás usuarios, al pulsar en la barra lateral izquierda una de las categorías borraremos todos los elementos que había en la pantalla y pondremos la lista de elementos devueltos por la consulta SQL. Si hay mas elementos de los que pueden caber en la pantalla aparecerá una barra de scroll en el lateral derecho de la pantalla.

- ***Consulta de producto***

El usuario anónimo, al igual que los demás usuarios, al pulsar sobre un producto de la lista desaparecerán todos los elementos de la lista y en su lugar aparecerá la foto del elemento así como la descripción del mismo. En la pantalla podrá dar a añadir a la cesta con la cantidad de elementos del producto que quiera comprar.

- ***Búsqueda de producto mediante cadena de caracteres.***

El usuario anónimo, al igual que los demás usuarios, puede poner una cadena de caracteres y se crea una búsqueda en la base de datos que contenga esa palabra que estamos buscando. Si hay coincidencias sacaremos por pantalla todos los elementos que contengan en el nombre o en la descripción del elemento la cadena que estamos

buscando. Si hay mas elementos de los que pueden caber en la pantalla aparecerá una barra de scroll en el lateral derecho de la pantalla.

- **Alta de usuario en el sistema**

El usuario anónimo puede darse de alta bien pinchando en el menú principal “Registrarse” o bien cuando intenta comprar lo que tiene el carrito y no esta loggeado la aplicación le envía a la pantalla de dar de alta nuevo usuario en el sistema. Una vez que ha pinchado en el enlace, aparecerá un formulario con los datos que tiene que rellenar. Los datos a rellenar son los datos personales (nombre, edad, sexo, etc.), así como la dirección de envió.

- **Información de contacto**

El usuario anónimo, al igual que los demás usuarios, puede pinchar en el menú contacto donde aparecerá un formulario de correo donde el usuario podrá rellenar y enviar a la tienda exponiendo sus inquietudes.

- **Añadir al carrito**

El usuario anónimo, al igual que los demás usuarios, puede añadir al carrito de la compra un elemento. Para ello debemos estar primero en la pantalla del producto. Una vez que estamos allí tenemos un botón “Añadir” más un display al lado del botón para la cantidad de elementos que queremos comprar, que nos permite añadir el elemento al carrito con la cantidad que le pongamos.

- **Quitar del carrito**

El usuario anónimo, al igual que los demás usuarios, podemos quitar elementos del carrito que habíamos puesto con anterioridad y que ahora no queremos. Para ello tenemos dos opciones, una de ellas es pinchar sobre la X de la línea que contiene el producto en la barra lateral derecha y otra es entrando en el carrito y pulsar en el icono que es una papelera. Hecho esto el elemento será borrado y el coste de lo que llevamos en el carrito se recalculara.

- **Usuario registrado**

El usuario registrado puede realizar todas acciones que realiza el usuario no registrado mas las siguientes funcionalidades

- **Realizar compra del carrito.**

Una vez que hemos hecho nuestra selección de productos podemos realizar la compra del mismo. Para ello debemos acceder al carrito y allí pulsar en el icono del dólar. Hecho esto aparecerá la factura con un icono en la parte de abajo que pone realizar pedido. Si pulsamos en el botón haremos efectivo el pedido y pasara a formar parte de nuestro historial de pedidos.

- **Cambiar los datos de la cuenta.**

El usuario que esta registrado tiene la opción de cambiar los datos que tiene asociados al sistema, bien porque están erróneos o bien porque han cambiado. El usuario puede pinchar en el enlace del menú “Cuenta”. Una vez allí puede pinchar modificar, así los campos de la cuenta que se puedan cambiar (hay campos que no se pueden cambiar

como por ejemplo el nombre de cuenta con el que se ha dado de alta) se harán modificables, y una vez que haya cambiado los datos de la cuenta deberá pulsar el botón guardar. Si todo esta correcto se modificaran los datos de la cuenta en la BBDD.

- **Acceder al historial de pedidos.**

El usuario registrado puede ver cuál es su historial de pedidos. Para ello debe pinchar en el menú “Cuenta” y una vez allí debe pinchar en el menú historial pedidos. Hecho esto se mandara una consulta SQL a la base de datos de todos los pedidos del usuario. La respuesta del servidor se pondrá en una lista que presentaremos por pantalla. Esta lista solo se puede consultar nunca modificar.

- **Acceder al pedido.**

Una vez que el usuario esta en el historial de pedidos puede visualizar cada uno de los pedidos pulsando en el icono que es una lupa. Una vez hecho esto aparecerá toda la información referente al pedido.

2.3.3 Requisitos de Rendimiento

Al estar diseñando una web donde se realizan compras y consultas contra un servidor, el elemento que puede dar un problema de cuello de botella, que nos repercute a nosotros, es el servidor. Una carga excesiva de usuarios haciendo peticiones al servidor puede llegar a colapsarlo llegando a la cota máxima de rendimiento del servidor y quedándose así usuarios que los cuales vean sus peticiones ralentizadas. Para poder controlar este problema, que no depende directamente de nosotros lo que debemos hacer es hablar con la empresa a la que contratemos el alojamiento de la página y que gestione nuestro servidor. Ellos serán los que nos darán mas capacidad de rendimiento a nuestro servidor.

Para la parte cliente de la aplicación los requisitos para que el funcionamiento sea óptimo, siempre teniendo en cuenta que las condiciones por parte del servidor son óptimas, con los equipos que se tienen en ahora en las casas y con una conexión a internet no inferior a 1 GB la aplicación funciona correctamente.

Ya que no se van a gestionar datos masivos la BBDD no tendrá requisitos de rendimiento.

2.3.4 Restricciones de Diseño

Para la realización de una aplicación web como la que estamos creando tenemos que seguir una serie de estándares que son los que dictaminan la World Wide Web Consistorium (W3C).

Un sitio basado en estándares web mostrará una mayor consistencia visual. Gracias al uso de XHTML para el contenido y CSS para la apariencia, se puede transformar rápidamente un sitio, sin importar que se trate de una página web o miles, realizando cambios en un solo lugar.

Los documentos que separan apariencia de contenido usan menos código, además, CSS permite conseguir efectos que antes requerían el uso de JavaScript e imágenes, por lo que los sitios basados en estándares utilizan menos ancho de banda y se muestran más rápido a los usuarios, mejorando dramáticamente la experiencia de estos.

Los documentos basados en XHTML válido son más relevantes para los motores de búsqueda, contienen mayor información y menos código, por lo que un sitio basado en estándares web tendrá una mejor posición.

De igual manera, la posición en directorios, editados por humanos, se verá beneficiada pues el sitio será más usable.

XHTML es una aplicación de XML, por lo que el contenido puede ser procesado de muchas formas, permitiendo la creación de sitios extensibles.

El uso de validadores nos permite crear XHTML bien formado.

Un sitio basado en estándares web es compatible con todos los navegadores actuales, y lo será con versiones futuras. Funcionará tan bien en un PC, un navegador aural y un teléfono móvil dentro de diez años.

Un sitio basado en estándares web es más fácil de mantener y actualizar, el código es más simple, de esta forma se elimina la dependencia de un solo desarrollador.

Un sitio basado en estándares web es más accesible, permitiendo a personas con discapacidades utilizar su contenido.

2.3.5 Requisitos hardware

Los requisitos hardware en la parte cliente para poder mover la aplicación son, hoy en día, superadas por cualquier ordenador personal de los que nos podamos encontrar, aun así los requisitos hardware serían:

Un procesador de 1.0 GHz.

Una RAM de 256 Mbytes.

Los demás datos del ordenador son irrelevantes. Teniendo un ordenador de estas características en adelante nuestra aplicación web funcionara correctamente.

Por la parte del servidor las limitaciones serán de un doble núcleo así como una RAM de 2 GB. Un disco duro de 50 GB para el almacenamiento de la BBDD y en un caso extremo, para la memoria virtual en caso de quedarnos sin RAM al estar sirviendo a demasiada gente. Además de tener una tarjeta de red, así como una conexión mínima de 1 GB.

2.3.6 Atributos del sistema

Mantenimiento

Para llevar el mantenimiento de la pagina debe ser añadiendo productos a la base de datos. Para ello se necesita hacerlo desde la interfaz de phpMyAdmin.

Si se quiere hacer un cambio estructural se deberá recurrir a personal cualificado ya que habrá que modificar código.

Seguridad

Para la gestión de la seguridad de la aplicación se llevara a cabo mediante petición de usuario y contraseña. Cualquier usuario que no pase por este punto lo único que podrá hacer es visitar la página.

Las diferentes cuentas con sus atributos estarán guardadas en el servidor, en la BBDD que se encontrara en el servidor. Para poder modificar las diferentes cuentas con sus privilegios deberá ser a través de phpMyAdmin.

2.3.7 Otros requisitos

Bases de datos

Será necesaria la utilización de una base de datos para poder almacenar toda la información necesaria para el correcto funcionamiento de la aplicación. Se ha optado por almacenar la misma en phpMyAdmin.

3 Análisis

Una vez que ya se han captado los requisitos y necesidades del cliente y de la aplicación que se quiere crear, es el momento de realizar el análisis de forma detallada antes de ponernos a desarrollar.

Para ello la nos hemos ayudado de diagramas UML con los que explicaremos las partes y el funcionamiento de nuestra aplicación y cómo interactúan los diferentes elementos que la componen.

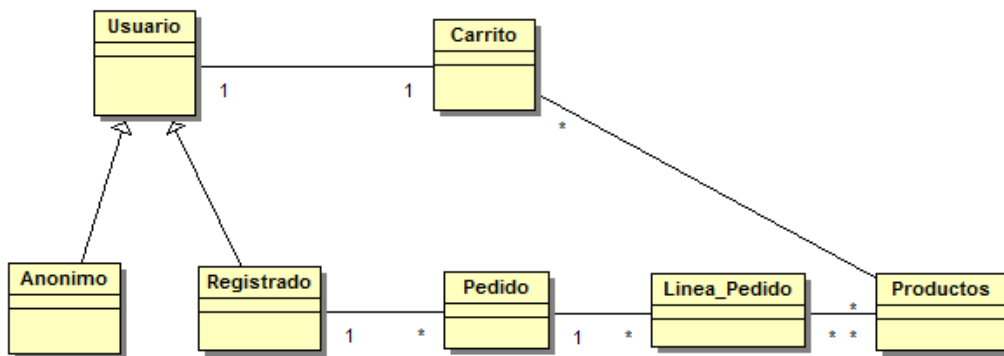
Los diagramas UML que utilizaremos son los siguientes:

- Diagrama de clases.
- Diagrama de casos de uso.
- Diagramas de secuencia.

Pasamos a explicar en detalle cada uno de los diagramas.

3.1 Diagrama de clases.

Para la creación de este diagrama no hemos sido muy estrictos ya que realmente no hay clases en nuestra aplicación debido a la utilización de Ajax y de ficheros PHP, pero para ver claramente la relación de las diferentes partes de la aplicación he considerado oportuno incluirlo.



3.2 Diagrama de casos de uso.

Aquí pasamos a mostrar el diagrama de casos de uso de nuestra aplicación.



3.3 Plantillas de casos de uso

Ahora vamos a especificar cada uno de los casos de uso de nuestra aplicación web mediante plantillas de casos de uso.

3.3.1 Consultar categoría

Caso de uso: Consultar categoría.

Actores: Usuario ocasional y registrado.

Descripción: Accede a los productos de la categoría seleccionada imprimiéndolos en pantalla.

Requerimientos: Ninguno.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario entra en la página web y selecciona una de las categorías de productos.	
	2.- El sistema busca en la base de datos todos los productos de la categoría y lo imprime por pantalla

Curso Alternativo

Acciones de los actores	Acciones del sistema
	2. El sistema no puede comunicarse con el servidor y la base de datos y muestra un mensaje acorde con el problema.

3.3.2 Consultar producto

Caso de uso: Consultar producto.

Actores: Usuario ocasional y registrado.

Descripción: Una vez accedida a la categoría se selecciona un producto para ver su descripción y si se quiere incluir al carro.

Requerimientos: Haber hecho una selección de categoría.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario selecciona uno de los productos para ver el detalle de él.	
	2.- El sistema recupera la información del producto y lo saca por pantalla.

Curso Alternativo

Acciones de los actores	Acciones del sistema
	2. El sistema no puede comunicarse con el servidor y la base de datos y muestra un mensaje acorde con el problema.

3.3.3 Búsqueda producto

Caso de uso: Búsqueda producto.

Actores: Usuario ocasional y registrado.

Descripción: El usuario puede hacer uso de un formulario donde poner una cadena a buscar entre los productos de la base de datos.

Requerimientos: Ninguno.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario pone una cadena en el cuadro de texto y pulsa buscar.	
	2.- El sistema busca en los nombres y en la descripción palabras que coincidan con la palabra a buscar, sacando por pantalla los elementos seleccionados.

Curso Alternativo

Acciones de los actores	Acciones del sistema
	2. El sistema no puede comunicarse con el servidor y la base de datos y muestra un mensaje acorde con el problema.

3.3.4 Alta usuario

Caso de uso: Alta usuario.

Actores: Usuario ocasional.

Descripción: El usuario rellena el formulario de registro para pasar a ser un usuario registrado.

Requerimientos: El usuario no debe estar registrado.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario selecciona la opción de registro de usuario. En esa página rellena los datos que le pide la aplicación.	
	2.- El sistema crea un nuevo usuario con los datos introducidos y le devuelve un mensaje de confirmación.

Curso Alternativo 1

Acciones de los actores	Acciones del sistema
1.-El usuario no rellena todos los campos del formulario.	
	2.- El sistema devuelve un mensaje diciendo que falta algún campo por rellenar y pone un botón para volver otra vez a la página de alta de usuario.

Curso Alternativo 2

Acciones de los actores	Acciones del sistema
	2.- El sistema comprueba que el nombre de usuario o email elegido ya están dados de alta en el sistema comunicándose al usuario rellenar y pone un botón para volver otra vez a la página de alta de usuario.

Curso Alternativo 3

Acciones de los actores	Acciones del sistema
	2.- El sistema comprueba que las contraseñas puestas no coinciden dando un mensaje de error y pone un botón para volver otra vez a la página de alta de usuario.

3.3.5 Contacto

Caso de uso: Contacto.

Actores: Usuario ocasional y registrado.

Descripción: El usuario accede a la página donde puede rellenar los datos y escribir un correo para comunicarse con la empresa.

Requerimientos: Ninguno.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario selecciona en el menú la opción de contacto, rellena el formulario y pulsa enviar.	
	2.- El sistema crea un correo y lo envía al propietario de la aplicación.

Curso Alternativo

Acciones de los actores	Acciones del sistema
1.-El usuario no rellena todos los campos del formulario.	
	2.- El sistema devuelve un mensaje diciendo que falta algún campo por rellenar y vuelve otra vez a poner la página de contacto.

3.3.6 Añadir al carro

Caso de uso: Añadir al carro.

Actores: Usuario ocasional y registrado.

Descripción: El usuario una vez en la página de descripción de un producto selecciona añadir al carro poniendo la cantidad de quiere de los mismos.

Requerimientos: Ninguno.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario accede al producto y pone la cantidad de producto que quiere y pulsa añadir.	
	2.- El sistema añade al carro el producto con la cantidad seleccionada, visualizándolo en el carro de la barra lateral derecha.

3.3.7 Quitar del carro

Caso de uso: Quitar del carro.

Actores: Usuario ocasional y registrado.

Descripción: El usuario quita productos que tenía en el carro.

Requerimientos: Ninguno.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario quita del carrito de la barra lateral derecha o desde su perfil un elemento.	
	2.- El sistema quita del carro el elemento seleccionado.

3.3.8 Realizar compra

Caso de uso: Realizar compra.

Actores: Usuario registrado.

Descripción: El usuario una vez que tiene hecha la compra puede hacer efectivo el pedido dando a comprar, una vez que ha accedido al carro.

Requerimientos: El usuario debe estar registrado en el sistema y en la pantalla del carro.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario entra en el carro tanto por la entrada que esta en el lateral como la que esta en la entrada de la cuenta. Allí el usuario pulsa comprar.	
	2.- El sistema crea el pedido y las líneas de pedido y saca la factura por pantalla.

Curso Alternativo

Acciones de los actores	Acciones del sistema
	2.- El sistema no puede comunicarse con el servidor y la base de datos y muestra un mensaje acorde con el problema.

3.3.9 Modificar datos de usuario

Caso de uso: Modificar datos de usuario.

Actores: Usuario registrado.

Descripción: El usuario accede a su cuenta y allí puede cambiar los datos con los que se dio de alta.

Requerimientos: El usuario debe estar registrado en el sistema.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario entra en su cuenta y pulsa modificar datos. Allí cambia los datos y pulsa guardar.	
	2.- El sistema cambia los datos del usuario de la base de datos, saca un mensaje comunicándolo y pone un botón para volver al menú de la cuenta.

Curso Alternativo

Acciones de los actores	Acciones del sistema
1.-El usuario no rellena todos los campos del formulario.	
	2.- El sistema devuelve un mensaje diciendo que falta algún campo por rellenar y pone un botón para volver a sacar el formulario.

3.3.10 Visualizar historial de pedidos

Caso de uso: Visualizar historial de pedidos.

Actores: Usuario registrado.

Descripción: El usuario accede a su cuenta y allí puede visualizar el historial de pedidos que ha realizado el usuario.

Requerimientos: El usuario debe estar registrado en el sistema.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario entra en su cuenta y pulsa ver pedidos.	
	2.- El sistema saca todos los pedidos realizados por el usuario.

Curso Alternativo

Acciones de los actores	Acciones del sistema
	2.- El sistema no puede comunicarse con el servidor y la base de datos y muestra un mensaje acorde con el problema. Aquí va un botón

3.3.1 Cambiar contraseña

Caso de uso: Cambiar contraseña.

Actores: Usuario registrado.

Descripción: El usuario accede a su cuenta y allí puede cambiar contraseña.

Requerimientos: El usuario debe estar registrado en el sistema.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario entra en su cuenta y pulsa cambiar contraseña. Allí pone su antigua contraseña y la nueva por duplicado.	
	2.- El sistema comprueba que la contraseña antigua es correcta, y que las nuevas coinciden.
	3.- El sistema cambia la contraseña del usuario en la base de datos.

Curso Alternativo 1

Acciones de los actores	Acciones del sistema
1.-El usuario no rellena todos los campos del formulario.	
	2.- El sistema devuelve un mensaje diciendo que falta algún campo por rellenar y pone un botón para volver a sacar el formulario.

Curso Alternativo 2

Acciones de los actores	Acciones del sistema
	2.- El sistema comprueba que la contraseña no se corresponde con la contraseña del usuario loggeado sacando por pantalla el mensaje pertinente y pone un botón para volver a sacar el formulario.

Curso Alternativo 3

Acciones de los actores	Acciones del sistema
	2.- El sistema comprueba que las contraseñas nuevas no coinciden sacando el mensaje pertinente y poniendo un botón para volver al formulario.

3.3.12 Cerrar sesión

Caso de uso: Cerrar contraseña.

Actores: Usuario registrado.

Descripción: El usuario cierra la sesión para dejar de ver su cuenta.

Requerimientos: El usuario debe estar registrado en el sistema.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario pulsa el botón de cerrar sesión tanto desde la barra lateral como desde el menú de cuenta de usuario.	
	2.- El sistema elimina la sesión del servidor y pone el formulario para logarse en la barra lateral derecha.

3.3.13 Recuperar contraseña

Caso de uso: Recuperar contraseña.

Actores: Usuario ocasional como registrado.

Descripción: El usuario accede al formulario de recuperar contraseña y ahí pone el correo con el que se dio de alta.

Requerimientos: Ninguna.

Curso normal

Acciones de los actores	Acciones del sistema
1.- El usuario pulsa en recuperar contraseña, pone su correo y pulsa enviar.	
	2.- El sistema busca el email en la base de datos y envía la contraseña.

Curso Alternativo 2

Acciones de los actores	Acciones del sistema
1.-El usuario no rellena todos los campos del formulario.	
	2.- El sistema devuelve un mensaje diciendo que falta algún campo por rellenar y pone un botón que vuelve otra vez a poner la página de recuperación de contraseña.

Curso Alternativo 3

Acciones de los actores	Acciones del sistema
	2.- El sistema no encuentra el email que ha puesto el usuario sacando un mensaje comunicándolo y pone un botón que vuelve otra vez a poner la página de recuperación de contraseña.

3.3.14 Visualizar un pedido

Caso de uso: Visualizar pedido.

Actores: Usuario registrado.

Descripción: El usuario accede al historial de pedidos. Desde allí puede ver un pedido en detalle.

Requerimientos: Estar registrado en el sistema y haber hecho algún pedido.

Curso normal

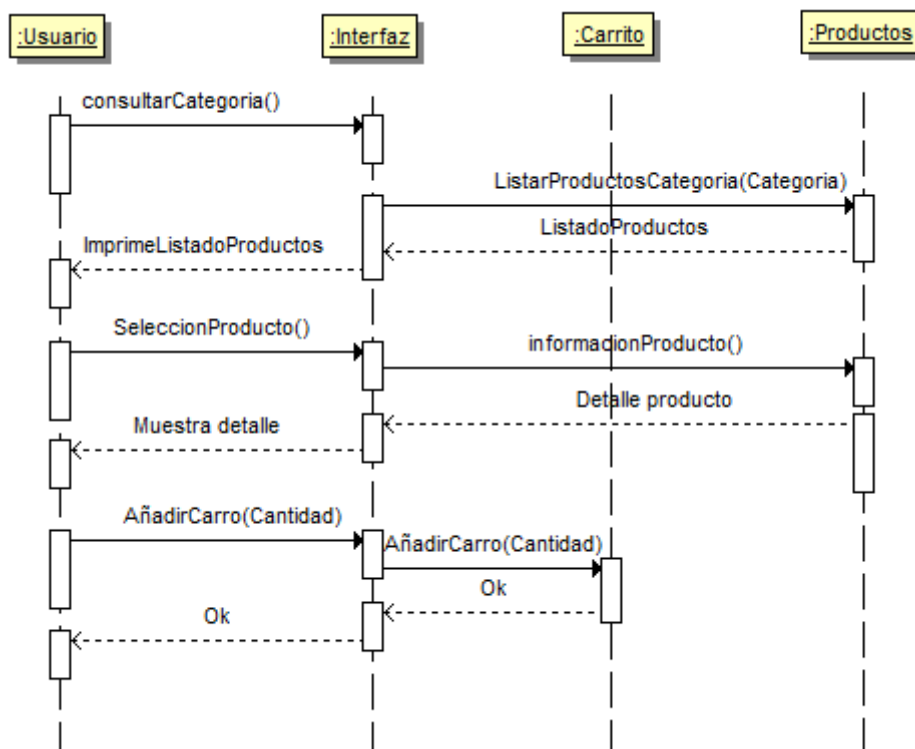
Acciones de los actores	Acciones del sistema
1.- El usuario entra en el historial de sus pedidos y ahí pulsa en el pedido que quiere ver.	
	2.- El sistema busca el pedido que quiere ver y las líneas del pedido que lo componen, sacándolo por pantalla.

3.4 Diagramas de secuencia.

Ahora pasamos a mostrar una serie de diagramas de secuencia que considero mas importantes, para reflejar la comunicación entre los diferentes elementos.

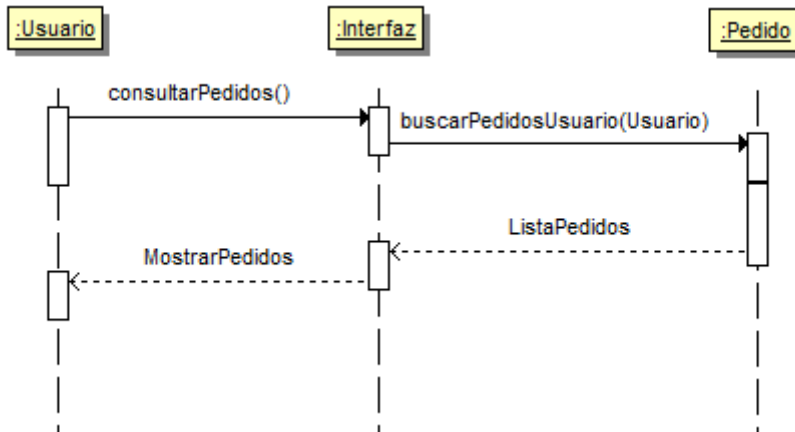
3.4.1 Añadir al carrito.

A continuación mostramos el diagrama de secuencia para añadir un elemento al carro de la compra desde la pantalla de inicio, donde tenemos que obtener primero los elementos de una categoría y luego elegir un producto y añadirlo al carro.



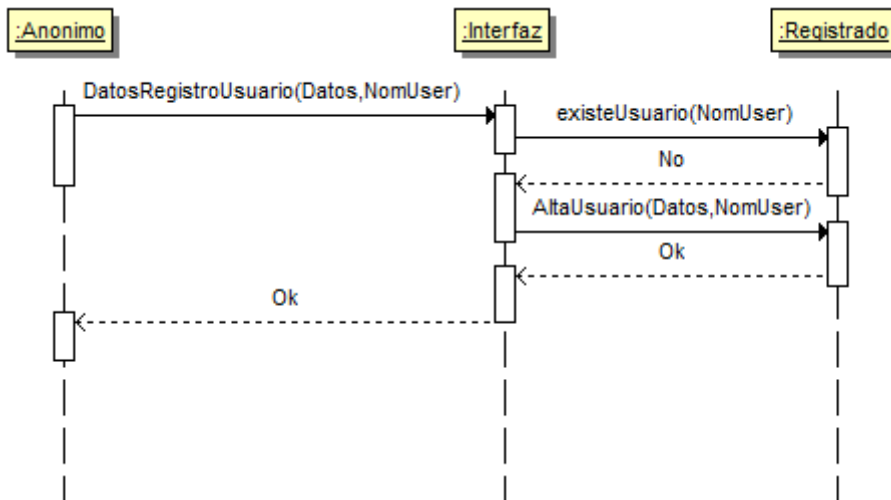
3.4.2 Listar el historial de los pedidos de un usuario

A continuación mostramos el diagrama de secuencia para listar el historial de pedidos realizados por un usuario.



3.4.3 Registro de usuario nuevo.

A continuación mostramos el diagrama de secuencia para realizar el registro de nuevo usuario en el sistema.

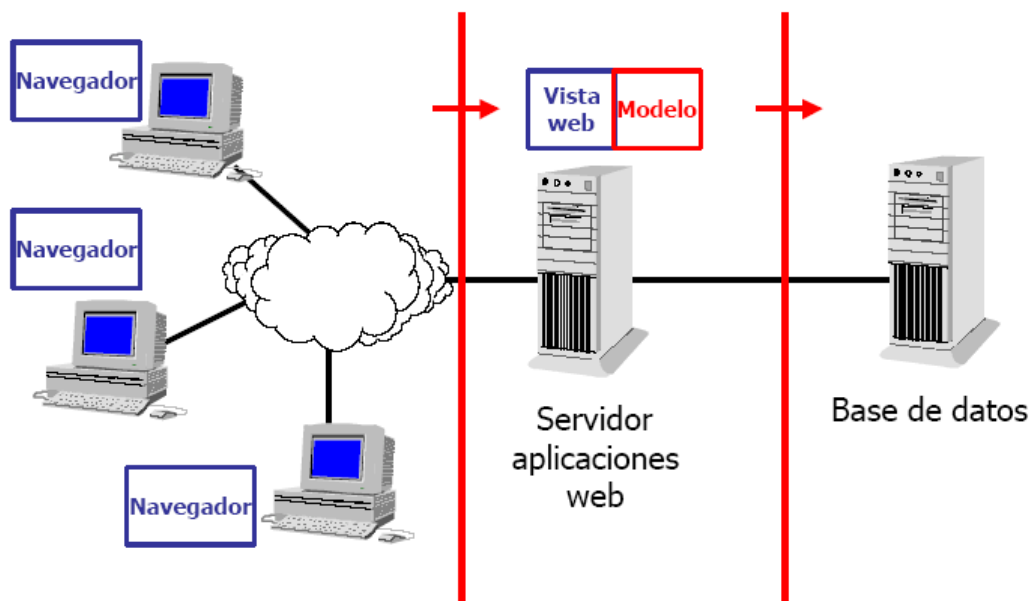


4. Diseño

En este apartado hablaremos de la arquitectura del sistema y de cómo está estructurada.

4.1. Arquitectura del Sistema

Para el diseño de la aplicación hemos utilizado la arquitectura tres capas. Esta arquitectura se basa en una capa de presentación, otra capa de negocio y la capa de datos.



- **La capa de presentación:** Esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Básicamente, se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa, manteniendo una comunicación exclusiva con la capa de negocio que veremos a continuación. Además dentro de esta capa entraría aquello que el usuario “ve” cuando se conecta a la aplicación. En este proyecto al estar realizado con JavaScript adquiere un poco más de relevancia ya que una parte mínima de las operaciones las realiza el cliente web, como formar formularios.
- **La capa de negocio:** Es la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación, las peticiones que el usuario ha realizado, y responsabilizándose de que se le envíen las respuestas adecuadas a la capa de presentación. Podríamos verla como una capa intermedia, a medio camino entre la capa de presentación y la capa de datos, puesto que se relaciona con ambas y por supuesto, procesa también la información devuelta por la capa de datos.
- **La capa de persistencia:** Por último, la capa donde se almacenan los datos. Mediante la capa de negocio, se puede encargar de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor (o los gestores) de bases de datos que la aplicación requiera.

4.2 Nivel de interfaz.

En esta parte pasamos a explicar la interfaz que ve el usuario y con la que interactúa. Al ser una aplicación que trabaja con Ajax tendremos siempre una misma página en la cual iremos cambiando el DOM para añadir o quitar objetos dependiendo de la operación a realizar.

Para explicar esta sección nos referiremos en cada parte a componer la página, ya que siempre nos encontramos en una misma página y la vamos cambiando dependiendo de la operación que elija el usuario.

Para explicar la interfaz empezaremos por la página de inicio que se ve de esta manera.



- **Navegador Horizontal:**

En el cuadro marcado con 1 nos encontramos con el menú horizontal donde encontramos la entrada de menú para componer la página de inicio, que es la que estamos viendo ahora mismo.

En segundo lugar del menú vemos contacto que nos compone el formulario para contactar con la empresa de la página. Cuando se pulsa esa entrada del menú el formulario se construye en la parte que esta con el recuadro con el numero 7.

En la tercera entrada encontramos el foro que nos llevaría a un foro externo a la aplicación.

- **Navegador lateral izquierdo:**

En el recuadro marcado como dos vemos el menú de categorías de semillas donde pulsando en una de ellas se compondrá la página con los elementos que forman la categoría en el cuadro marcado con 7.

- **Carrito:**

En el cuadro tres encontramos la cesta de la compra. A medida que vayamos añadiendo productos se irán situando aquí. Veamos una captura.

Carrito			
2	SANDIA S	1.95€	X
3	PRIMAVER	1€	X
2	EUROPEAN	2.49€	X
	TOTAL::	11.88€	
Mostrar carrito de la compra			
Buscador de productos			
Palabra a buscar			

Como observamos podemos eliminar elementos de la cesta pulsando en la X situada a la derecha de cada línea de elementos, y pulsando [Mostrar cesta de la compra](#) accedemos a la cesta para poder comprar el contenido.

Si pulsamos en el icono de la papelera podremos borrar la línea donde está la cesta. Si pulsamos en el símbolo sacará una factura del pedido y un botón de realizar compra que hace efectivo el pedido, haciendo que pase a formar parte del historial de pedidos.

- **Buscador:**
En el recuadro con el número 4 encontramos el buscador. Aquí pondremos la cadena a buscar y nos mostrar los elementos que coinciden con la búsqueda. Si no encuentra ningún elemento lo comunica con el mensaje pertinente.
- **Formulario de registro**
En esta parte encontramos el formulario que utilizamos para logarnos en la página, así como para hacernos cuenta o para recuperar la contraseña.
- **-Foro:**
Aquí encontramos otra entrada al foro.
- **Cuerpo de la página:**
En esta sección es donde iremos poniendo las diferentes composiciones de pantalla.

La siguiente parte importante de la aplicación es la parte de la cuenta de usuario que es donde podemos acceder a nuestros datos personales y nuestros pedidos. Aquí tenemos una captura de cómo sería la vista.

En esta vista podemos acceder a:

- **Administrar datos personales:**
Aquí podremos cambiar los datos que nos permita la aplicación mediante un formulario como sigue.
- **Cambiar contraseña:** Aquí podremos cambiar la contraseña mediante el siguiente formulario.

- **Cerrar sesión:** Desde aquí podremos salir de la sesión, borrando también todos los elementos que teníamos puestos en el carro.
- **Pedidos:** En esta entrada podremos ver todos los pedidos que hemos hecho en la aplicación. Es un historial. Dentro de la tabla de los pedidos podemos acceder en detalle a cada uno de ellos pulsando en la lupa, como podemos ver a continuación.

4.3 Nivel de negocio.

En esta parte es en donde trabajaremos con las peticiones hechas por la capa presentación las cuales de la mayoría necesitaran de llamadas a la base de datos.

Para poder distinguir esta parte se ha separado los ficheros php. Para poder realizar la aplicación lo mas dinámica posible debe ser el cliente web el que debe hacer las peticiones mediante AJAX y el resultado que recibirá en HTML el propio AJAX lo montara en una parte del DOM. Puede ser en la parte central como en el carrito, así como cuando el usuario se loguea. En todos estos casos es AJAX el que le hace la petición de lo que se quiere hacer, se lo comunica a un fichero PHP y este lo redirección al fichero php correspondiente.

Esto se hace para poder, por así decirlo, llamar a funciones desde AJAX a un fichero php. Desde AJAX pasamos los parámetros y un índice a un fichero php. Dependiendo del índice que recibe el fichero PHPFile.php redirige a una función o a otra, siendo ese fichero el que llama a una función php en concreto de un determinado fichero.

Tenemos un fichero PHPFile.php que actúa como redireccionador y luego un fichero func_usuarios.php, donde encontramos todas las funciones referentes a usuarios como darse de alta, cerrar sesión, modificar datos, etc. También tenemos el fichero func_productos.php, func_carrito.php, y func_pedidos.php, en los cuales hacen las respectivas operaciones.

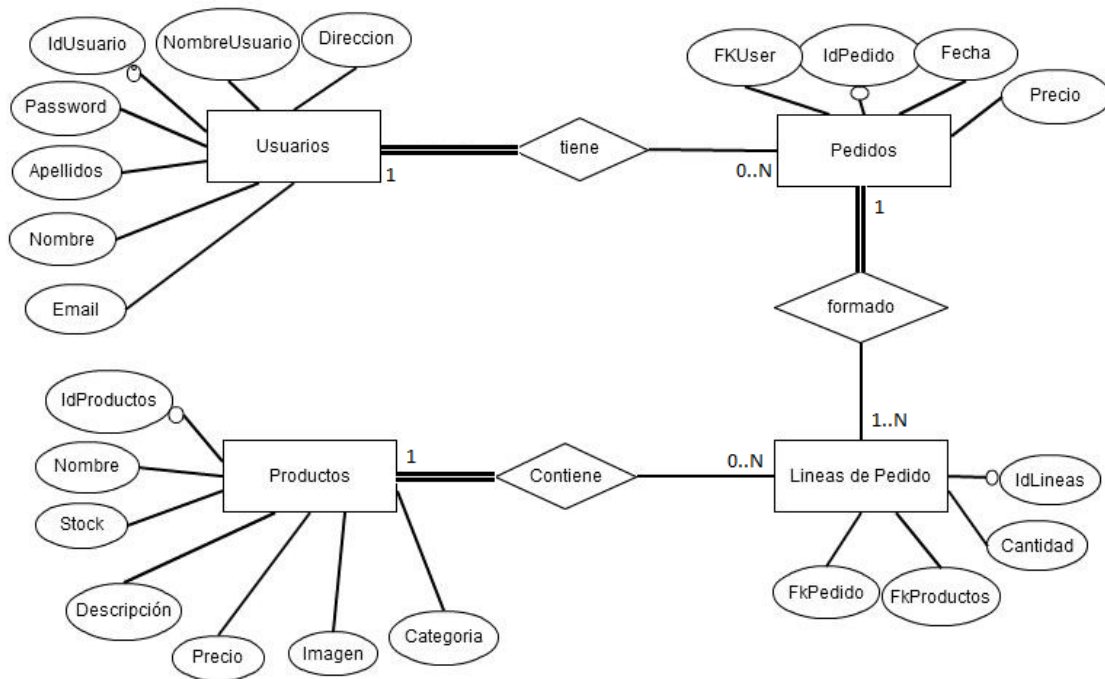
Casi todas las funciones se comunican con la base de datos para poder traer información, eliminarla o modificarla, por lo que esta capa hace de enlace entre el usuario y la base de datos. Además al estar separadas por tipo es mas legible y entendible para programadores externos, para en un futuro posibles modificaciones.

4. 4 Capa de persistencia.

En esta capa es donde están alojados los datos. Aquí es un gestor de base de datos el que mantiene eficientemente la persistencia de los datos y la integridad.

Esta capa es la encargada de buscar los datos pedidos por la capa de negocio, así como la modificación y la eliminación de registros.

A continuación mostramos el diagrama entidad-relación de la base de datos con la que trabaja la aplicación.



4.4.1 Diseño lógico.

Usuarios(IdUsuario: INTEGER, NombreUsuarios: varchar(10), Password: VARCHAR (20), Nombre: VARCHAR(20), Apellidos: VARCHAR (20), Direccion: VARCHAR (20), Email: varchar(50))
 CP{IdUsuario}
 VNN{ IdUsuario, NombreUsuarios, Password, Nombre, Apellidos, Direccion, Email}

Pedidos(IdPedidos: INTEGER, Fecha: DATETIME, Precio: FLOAT, FkUser: INTEGER)
 CP{IdPedidos}
 CAj{FkUser} → Usuarios
 VNN{ IdPedidos, Fecha, Precio, FkUser }

LineasPedido(IdLineas: INTEGER, cantidad: INTEGER, FkProducto: INTEGER, FkPedido: INTEGER)
 CP{IdLineas}
 CAj{ FkPedido } → Pedidos
 CAj{ FkProducto } → Productos
 VNN{ (IdLineas, cantidad, FkProducto, FkPedido) }

Productos(IdProductos: INTEGER, Nombre: VARCHAR(100),Descripcion: VARCHAR(500) Precio: FLOAT,Imagen: VARCHAR(200),Categoría: VARCHAR(100))
 CP{ IdProductos }
 VNN{ IdProductos, Nombre, Descripcion, Precio ,Imagen, Categoría}

5. Implementación e integración

En esta sección pasamos a detallar las tecnologías que hemos utilizado para hacer la aplicación. También hablaremos de las herramientas que han facilitado nuestro trabajo y por último hablaremos de los detalles más importantes que caracterizan a la implementación.

5.1. Tecnologías

Una de las partes importantes del proyecto es el dinamismo que transmite debido a la modificación del DOM, trabajando siempre con la misma página HTML y modificando esta en el navegador del ordenador del usuario.

Para poder realizar la aplicación hemos utilizado las siguientes tecnologías, de las cuales más abajo se encuentran las definiciones que he buscado en la red. Las he referenciado con el número entre paréntesis, al final del texto con las que están numeradas las direcciones web en la bibliografía. (18)

5.1.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos. (9)

5.1.2 PHP

PHP (HyperText Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado

del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.

PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI.

PHP fue desarrollado originalmente por Rasmus Ledford en 1994 como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y consiguió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Cuando Rasmus tuvo la necesidad de crear páginas dinámicas que trabajasen con formularios, creó una serie de etiquetas a las que denominó "Form Interpreters", y lo sacó al público con el nombre de PHP/FI en 1995. Luego salió la versión mejorada, llamada PHP/FI 2.0. (10)

5.1.3 CSS:

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los **Estilos** definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (11)

5.1.4 AJAX:

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM). (12)

5.1.5 HTML:

El HTML o HyperText Markup Language, en inglés, es el lenguaje utilizado para las páginas web o de Internet. En español su traducción vendría a ser Lenguaje de Etiquetado de Documentos hipertextual. El etiquetado se refiere a que los archivos HTML, además de texto, contienen una



serie de atribuciones o información adicional acerca del texto. Hipertexto, término acuñado en 1965 por Ted Nelson, es información no lineal, lo que en palabras sencillas significa texto con enlaces o links a otros textos. El lenguaje HTML estructura la información, que luego es visualizada utilizando un navegador. (13)

5.1.6 Arquitectura Cliente Servidor:

El modelo cliente-servidor (client-server), describe el proceso de interacción entre la computadora local (el cliente) y la remota (el servidor).

El cliente le hace peticiones (requests, solicitudes, requerimientos) al servidor, el cual procesa dicho requerimiento y retorna los resultados al cliente apropiado.

Por lo general, los clientes y los servidores se comunican entre sí a través de una red, pero también pueden residir ambos en un mismo sistema (el mismo hardware).

Siguen este modelo de cliente/servidor servicios como el intercambio de emails, el acceso a webs, el acceso a bases de datos, y muchos otros protocolos de internet se basan en esta idea (HTTP, SMTP, Telnet, DNS), etc. (14)

5.2. Herramientas

A continuación pasaremos a explicar brevemente las herramientas que hemos utilizado para el desarrollo del proyecto.

5.2.1 WAMP

(Windows-Apache-MySQL- PHP/Python/PERL). El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto) y el sistema operativo Windows. Este grupo de aplicaciones generalmente son usados para crear servidores web.

WAMP provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script web (PHP, Python o PERL).

Otra popular combinación de aplicaciones es el LAMP, que es totalmente gratuito. También existen el MAMP y el FAMP. (15)

5.2.2 PhpMyAdmin

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. (16)

5.2.3 Notepad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Solo funciona en Microsoft Windows.

Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores.
Se distribuye bajo los términos de la Licencia Pública General de GNU. (17)

5.3. Detalles de la implementación

Ahora pasaremos a explicar los detalles de como hemos implementado la aplicación.

5.3.1 Perfiles de usuario.

El uso de la aplicación la va a realizar dos perfiles distintos de usuarios. Uno es el perfil del usuario ocasional y otro el usuario registrado.

- **Perfil de usuario ocasional**

Este usuario tiene una lista de operaciones reducida. Al no estar aun registrado las únicas operaciones que puede hacer es visitar las diferentes categorías que dispone el catalogo. Puede añadir productos a la cesta de la compra así como eliminarlos. Estando en esta parte de la compra se le invita a que realice el pedido, así podremos captar su atención. Si el usuario intenta realizar la compra salta un mensaje diciendo que debe registrarse para poder hacer efectiva la compra.

Por otra parte tiene acceso para ponerse en contacto con la empresa, para exponer sus inquietudes, además de tener acceso al foro. En el foro solo podrá visitar los hilos que haya abiertos pero no podrá postear ni abrir nuevos debates.

- **Perfil de usuario registrado**

Este es el perfil de los usuarios que se han dado de alta en la aplicación. Puede realizar todas las operaciones que realiza el usuario no registrado.

Para poder logarse en la aplicación deben poner su usuario y contraseña en el cuadro lateral derecho. Una vez que el sistema comprueba que el usuario y la contraseña son correctos la página se modificara dando la bienvenida al usuario y modificando la página con dos nuevas entradas en vez del cuadro de login.

En estas entradas tenemos la cuenta y el cierre de sesión, características de este usuario. En la entrada cuenta el usuario accede a su zona personal, donde puede cambiar sus datos y mirar su historial de pedidos.

5.3.2 Autenticación de usuarios.

La aplicación para el seguimiento de usuarios se basa en sesiones. El usuario cuando se loguea se crea en el servidor una entrada en la sesión con el nombre del usuario. Así cuando el usuario intenta acceder a alguna parte de la aplicación en la cual necesita estar logueado, la aplicación lo comprueba, si no hay nadie logueado se le invita a que se loguee.

Para la comprobación de los datos AJAX hace una petición al servidor con los datos del usuario y este los comprueba en la base de datos. Si la base de datos da una respuesta válida se crea la entrada en la sesión, si no se le informa al usuario de que la contraseña o usuario son incorrectos.

```

function conectarUsuario()
{
    $nom=$_POST['Login'];
    $pas=$_POST['Password'];

    $conex = mysql_connect("localhost", "root", "")
        or die("NO se pudo realizar la conexioacuten");

    // Seleccioacuten de la base de datos
    mysql_select_db("bbddproyecto")
        or die("ERROR con la base de datos");

    //Preparacioacuten y ejecucioacuten de la consulta
    $consulta = "SELECT * FROM Usuarios WHERE NombreUsuario='".$nom.'" and Password='".$pas.'"";
    $resultado = mysql_query($consulta);
    $row = mysql_fetch_row($resultado);
    if(!$row){
        echo "<br>";
        echo "<h3>El usuario o la contraseñtildea son incorrectas.</h3>";
    }else
    {
        iniciarSesion($nom);
    }
}

```

Cuando el usuario cierra la sesión destruimos la sesión con `sesión_destroy()`, así también eliminamos lo que tenía el usuario en el carrito.

5.3.3 Selección fichero PHP.

La aplicación como esta hecha se basa en un fichero php que actúa como índice. Este índice redirecciona la petición que hace AJAX al fichero php que contiene la función que se solicita, haciendo así que AJAX solicite una función con sus parámetros. La idea es que no haya un fichero php por cada función que necesita llamar Ajax.

Esta función se puede encontrar en el fichero `PhpFile.php`

La función AJAX pasara al fichero índice un numero de índice, el cual le dice al fichero la función que quiere ejecutar, y una serie de parámetros.

5.3.4 Inserción de HTML en el DOM

La parte importante de la aplicación es la inserción de HTML en el DOM. De esta manera solo tenemos una página en la que iremos construyendo dependiendo de la zona en la que estemos.

La modificación del DOM viene por parte de AJAX. AJAX hace una petición de una función al servidor, este le devuelve el código que tiene que incrustar o modificar (en varios casos quitaremos un código para poner otro) y AJAX lo pone en el DOM creando así una nueva página y dando dinamismo a la misma.

El código insertado que viene desde AJAX es cuando el código introducido no necesita información por parte de la base de datos, por esta razón podemos tener funciones en el fichero JavaScript que ponen HTML. Normalmente estos son los formularios que no necesitan de información como el de registrarse en la aplicación y el de contacto.

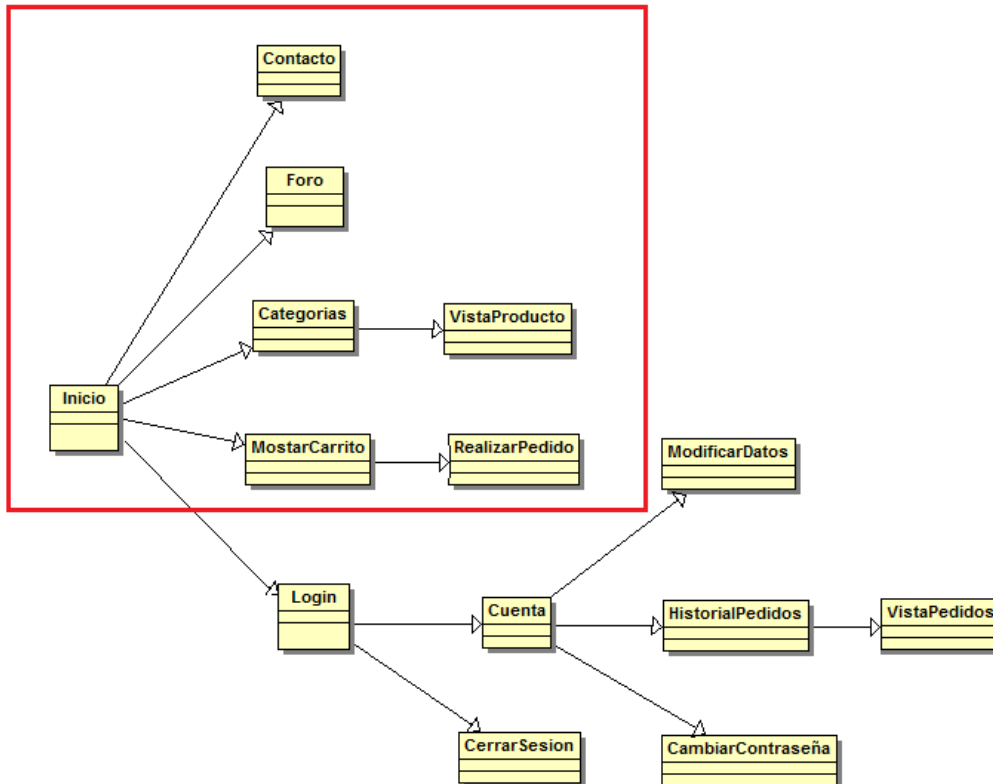
Veamos un ejemplo de cómo se inserta código como de un típico formulario que se puede montar desde JavaScript sin la necesidad del servidor. Como se puede observar el código se incrusta en el div con identificador llamado "cont".

```
function formNuevoUser()
{
  eInfo = document.getElementById("cont")
  eInfo.innerHTML = "<fieldset id='registrouuario'>" +
    "<form action='' method='POST' style='display: block;position: relative;margin" +
    " <legend>Registro de Nuevo Usuario</legend>" +
    "<label for='user'>Usuario:</label>" +
    "<input class='texto' id='username' type='text' size='20' />" +
    "<label for='pass'>Contrase&ntildea:</label><input class='texto' id='password'" +
    "<label for='pass'>Repita Contrase&ntildea:</label><input class='texto' id='pa" +
    "<label for='pass'>E-mail:</label><input class='texto' id='email' type='text'" +
    "<label for='pass'>Nombre:</label><input class='texto' id='firstname' type='te" +
    "<label for='pass'>Apellidos:</label><input class='texto' id='lastname' type=''" +
    "<label for='pass'>Direccisoacuten:</label><input class='texto' id='address' t" +
    "<input class='botonenviar' name='insert' type='button' value='Enviar' class=''" +
    "<input type='hidden' name='tipo' value='4'>" +
    "</form>"
}
```

5.4 Diagrama de navegabilidad.

Para poder ver una relación de las páginas que puede componer nuestra aplicación hacemos uso de un diagrama de navegabilidad. Con este diagrama vemos la diferentes paginas que componen nuestra web y como podemos llegar a ellas.

Hemos distinguido con colores para ver las páginas que puede ver un usuario ocasional, ya que todas las demás son la que puede ver un usuario registrado.



6. Evaluación y pruebas

Una vez que hemos acabado con la fase de implementación debemos realizar una serie de pruebas y evaluar nuestra aplicación antes de darla por concluida. Gracias a esto podremos identificar fallos para corregirlos antes de dar por terminada nuestra aplicación.

6.1 Evaluación.

Para evaluar nuestra página web vamos a tener que buscar una serie de características comportándonos lo mas objetivamente posible.

Las características a las que nos referimos son las que nos aporta la usabilidad. Cuando hablamos de usabilidad en una página web nos estamos refiriendo a la capacidad que tiene esta para resultar intuitiva, fácil, productiva y entendible para un usuario ocasional, el cual no suele visitar una página. En resumen, que la pagina le resulte lo mas productible posible en el menor tiempo de visita.

Para conseguir esto nos basamos en 4 conceptos que hacen una web usable:

- Ponernos siempre en la posición del usuario para intuir como desearía la pagina para encontrar lo que busca.
- Tener en la pagina un árbol de navegación sin demasiados enlaces e intuitivo.
- Hacer una página con un diseño visual atractivo sin que sea recargado ni pesado.
- Tener en cuenta las posibles limitaciones que tengan los usuarios como son el ancho de banda o la resolución.

En todas las fases que hemos llevado a cabo para la resolución de la aplicación se ha tenido siempre encuentra estos factores. Nuestra página tiene el formato normal que tienen normalmente las tiendas online, por lo que es lo que se espera encontrar el usuario. Nuestro árbol de navegación tiene, como máximo, una profundidad de 4 pantallas que es la de su caso más extenso, el de realizar el pedido. La página se ha hecho de la manera más atractiva que se ha podido, ya que considero que ese es un trabajo de un diseñador y que está orientado mas la bellas artes, de todas maneras, he puesto todo mi empeño en que así lo sea. Y por último se ha tenido en cuenta las resoluciones, y no se ha recargado con flash ni animaciones complicadas por lo que resulta muy ligera.

6.2 Pruebas.

Las pruebas que vamos a realizar para poder hacer una aproximación de que esta correcta la pagina en algunos sentidos es la de comprobación de enlaces rotos, la de validación de estándares W3C, la comprobación de el funcionamiento en los distintos navegadores y las pruebas de resolución de pantalla, antes de pasar a hacer unas pruebas funcionales de algunos casos de uso, para comprobar que se ejecutan correctamente.

6.2.1 Enlaces rotos.

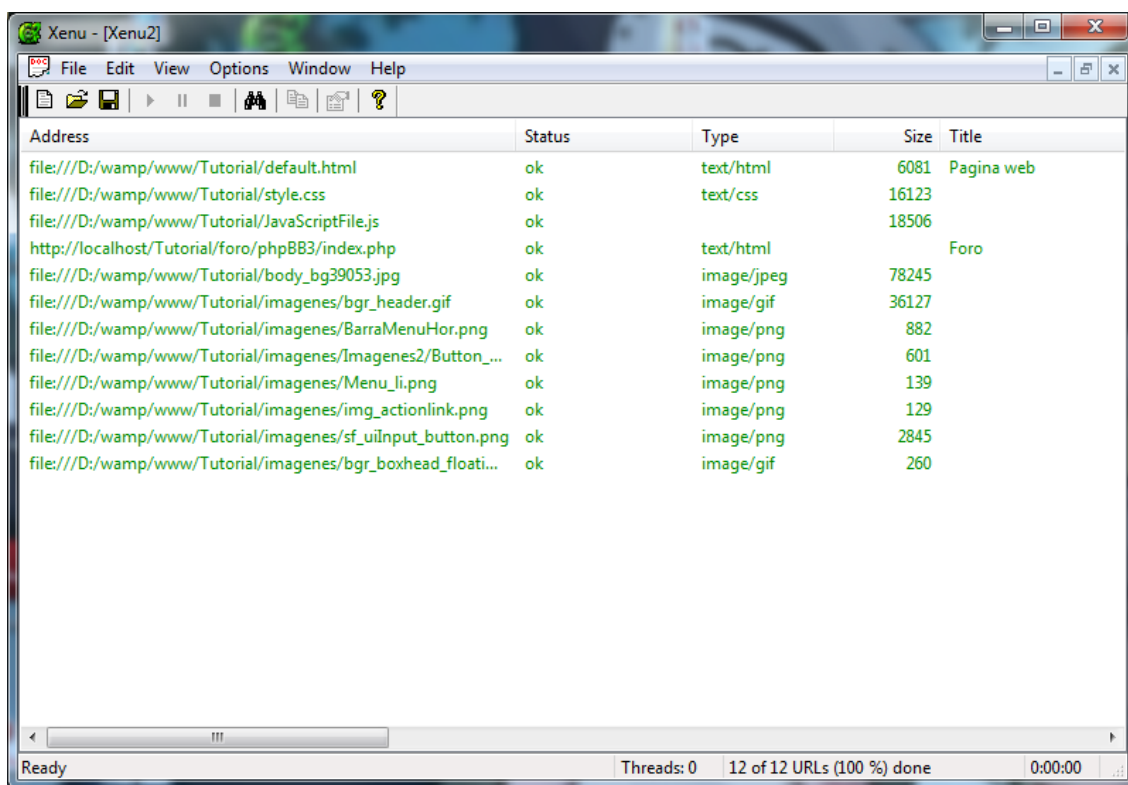
Para comprobar toda la navegabilidad de la aplicación comprobando los enlaces nos hemos ayudado de aplicaciones gratuitas que hay en la web.

Un enlace roto dificulta a los buscadores la indexación de la página, le dice al buscador que algo anda mal, que tiene poco mantenimiento o que no se actualiza. Tampoco deja muy buena

impresión a los usuarios. Es decir, los enlaces rotos perjudican nuestro posicionamiento web.

Después de estar buscando nos hemos tenido que basar en una aplicación que lo compruebe en local, ya que nuestra página no está en la red. El programa que hemos elegido es el Xenu.

Después de hacer unos cambios en el fichero del css hemos quitado todos los enlaces rotos. Aquí mostramos una captura de pantalla del resultado de pasar la ruta donde se muestra que todos los enlaces están correctos.



6.2.2. Pruebas de Validación HTML y CSS.

Otra de las pruebas útiles es la validación de los estándares de HTML y CSS. Gracias a esto podemos encontrar errores que los navegadores son capaces de ocultar y que no se saben que están allí, pudiendo aparecer en el momento mas inesperado.

Para poder realizar estas pruebas hemos ido a la página oficial de W3C y allí hemos puesto nuestros ficheros uno por uno y hemos comprobado que todo esta correcto.

En primer lugar hemos puesto el fichero index.html, en el cual hemos encontrado unos errores de cerrar tags y una lista de elementos que estaba vacía y que había que borrar. Después de esto hemos dejado el documento correcto para el estándar. El mensaje que ha mostrado la aplicación es el siguiente.

This document was successfully checked as HTML 4.01 Transitional!	
Result:	Passed
File:	<input type="text" value="Seleccionar archivo"/> No se ha seleccionado ningún archivo <small>Use the file selection box above if you wish to re-validate the uploaded file default.html</small>
Encoding:	utf-8 <input type="button" value="(detect automatically)"/>
Doctype:	HTML 4.01 Transitional <input type="button" value="(detect automatically)"/>
Root Element:	html

Seguidamente hemos comprobado el fichero CSS. Después de pasarlo nos ha encontrado 10 errores y 18 warnings. Una vez que se ha limpiado el fichero de errores hemos vuelto a pasarlo por la aplicación web y el resultado ha sido positivo dando el siguiente mensaje.

Resultados del Validador CSS del W3C para style.css (CSS versión 3)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:

```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
  
</a>  
</p>
```

```
<p>  
<a href="http://jigsaw.w3.org/css-validator/check/referer">  
  
</a>  
</p>
```

6.2.2. Pruebas de compatibilidad en navegadores

Para estas pruebas lo que hemos hecho es poner nuestra página en los diferentes navegadores y pasar una serie de pruebas como son loguearse, ver una categoría, hacer una búsqueda, añadir al carrito, quitar elementos del carrito y hacer un pedido.

Al hacer estas pruebas podemos decir que estamos pasando por casi toda la aplicación y podemos afirmar en gran medida que, al pasar un navegador todas estas pruebas funcionales funcionara correctamente para el resto de la aplicación.

Las pruebas y resultados se a hecho en los siguientes navegadores:

-Safari 15.0.1:

Al ser el navegador donde hemos desarrollado la aplicación funciona todo correctamente.

-Google Chrome 21.0.1180.89 m:

Este es el segundo navegador donde se hacían las pruebas, también funciona correctamente.

-Explorer 9:

En este navegador tenemos problemas con las cajas que montamos para los elementos de cada categoría. Salen descuadrados como vemos a continuación, por lo demás funciona correctamente.



-Opera 12.02.

Para este navegador tampoco se ha encontrado nada diferente. Funciona igual que el Mozilla y que el Chrome.

-Safari 5.1.7.

Para este navegador tampoco se encuentra ningún problema.

6.2.3. Pruebas de compatibilidad de resolución

Para realizar estas pruebas probamos con diferentes resoluciones: 1280x768, 1280x720, 1024x768, 1024x600, 800x600.

En todas ellas se comporta bien la página. En 800x600 al cambiar a esa resolución pone scrolls tanto abajo como laterales, pero si aumentamos la resolución que nos proporciona el propio explorador quedaría perfecto, como se ve en la captura.



6.2.4 Casos de prueba.

Como es normal a medida que se va haciendo la aplicación y añadiendo funcionalidades las vas probando, para comprobar que funcionan correctamente. Este son los casos de prueba básicos de un programador. De todas maneras estos son una primera aproximación para poder decir que creemos que hace lo que queremos que haga, pero normalmente se nos escapan excepciones que son las realmente peligrosas como que un servidor no este operativo y no muestra mensaje, repercutiendo en la calidad de la pagina ya que el usuario no recibe un mensaje que le explique el problema.

Para evitar esto vamos a pasar todos los casos de prueba con sus distintas opciones, además de poner la aplicación a prueba poniendo todas las posibles eventualidades que podrían poner los usuarios.

Por razones de extensión solo vamos a plasmar cuatro casos de todos ellos, los que he creído que son los mas importantes.

6.2.4.1 Loguear usuario

En un sistema de loguin el usuario se ha debido dar de alta en el sistema formalmente, una vez hecho esto puede acceder como usuario a la aplicación.

Para ello ponemos el nombre de usuario y contraseña en el lateral derecho de la aplicación:

The screenshot shows a login form with a green header 'Identificarse'. It contains two input fields: 'Nombre de usuario' with the text 'kikekan' and 'Contraseña' with masked characters. Below the fields is a blue 'Identificarse' button. At the bottom, there are two links: '¿Ha olvidado su contraseña?' and 'Registrarse'. A 'Foro' link is visible at the very bottom of the form area.

Al introducir un usuario y contraseña correctos accedemos a la aplicación como usuario registrado, creando una nueva entrada en la sesión.

The screenshot shows the user's profile page. The top right corner displays 'Usuario: kikekan'. The main content area on the left says 'Bienvenido kikekan'. On the right, there is a 'Carrito' section with a 'TOTAL::' field showing '0€' and a 'Mostrar carrito de la compra' link. Below that is a 'Buscador de productos' section with a 'Palabra a buscar' input field and a 'Buscar' button. At the bottom right, there is an 'Identificarse' section with links for 'Mi cuenta' and 'Cerrar sesión'. A 'Foro' link is also present at the bottom.

Haciendo las comprobaciones oportunas si ponemos un usuario o una contraseña erróneos, la aplicación devuelve el siguiente error.



6.2.4.2 Comprar producto

Estamos ante el caso de uso mas extenso de la aplicación. En primer lugar debemos tener hecha una selección de los productos que queremos comprar. Estos productos los veremos en la cesta de la barra lateral derecha. Una vez allí podemos pulsar el enlace "Mostrar carrito de la compra" para pasar a la página del carrito de la compra, como vemos en la siguiente captura.



Una vez allí pulsamos sobre el símbolo del dólar para pasar a hacer efectiva la compra pasando a la pantalla de factura donde tenemos un botón para realizar el pedido.

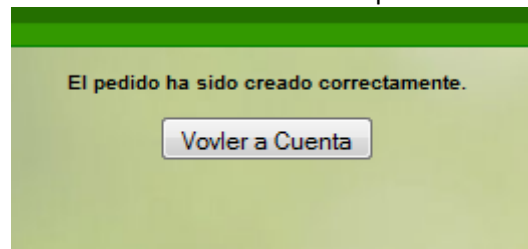
F A C T U R A

Kike Diego Oviedo
Av Torre del virrey nº 19

Factura #	000123
Fecha	16-09-2012
A pagar	31.73 €

Producto	Precio unidad	Cantidad	Total
ACHICORIA	1	4	4€
PEREJIL GIGANTE DE NAPOLES	2.95	4	11.8€
EUROPEAN ALDER	2.49	4	9.96€
FALSO ARBOL TE ROSA 25 semillas	1.99	3	5.97€
Total			31.73 €

Una vez realizado el pedido el sistema confirma la compra.



Si volvemos a la cuenta podemos observar en el historial que tenemos el pedido que hemos realizado.

Pedidos

IdPedido	Fecha	Ver
1	2012-09-09	
11	2012-09-09	
12	2012-09-15	
13	2012-09-15	
14	2012-09-16	

Total de Pedidos: 5

6.2.4.2 Cambiar contraseña.

Para esta prueba debemos estar primero logueado en el sistema. Una vez hecho esto vamos a nuestra cuenta y allí pulsamos cambiar contraseña. La página nos mostrara un formulario

donde pondremos la contraseña antigua, la nueva, y la repetición de la nueva, como seguridad por si el usuario se confunde al poner por primera vez la contraseña. El formulario quedaría así.



Cambiar Contraseña

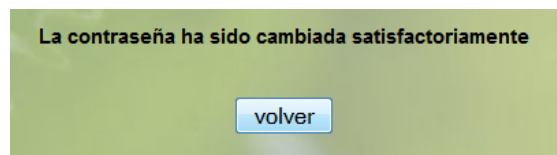
Antigua Contraseña:
.....

Nueva Contraseña:
.....

Repita Contraseña:
.....

Guardar

Una vez pulsado el botón guardar si todos los campos están correctos el sistema sacara un mensaje diciendo que efectivamente se ha cambiado la contraseña, como observamos en la siguiente captura.

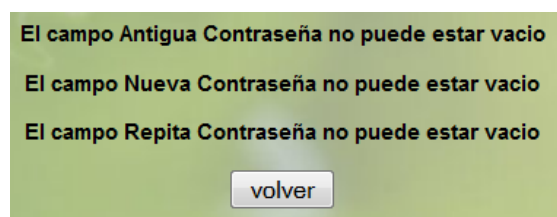


La contraseña ha sido cambiada satisfactoriamente

volver

Vamos a probar los casos excepcionales como dejar algún campo vacío, poner mal la contraseña antigua y no poner las dos contraseñas nuevas iguales.

En la siguiente prueba dejaremos todos los campos vacíos para que nos sirva como prueba de si los dejásemos uno a uno. Así comprobamos que efectivamente funciona la comprobación de texto en los campos. La captura es la siguiente.



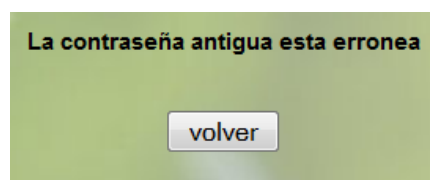
El campo Antigua Contraseña no puede estar vacío

El campo Nueva Contraseña no puede estar vacío

El campo Repita Contraseña no puede estar vacío

volver

Si ponemos mal la antigua contraseña nos pondrá el siguiente mensaje.



La contraseña antigua esta erronea

volver

Si ponemos diferentes las dos contraseñas nuevas el sistema dará el siguiente mensaje.

Las contraseñas nuevas no se corresponden

volver

6.2.4.2 Cambiar datos de contacto

Para poder cambiar los datos del usuario se necesita estar loggeado en el sistema. Una vez hecho esto podemos acceder al cambio de datos desde la página de nuestra cuenta. Allí pulsaremos en “Administrar datos personales” y cambiaremos los datos que nos interesen cambiar. Una vez hecho esto el sistema nos pondrá un mensaje como que ha cambiado los datos. Luego podremos volver a verlos en nuestra cuenta y otra vez “Administrar datos personales”.

En primer lugar cambiamos los datos que quedamos modificar.

Modificar Datos Usuario

Usuario:

E-mail:

Nombre:

Apellidos:

Dirección:

Guardar

Una vez que pulsamos guardar el sistema da el siguiente mensaje.

Los datos han sido modificados satisfactoriamente

volver

Para comprobar que el sistema reconoce que se haya dejado un campo sin rellenar dejaremos todos los campos que podamos, el de nombre de usuario no nos deja ya que esta fijo, vacios comprobando que el sistema lo reconoce. Esta es la captura de pantalla.

El campo email no puede estar vacio

El campo nombre no puede estar vacio

El campo apellidos no puede estar vacio

El campo direccion no puede estar vacio

Vovler

7. Conclusiones

Una vez hecho el proyecto podemos hacer una valoración clara de lo que se ha conseguido. Respecto a la productividad de lo aprendido he de decir que he conseguido mas de lo que me esperaba, no creía que los conocimientos en HTML, AJAX, php y mysql fueran a aumentar tanto. La verdad es que creía que me iba a costar mas el aprender los lenguajes que he comentado.

Respecto a CSS creo que hay que dedicarle especial atención. En primer lugar es algo a lo que no te enseñan en la carrera, al estar acostumbrado a programar, pasas a poner estilos a DIV y te resulta complicado. En concreto tengo que hacer especial mención al poder centrar la página y que al dar zoom mas y menos siempre se quedase en el centro. Para poder manejarlo un poco me costó mucho tiempo y no he indagado a penas nada. De todas maneras ahora me manejo con soltura y cualquier hoja de estilo que tenga en mis manos la podre moldear a mi gusto, siempre y cuando tenga tiempo para ello.

Por otro lado esta la satisfacción de haber realizado un proyecto personal que se puso como meta al principio del curso. Tuvo sus altibajos, porque no sabía si lo iba a acabar o no, pero al final se ha conseguido y el resultado y estoy muy contento

Hay que decir que el proyecto está abierto a muchas mejoras, empezando por la parte administrador. Poder controlar todos los productos y los usuarios desde la aplicación hacer que la pagina se vuelva totalmente mantenible por el cliente, lo que le da un valor mucho mayor.

También sería interesante hacer uso de JQuery ya que da resultados muy profesionales con apenas código lo que hacer que sea especialmente interesante.

Por último hacer una mención a la propia memoria que me ha ayudado bastante a realizarme como persona trabajadora, al ser algo a lo que le tienes que dedicar horas y de las cuales muchas veces no escribes ya que tienes que hacer un proceso de investigación que requiere de mucho tiempo.

Dicho todo esto el balance es tremendamente positivo y una experiencia para recordar toda la vida.

8. Bibliografía

En esta sección solo podre enlaces a páginas web ya que es de donde he recogido el material para hacer el proyecto.

- [1] <http://es.wikipedia.org/wiki/Wikipedia:Portada/>
- [2] <http://www.programacionweb.net/>
- [3] <http://www.php.net/>
- [4] <http://www.javascriptsource.com/>
- [5] <http://www.librosweb.es/css/>
- [6] <http://validator.w3.org/>
- [7] <http://oness.sourceforge.net/proyecto/html/ch03s02.html/>
- [8] <http://www.tiendadesemillas.es/>
- [9] http://www.danielpecos.com/docs/mysql_postgres/x57.html/
- [10] http://php.ciberaula.com/articulo/introduccion_php/
- [11] <http://www.photoshopytutoriales.com/introduccion-al-css/comment-page-1/>
- [12] <http://es.wikipedia.org/wiki/AJAX>
- [13] <http://www.misrespuestas.com/que-es-html.html>
- [14] <http://juliansanchez6606gta.blogspot.com.es/2011/05/ruby-es-un-lenguaje-interpretado-de.html>
- [15] <http://www.alegsa.com.ar/Dic/wamp.php>
- [16] <http://es.wikipedia.org/wiki/PhpMyAdmin>
- [17] <http://es.wikipedia.org/wiki/Notepad%2B%2B>
- [18] <http://biblioteca.usalca.cl/html/servicios/Citas%20bibliograficas/guia.pdf>
- [19] <http://www.upv.es/entidades/ABDC/infoweb/bg/info/798082normalc.html>
- [20] <http://www.upv.es/entidades/ABDC/infoweb/bg/info/798082normalc.html>

Durante el proyecto me he estado ayudando de otros proyectos hechos por alumnos que fueron dirigidos por Sergio Sáez Barona, es decir, nuestro director de proyecto. Estas son las direcciones donde puedes encontrarlas en riunet:

- <http://riunet.upv.es/handle/10251/9110>
- <http://riunet.upv.es/handle/10251/8634>
- <http://riunet.upv.es/handle/10251/8736>
- <http://riunet.upv.es/handle/10251/11166>
- <http://riunet.upv.es/handle/10251/15249>
- <http://riunet.upv.es/handle/10251/16952>
- <http://riunet.upv.es/handle/10251/10028>
- <http://riunet.upv.es/handle/10251/9110>
- <http://riunet.upv.es/handle/10251/8634>
- <http://riunet.upv.es/handle/10251/9120>