

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA DE COMPUTADORES

WEB AUTOMATIZADA PARA EL PAGO COLABORATIVO
AUTOMATIC WEB FOR COLLABORATIVE PAYMENT

Realizado por
Jose Manuel Martín Calderón

Tutorizado por
María del Carmen Aranda Garrido
Carlos Caballero González

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, Septiembre del 2016

Fecha defensa:
El Secretario del Tribunal

Resumen

Este trabajo tiene como objetivo el desarrollo de una tienda online donde los usuarios puedan pagar de forma colaborativa una compra. Aunque podemos encontrar multitud de webs de compras ninguna implementa esta forma de pago entre varios usuarios.

Para ello, se ha realizado una metodología consistente en dividir el trabajo en diferentes fases. Comenzando por un estudio de viabilidad escogiendo las herramientas adecuadas. Pasando por las diferentes fases del ciclo de vida del software. Estas fases son: Análisis en profundidad, Diseño para cumplir los requerimientos, Codificación de la idea y finalmente la Implantación del proyecto.

Además, se han aprovechado las herramientas que suministra el sistema operativo para la automatización de tareas para la gestión de los procedimientos necesarios para la administración de la plataforma.

En definitiva se ha desarrollado una aplicación que cumpla la metodología empleada utilizando conocimientos adquiridos y a su vez otros aprendidos durante el desarrollo de este trabajo. Habiendo sido éste una tarea gratificante.

Palabras Clave: Tienda Online, Colaboración, Terminal de Pago, PHP, HTML

Abstract

The aim of this project is to develop an online store where users could collaborate to pay a purchase. The idea is due to the fact that you can find many shopping websites but they do not have this way of payment between several users.

To do that, we have made a methodology to divide the work in different phases. We have started with a feasibility study by choosing the right tools. Going through the different phases of the software life cycle. These phases are: Insights, designed to meet the requirements, coding of the idea and eventually implement the project.

Moreover, we have used the tools, which are provided by the operating system for automating tasks in the management of the necessary procedures for the administration of the platform.

To sum up, we have developed an application that executes the methodology using knowledge acquired. In addition, we have used the knowledge acquired during the development of this project. It has been a rewarding task.

Keywords: Online Store, Collaboration, Payment terminal, PHP, HTML

Índice

Índice de Imágenes	6
Capítulo 1: Introducción.....	8
1.1. Objetivos del proyecto	9
1.2. Estudio del Arte	9
1.3. Fases del Proyecto.....	11
1.4. Estructura de la Memoria	11
1.5. Componentes empleados	12
1.6. Trabajo Realizado	13
Capítulo 2: Estudio de Viabilidad.....	14
2.1. Alcance del Sistema.....	14
2.2. Análisis de los sistemas actuales	15
2.3. Requisitos del sistema	16
2.4. Valoración de las alternativas.....	17
2.5. Selección de la solución	23
Capítulo 3: Análisis del Sistema	24
3.1. Definición del sistema	24
3.2. Definición de los requisitos de funcionamiento.....	25
3.3. Casos de uso	26
3.4. Interfaces de usuario	31
3.5. Especificación del plan de pruebas	37
Capítulo 4: Diseño del Sistema	38
4.1. Arquitectura	38
4.2. Identificación de subsistemas.....	40
4.3. Revisión de casos de uso	40
4.4. Estructura de la base de datos.....	41
4.5. Análisis de componentes del sitio web.....	42
Capítulo 5: Desarrollo	43
5.1. Planificación de las actividades de desarrollo e integración de sistema	43
5.2. Desarrollo del proyecto	44
5.3. Documentación	50

Capítulo 6: Implantación	51
6.1. Automatización de tareas	51
6.2. Pruebas realizadas.....	52
Capítulo 7: Mantenimiento.....	58
Capítulo 8: Conclusiones.....	59
Capítulo 9: Bibliografía	61

Índice de Imágenes

Imagen 1: Esquema de las tiendas online actuales.....	15
Imagen 2: Uso de sistemas operativos como servidores webs.	19
Imagen 3: Uso de lenguajes de programación en los sitios webs	21
Imagen 4: Porcentajes de Servidores webs más usados	22
Imagen 5: Caso de Uso - Gestión de venta.....	26
Imagen 6: Caso de Uso - Confirmación del pedido	27
Imagen 7: Caso de Uso - Confirmación de pago.....	28
Imagen 8: Caso de Uso - Pago del pedido.....	28
Imagen 9: Caso de Uso - Confirmación del TPV	29
Imagen 10: Caso de uso - Administración.....	30
Imagen 11: Caso de Uso - Gestión de Eventos.....	30
Imagen 12: Prototipo - Página principal.....	31
Imagen 13: Prototipo - Estado del pedido	32
Imagen 14: Prototipo - Confirmación del pedido.....	33
Imagen 15: Prototipo - Confirmación de Email	34
Imagen 16: Detalle Prototipo - Añadir email	35
Imagen 17: Prototipo - Pagar pedido.....	35
Imagen 18: Prototipo - Pago Correcto	36
Imagen 19: Prototipo - Pago Incorrecto	37
Imagen 20: Arquitectura	39
Imagen 21: Subsistemas de Tienda Virtual	40
Imagen 22: Diagrama E/R	42
Imagen 23: Diagrama de Gantt	43
Imagen 24: Página principal de la aplicación.....	45
Imagen 25: Interfaz - Carrito de compras	45
Imagen 26: Interfaz - Envío de correos	46
Imagen 27: Interfaz - Confirmación de email.....	46
Imagen 28: Información recibida por email.....	47
Imagen 29: Interfaz - Confirmación de pago.....	47
Imagen 30: Proceso de pago en el Terminal.....	48

Imagen 31: Interfaz - Pago Correcto 1	48
Imagen 32: Interfaz - Pago Correcto 2	49
Imagen 33: Interfaz - Pago Incorrecto	49
Imagen 34: Cron del sistema.....	51
Imagen 35: Resultado del test del código.....	52
Imagen 36: Resultado del test CSS.....	52
Imagen 37: Detalle de advertencia CSS.....	53
Imagen 38: Tamaño pequeño de la aplicación	54
Imagen 39: Tamaño mediano de la aplicación	54
Imagen 40: Visualización en Internet Explorer	55
Imagen 41: Visualización en Mozilla Firefox.....	55
Imagen 42: Visualización en Google Chrome.....	55
Imagen 43: Visualización en Opera.....	56

Capítulo 1: Introducción

Actualmente el negocio de venta por internet tiene un gran auge y su utilización podemos considerar que ha tenido una gran repercusión con el paso de los años. Las ventajas de este tipo de venta son muy considerables y quedan reflejadas con respecto al llamado negocio tradicional.

En ese mismo sentido, recientemente encontramos un apogeo floreciente de una labor colaborativa entre usuarios de una plataforma online. Todos estos usuarios trabajan por un bien en común.

En referencia a lo comentado anteriormente, el eje de este trabajo sería unir las dos ideas mencionadas anteriormente. Crear una plataforma donde un usuario pueda pagar una compra y compartir el desembolso con varios usuarios.

A partir de esta idea, se ha examinado las diferentes webs que podemos encontrar en Internet para realizar compras de productos, podemos ver que ninguna de ellas ofrece el servicio mencionado anteriormente. Las compras en todas ellas podemos decir que son personales y en ningún momento se puede involucrar a terceros. Igualmente, en la mayoría la única relación entre ellos, es para opinar sobre una compra realizada; dando en su caso una valoración.

Así pues, el objetivo principal de este proyecto podemos considerarlo como el de analizar y realizar una plataforma web donde podamos realizar compras, para posteriormente poder efectuar el pago de ésta entre varios usuarios.

Así mismo, se analizarán las diferentes webs existentes para recabar información la cual pueda ser usada en nuestro proyecto, sobre todo en lo relacionado con la gestión de los productos y sus clientes.

A lo largo de esta memoria, se detallarán y desarrollarán todos los pasos necesarios para crear e implementar esta plataforma. En particular, se explicarán las diferentes fases por las que el software tiene que pasar, para que el resultado final sea una implementación adecuada a los objetivos planteados inicialmente.

Para ello, utilizaremos los conocimientos adquiridos durante el desarrollo del grado en ingeniería de computadores, los cuales propiciarán que el resultado sea factible; con respecto a los objetivos a alcanzar.

1.1. Objetivos del proyecto

Los objetivos del proyecto, después de un estudio pormenorizado de los requerimientos que necesitamos para nuestra aplicación, son los siguientes:

✓ Objetivos generales:

- Analizar diferentes webs de venta de artículos atendiendo a la gestión de los clientes y los artículos. Teniendo en cuenta los elementos del diseño e implementación de ellas.
- Realizar un análisis de las posibles herramientas que podemos utilizar para implementar la aplicación que nos atañe.
- Diseñar un sitio web para poder realizar una compra y que ésta se pueda compartir entre varios usuarios.
- Automatizar tareas para una administración correcta e idónea de la plataforma.
- Implementar los módulos necesarios para el correcto funcionamiento del sitio web que queremos diseñar, atendiendo a la usabilidad y accesibilidad de nuestro portal.

✓ Objetivos específicos:

- El comprador podrá seleccionar entre los diferentes artículos que oferta el portal web para su venta.
- El cliente deberá conocer el importe total del pedido que ha realizado, después de haber seleccionado los artículos que le ha interesado.
- El cliente que denominaremos anfitrión, seleccionará a los usuarios que compartirán el total del pedido que ha realizado el anfitrión.
- Gestionar el envío de correos de notificación a los clientes que comparten la compra realizada.
- Gestionar el pedido mediante una pasarela de pago atendiendo a los parámetros establecidos por dicha plataforma y atendiendo a la seguridad en todo momento para el cliente.
- Comprobar el estado de los pedidos cada cierto tiempo para gestión adecuada.

1.2. Estudio del Arte

Actualmente el uso de internet y las posibilidades que nos ofrece van en aumento de manera exponencial. Podemos encontrar todo tipo de elementos webs como pueden ser wikis, blogs, redes sociales, mundos virtuales, etc. Esto hace que las personas estén cada vez más familiarizadas con internet y les resulte muy sencillo su uso. Esto implica por un lado que cada vez es más cómodo y habitual realizar transacciones en la red. Por otro lado encontramos que algunos de los elementos mencionados

anteriormente realizan un trabajo colaborativo entre distintas personas para un bien común.

En la actualidad, la presencia de tiendas en internet ha crecido muy rápido. Ello implica que hoy en día, es difícil encontrar a alguien que no haya realizado alguna compra en una tienda online. Este tipo de establecimientos ofrecen una serie de ventajas con respecto a las llamadas tiendas tradicionales como puede ser la amplitud de horario, la eliminación de colas, la comodidad en la compra y probablemente unos precios más competitivos al no necesitar un local o infraestructura física.

También podemos encontrar desde la aparición de Internet, y más concretamente desde la aparición de la llamada Web 2.0 [1], la existencia de un trabajo colaborativo entre varios interesados.

Con este proyecto, intentamos unir las dos ideas mencionadas en los dos párrafos anteriores. Por un lado una web que será una tienda virtual y por otro lado dentro de ella tener un sistema para poder realizar un pago de la compra entre varios usuarios implicados en ésta.

La idea parte de que muchas de estas compras son para regalos, en los cuales intervienen posteriormente más personas en la repartición del pago de éste. Sin embargo en las plataformas actuales solo encontramos la posibilidad de realizarlo directamente por una única persona. Siendo posteriormente y entre los participantes de dar el regalo, la repartición de estos gastos.

Del mismo modo encontramos productos que van a ser utilizados por varias personas y que su precio es relativamente elevado, por lo que se compartirán el gasto de la compra. En este caso la herramienta que estamos describiendo es de gran utilidad por las posibilidades que nos ofrece.

Para ello existen múltiples soluciones muy completas de software libre para la programación de tiendas de tipo online y que pueden ofrecer crear una tienda con facilidad y con coste económico con respecto a un desarrollo a medida. Las más conocidas son Magento, OsCommerce o Prestashop. Todas ellas ofrecen la posibilidad de crear una tienda online básica en la cual puedes incluir un catálogo de productos, una cesta de la compra y diferentes formas de pago y envío.

Sin embargo, en todas estas herramientas mencionadas anteriormente es complicado cambiar la forma de pago para ingresar un elemento en este caso un formulario donde se puedan incluir varios usuarios en la compra. Y que para cada uno se le puede destinar un terminal de pago.

1.3. Fases del Proyecto

Con el objetivo de mejorar la gestión del proyecto, su control y lograr que se acerque lo máximo a los objetivos planteados en el proyecto, repartiremos éste en varias fases, las cuales son las siguientes:

- Estudio de viabilidad: para llevar a cabo este proyecto se verán las posibilidades que nos ofrecen las herramientas que nos encontramos en el mercado. En función de ello escogeremos la solución más adecuada para lograr que al final el producto que obtengamos muestre los elementos que deseamos que tenga.
- Fase de Análisis: en esta fase describiremos con más profundidad la solución seleccionada en el estudio de viabilidad anterior, atendiendo a los requerimientos.
- Fase de diseño: desarrollo de un diseño del proyecto que permite cumplir los requisitos del cliente en cuanto a términos de calidad y plazos de tiempo. Para ellos debemos obtener los modelos y especificaciones basándonos en la información obtenida en la fase de análisis. En esta fase también definiremos una estrategia con la cual realizaremos las pruebas pertinentes para el correcto funcionamiento.
- Fase de codificación: en esta fase se realizará el desarrollo de acuerdo al diseño realizado en la fase diseño, cumpliendo los estándares pertinentes para la webs.
- Fase de implantación: se realizará un despliegue del código resultante en la fase de codificación. Realizando las pruebas oportunas diseñadas en fases anteriores.

1.4. Estructura de la Memoria

En este documento se desarrollaran las diferentes fases que hemos definido en el punto anterior. Cada fase estará definida como un capítulo y el detalle de cada fase será un subcapítulo del mismo.

También completaremos la información general de este trabajo añadiendo al final del mismo nuestras propias conclusiones, hitos conseguidos, posibilidades de ampliación y referencias bibliográficas.

En consecuencia, la organización de la memoria del trabajo fin de grado queda definida de la siguiente manera.

En el capítulo 1 se hace un análisis introductorio del proyecto, explicando los objetivos a alcanzar, así como las fases y componentes utilizados. Finalmente se expondrá los problemas encontrados y las soluciones aportadas.

En el capítulo 2 se desarrolla el esquema general de funcionamiento del sitio web. También se hará un estudio de viabilidad para analizar el problema a resolver y plantear las posibles soluciones. En último lugar se mostrará la solución que se considera más adecuada para la ejecución del trabajo.

En el capítulo 3 se realiza un análisis detallando los requisitos del sistema en mayor grado, para ello se define la arquitectura del sistema, los estándares y normas establecidos. Posteriormente se muestra los casos de uso y las interfaces más importantes de este trabajo.

En el capítulo 4 se explica el diseño del interfaz realizado, exponiendo los principios de usabilidad del sitio. Se identifican los subsistemas que tiene el sitio, así como los componentes del proyecto.

En el capítulo 5 se analizan las tareas que se han llevado a cabo durante el desarrollo e integración del proyecto.

En el capítulo 6 se desarrollan los resultados de los ensayos y pruebas llevados a cabo sobre el conjunto del sistema.

En el capítulo 7 se explica las pautas necesarias para un mantenimiento correcto de la aplicación.

En el capítulo 8 se exponen las conclusiones en el desarrollo del trabajo. También qué objetivos se han cumplido y qué elementos se podrían mejorar en un futuro.

Finalmente, se dejará una mención de la bibliografía usada durante el desarrollo de este proyecto.

1.5. Componentes empleados

Los distintos componentes del sistema tienen que tener como requisito principal que sean libres, para que el proyecto tenga el menor coste posible. Así pues las herramientas escogidas son las siguientes.

- Sistema Operativo: Linux Ubuntu 14.04
- Servidor web: Apache
- Sistema Gestor de Bases de Datos: MySQL
- Lenguajes de Programación: PHP, HTML y JavaScript
- Navegador: Firefox, Chrome y Opera
- Depuración de Código: Firebug y Web Developer
- Editor entorno desarrollo de código: Sublime Text 2
- Realización de diagramas UML: GenMyModel
- Modificación de imágenes e iconos: GIMP
- Realización de la memoria y demás documentación: LibreOffice
- Realización de la planificación: GanttProject

1.6. Trabajo Realizado

El trabajo realizado a grandes rasgos y que se detallará durante esta memoria ha sido el siguiente.

- Definir los requisitos necesarios y estudiar la mejor solución.
- Realización el análisis y diseño de una solución que se adapta a los requisitos definidos anteriormente.
- Implementación de esta solución en los parámetros establecidos.
- Ejecución de las pruebas pertinentes en el entorno.
- Aceptación de las pruebas y puesta en marcha de la solución creada para este propósito.
- Creación de la memoria que servirá como documentación de consulta para este trabajo.

Capítulo 2: Estudio de Viabilidad

En este capítulo elegiremos una solución que se adapte al conjunto de necesidades planteadas. Para escoger una solución apropiada habrá que tener en cuenta las posibles situaciones y los requisitos planteados para nuestro proyecto.

2.1. Alcance del Sistema

El alcance del portal web aporta una plataforma para poder comprar unos productos a partir de una lista, para posteriormente poder hacer un pago automatizado entre varios clientes. Para ello las necesidades que se nos podrían plantear son las siguientes.

- Implantar una sección donde un cliente, el cual podremos llamar anfitrión, pueda seleccionar diferentes productos del catálogo que suministra el sitio web.
- Ante la posibilidad de diferentes clientes del portal; el lugar debe ser muy sencillo e intuitivo, donde el usuario esté guiado totalmente.
- Se le informará al cliente del coste total del pedido.
- El cliente anfitrión completará los correos electrónicos de los demás usuarios involucrados en acompañar el pago.
- Se le enviará un correo electrónico y se detallará a cada cliente el coste que tendrá su parte correspondiente del pago.
- Se suministrará a cada cliente un acceso a una pasarela de pago virtual para que puede completar el desembolso, mediante el pago con tarjeta de crédito.
- La aplicación comprobará el estado de los pedidos regularmente mediante procedimientos.

A partir de esta información podemos definir los objetivos que debemos cubrir con el sistema, desde varios puntos de vista.

Por un lado, tenemos el punto de vista económico. En este caso el coste deberá ser cero o en su caso lo más económico posible. Para ello tendremos que utilizar herramientas de software libre, por eso, en lo que se refiere a herramientas, el coste económico es cero.

En lo que respecta al coste de desarrollo, lo que afectará más que el valor económico, será la dificultad de aprendizaje de los diferentes elementos que tendremos que desarrollar, por lo que el factor a tener en cuenta será la curva de aprendizaje.

Desde el punto de vista operativo, el código que se realice para el portal deberá ser funcional, intuitivo y rápido para que a los clientes se les pueda facilitar la utilización del conjunto del sistema.

2.2. Análisis de los sistemas actuales

Actualmente encontramos una multitud de portales dedicados a la venta. Estos incluyen un sinnúmero de elementos los cuales hacen al usuario más fácil su compra. Además podemos encontrar desde lugares que se dedican a la venta exclusiva de productos específicos hasta webs que venden todo tipo de productos, siendo estas últimas las más frecuentes y las más conocidas.

Por otro lado algunos de ellos, los menos sofisticados, usan para su implementación un gestor de contenidos específico para la gestión y mantenimiento de tiendas virtuales; resultando un trabajo más sencillo de realizar.

En cualquiera de los casos mencionados anteriormente, tanto para un caso como para el otro, el esquema principal de los negocios actuales, podemos considerar que queda representado en el esquema representado en la imagen 1.



Imagen 1: Esquema de las tiendas online actuales

Sin embargo, ninguna implementa un posible sistema de pago entre varios usuarios o clientes, teniendo únicamente la posibilidad de pagos individuales por una sola persona. Por ello la idea de este proyecto es cambiar la relación entre el usuario, la tienda virtual y la pasarela de pago, para gestionar el objetivo de este trabajo.

2.3. Requisitos del sistema

Tras realizar el estudio de la situación actual y de las necesidades planteadas ya podemos enumerar los requisitos iniciales del sistema.

Desde el punto de vista de los requisitos técnicos se tendrán en cuenta los siguientes aspectos que afectarán al diseño del sistema y a los estándares que se tienen que usar.

- Se utilizará software libre para el desarrollo del proyecto, para ello tras el estudio de varios editores se buscará un programa el cual su licencia sea la mencionada anteriormente.
- La tienda on-line debe ser compatible con los principales navegadores web [2] que existen en la actualidad, en este caso Chrome, Firefox, Internet Explorer, Opera y Safari.
- El desarrollo que se implemente deberá cumplir los estándares de *World Wide Web Consortium* [3] en cuanto a la validación del código HTML y CSS empleados.
- La web cumplirá las condiciones de accesibilidad según WC3, cumpliendo el estándar WAI [4].
- Además tendremos que conseguir que el portal tenga una usabilidad acorde, para que la herramienta sea fácil de usar.

Desde el punto de vista de los requisitos legales, habrá que tener en cuenta para la realización del proyecto, los siguientes elementos.

- Los productos utilizados, además del producto desarrollado, deberá tener licencia GPL [5] o compatible para la libre modificación de éste, así como su estudio o copia de su código.
- Otro aspecto a tener en cuenta será que, como en el proyecto se almacenará y utilizará información sensible de carácter personal, se deberá respetar y tener en cuenta en todo momento lo que la Ley de Protección de Datos dice al respecto sobre esta materia.

En cuanto a lo que se refiere a los requisitos funcionales, se verificará los siguientes puntos.

- La tienda online una vez implementada deberá funcionar de una manera rápida y eficaz.
- La información estará estructurada de forma clara y su consulta será intuitiva para el cliente. Además cada zona estará claramente definida.
- En la página principal se da una visión de lo que ofrece el portal, así como sus características.
- Se incorporará una búsqueda de los elementos de la tienda para que el usuario le resulte más fácil encontrar lo que está buscando.

- El proceso de compra por parte del usuario es básico para la satisfacción de éste. Por tanto, dicho proceso deberá ser lo más fácil y ágil; en la medida de lo posible.
- El proceso deberá realizarse en el menor número de pantallas posibles. Es decir, se intentarán realizar los menos saltos posibles para que los clientes le resulte más fácil el manejo de la aplicación.
- El proceso deberá ser intuitivo, incluso podría ser guiado para evitar que el usuario pueda realizar el proceso con éxito.
- El usuario podrá eliminar un producto que haya seleccionado anteriormente en la compra, antes de solicitar el pedido.
- En el proceso de confirmación del pedido, el usuario podrá confirmar los correos electrónicos de los diferentes clientes que pagaran su parte proporcional del pedido.
- El sistema enviará un correo a cada uno de ellos para que por un lado conozca el pedido y su parte proporcional del coste de éste. Y por otro lado un enlace para pagar el pedido.
- Para cada cliente el sistema abrirá un terminal de pago virtual con su respectiva cuota de pago.
- El sistema comprobará que todos los clientes han pagado su parte proporcional para enviar el pedido.
- El sistema gestionará automáticamente cualquier cambio en los pedidos, realizando los oportunos movimientos en el portal.

2.4. Valoración de las alternativas

Teniendo en cuenta que tenemos la premisa de usar software libre, podemos estudiar los diferentes sistemas de gestión de contenidos gratuitos (CMS) existentes en el mercado para ver si sería viable los que nos propone con lo que nosotros queremos realizar.

Basándonos en el estudio realizado por 4webs.es [6] podemos analizar los diferentes gestores de contenidos para tiendas virtuales.

- **OsCommerce** [7]: fue uno de los primeros en aparecer en el campo de la creación de tiendas online. Posee multitud de funcionalidades muy completas que van desde la gestión de métodos de pago y gestión de envíos. Además permite integrar varios idiomas. La instalación inicial es muy básica y requieren de muchos módulos para comenzar a operar.

Asimismo, el proyecto se encuentra estacando por lo que actualmente tiene muchos problemas de seguridad.

- **PrestaShop** [8]: es uno de los gestores más recientes del mercado pero no por ello deja de ser potente y sencillo de utilizar. Es muy fácil de instalar y

posee un panel de administración muy completo e intuitivo. Es una de las aplicaciones de comercio electrónico más rápido del mercado y con un diseño atractivo.

- **Magento** [9] : también es un proyecto joven, sin embargo es muy utilizado por muchos programadores. Posee muchas características y utilidades avanzadas. Además posee un diseño muy cómodo e intuitivo.

Asimismo, todas las herramientas mencionadas anteriormente tiene un gran punto a favor; la existencia de una gran comunidad que apoya a cada proyecto, ya sea mediante foros o wikis. En particular puede ser de gran ayuda en la búsqueda de una solución a un problema que tengamos.

También hay que tener en cuenta, la posibilidad de no usar ningún CMS por la curva de aprendizaje que se necesita para conocer como instalar un módulo para las intenciones que queremos realizar, pudiendo ser de gran dificultad acceder a cambiar el sistema para una gestión de pago diferente.

En cuanto a la elección del Sistema Operativo, si vamos a seleccionar uno basado en GNU y teniendo en cuenta los datos suministrados por TOP500.org [10], que lleva estadísticas sobre las 500 principales supercomputadoras del mundo: 469 usan una distribución basada en GNU/Linux, 20 Unix, 7 mezclas, 1 BSD y 3 Windows. Vemos claramente y con diferencia que los sistemas basados en GNU/Linux son los más usados en el mundo por su gran rendimiento.

Por otro lado si nos basamos en las estadísticas de los sistemas operativos GNU/Linux suministrada por w3techs.com [11], encontramos que el más utilizado es Debian, aunque casi con el mismo porcentaje que Ubuntu. Seguido muy de cerca de CentOS. Las demás versiones vemos que quedan muy lejos de éstas. En la imagen 2 podemos verlo reflejado.

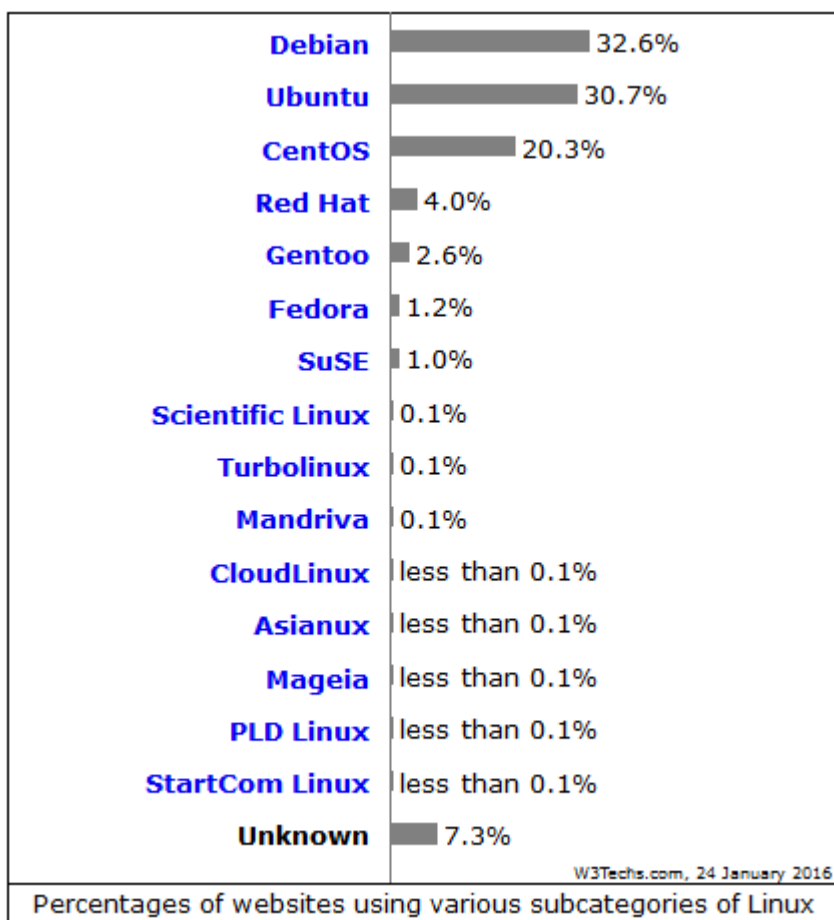


Imagen 2: Uso de sistemas operativos como servidores Webs.

A continuación pasaremos a analizar los tres sistemas operativos más populares mencionados anteriormente.

- **Debian** es un sistema operativo que es desarrollado por muchos voluntarios de todo el mundo. La versión actual para servidores es la 8.0 conocida como “Jessie”, la cual incluye las versiones para Apache, PHP, MySQL y PostgreSQL.

Un elemento que avala a este Sistema Operativo es que una distribución no es publicada hasta que sea una versión estable y se hayan solucionado todos los errores de la versión anterior. Para ello se realizan y superan unos test de fallos; y hasta que la versión no esté totalmente libre de errores no es publicada como estable. Por ello Debian es una de las distribuciones de Linux más robustas, seguras y potentes que podemos encontrar.

Otra característica es que tiene un proceso de instalación algo complicado, además de algunas incompatibilidades con el hardware de las maquinas.

Posee un grupo de herramientas para facilitar el proceso de instalación y actualización del software.

Aunque tiene varios entornos gráficos como GNOME, KDE o Xfce; se puede instalar sin ningún entorno gráfico, haciendo menos pesado el contenido del servidor. Por otro lado, el soporte de una versión tiene una duración de diez años.

- **Ubuntu:** es una versión que se ramificó de Debian y que actualmente está teniendo un gran éxito. Desarrollado por la empresa *Canonical Ltd.* La idea de este sistema operativo es hacer una versión de Linux fácil de usar para usuarios no muy experimentados.

Tiene una gran compatibilidad con la mayoría del hardware de las máquinas actuales, así como de máquinas más antiguas. La versión actual de Ubuntu Server 14.04.3 posee los paquetes de Apache, PHP y MySQL

La desventaja que puede tener es que no es tan robusta como puede ser Debian. Por otro lado cada seis meses hay una versión de Ubuntu y el soporte de cada versión tiene una duración de cinco años.

- **CentOS:** es una bifurcación de la distribución Red Hat Enterprise Linux compilado por voluntarios a partir del código fuente publicado. Por ello podemos pensar que ofrece un software de clase empresarial de manera gratuita.

Se define por ser robusto, estable y fácil de instalar y utilizar. Es una versión de Linux bastante estable y adecuada para servidores webs. Como inconveniente tiene su dificultad de actualizar a una nueva versión.

La versión actual es 7.1511 y trae servicios para PHP, Apache, MySQL y PostreSQL.

En lo relacionado con la elección de los servicios webs, lo más fácil y útil es usar el paquete LAMP, que es una distribución que posee los servicios básicos para funcionar un sitio web: Apache, MySQL y PHP.

Para ver este elemento queda claro en las estadísticas suministradas por w3techs.com. Por un lado PHP es el lenguaje de programación más usado en los sitios webs. La imagen 3 muestra las estadísticas de los lenguajes de programación usados.

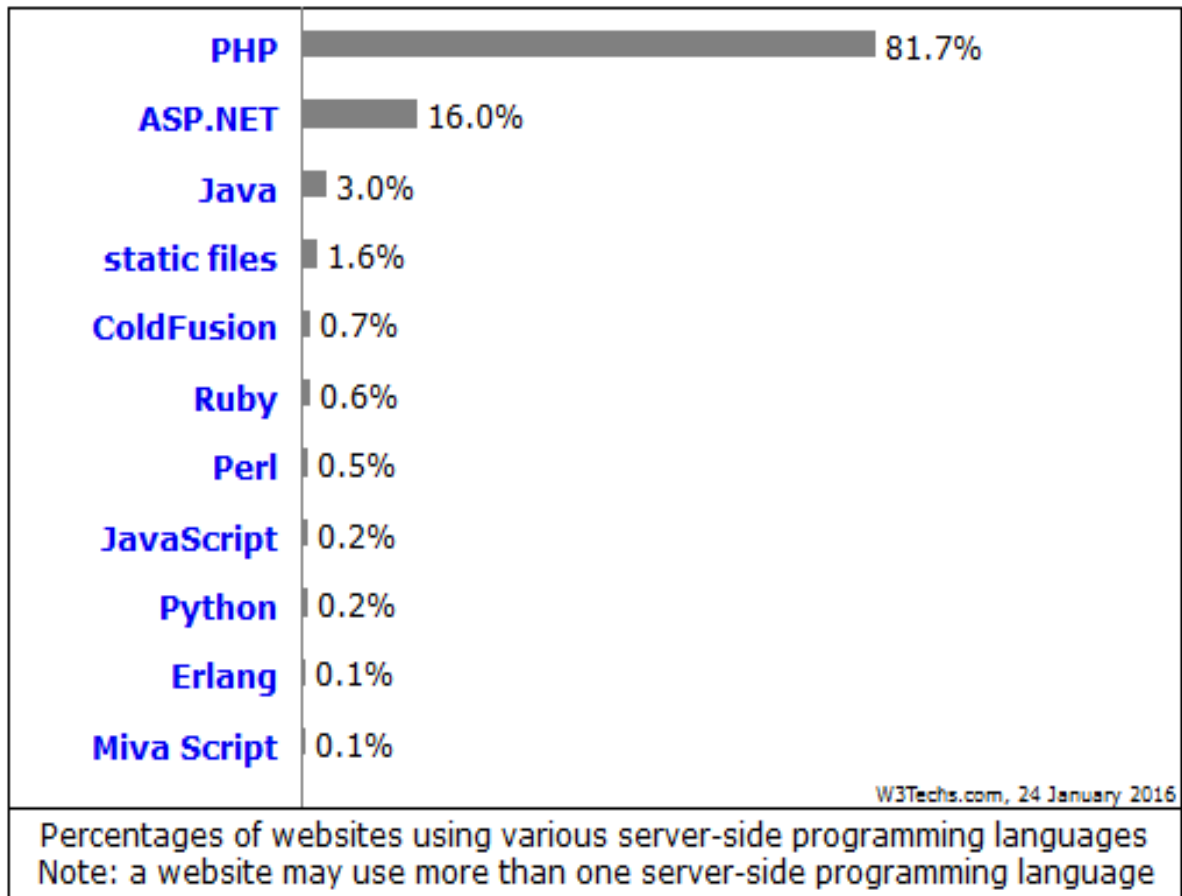


Imagen 3: Uso de lenguajes de programación en los sitios webs

Siguiendo con las estadísticas, en cuanto al servidor web más usado es Apache. Básicamente por ser un servidor web de código abierto que está implementado para multitud de sistemas operativos. Actualmente la última versión estable es la 2.4.18.

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la fundación “Apache Software”. Su mayor característica es que es altamente configurable. Aunque también cabe destacar su robustez, su seguridad y su rendimiento.

Por otro lado, a gran distancia nos encontramos ASP.NET, un servidor que suministra Microsoft mediante los diferentes módulos que posee los Windows Server y más concretamente por medio del servidor IIS.

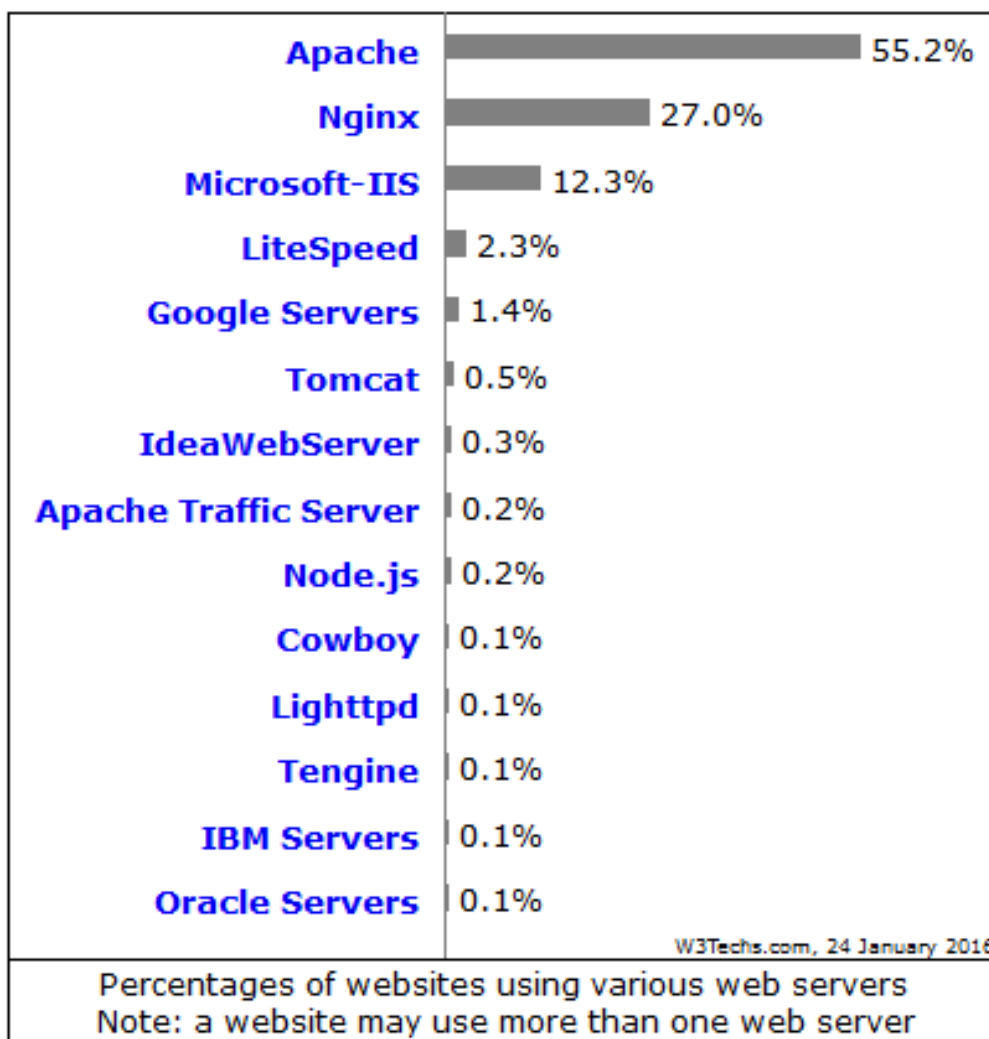


Imagen 4: Porcentajes de Servidores webs más usados

Como vemos en la imagen 4, el servidor web más utilizado es Apache, seguido de lejos por Nginx. Aunque destacaremos que en los últimos tiempos está subiendo bastante y muchas empresas importantes lo están incorporando como servidor web de sus aplicaciones.

Por último en lo referente a la base de datos. Podemos utilizar por un lado MySQL. Es sin duda el sistema más usado y mejor documentado. La inmensa mayoría de los scripts y aplicaciones que podemos encontrar están desarrollados para trabajar en una base de datos MySQL. Actualmente la versión que es estable es la 5.7.9.

Por otro lado podíamos utilizar la otra alternativa libre a MySQL, es cual es PostgreSQL. Teniendo un nivel de escalabilidad y versatilidad muy similar. Cabe destacar la robustez que ofrece. Tiene por detrás una dilatada trayectoria de desarrollo comunitario y su popularidad es creciente.

2.5. Selección de la solución

En base a la información recopilada anteriormente sobre gestores, sistemas operativos y diferentes elementos necesarios para la creación de sitio web; se considera que la mejor solución realizar el sistema de pago colaborativo sería la siguiente.

Por un lado, se ha decidido no usar ningún gestor de contenidos por la curva de aprendizaje que se necesita para conocer el funcionamiento del CMS y poder modificarlo a la hora de realizar el pago de un pedido. Ya que éste será nuestro principal propósito, modificar la forma de pago con un carácter colaborativo. Sin embargo los gestores de contenidos se encuentran demasiados modulados, haciendo esta labor más complicada.

Aunque a la hora de crear el módulo de la tienda puede ser más lento de crear al empezar desde cero, se compensará con conocer todo el código realizado y poder intercalar los elementos necesarios para acometer el objetivo de nuestro proyecto.

Por otro lado, se ha decidido utilizar un servidor Linux con distribución Ubuntu, por tener un gran auge y no tener un coste. Además, como hemos visto anteriormente, las distribuciones Linux son las más utilizadas.

A su vez se instalará el paquete LAMP el cual instalará las versiones de Apache, PHP y MySQL; abarcando de esta manera todas las herramientas necesarias para realizar el proyecto.

Capítulo 3: Análisis del Sistema

Una vez seleccionada la solución pasaremos a detallar de una manera más específica el diseño para evitar la posibilidad de errores en las fases posteriores de la vida útil del proyecto.

3.1. Definición del sistema

El sitio web deberá cumplir los requisitos mencionados en el capítulo anterior y que se detallan a continuación.

Desde el punto de vista técnico cumplirá los siguientes aspectos.

- Se albergará el sitio web en sistema operativo basado en GNU/Linux, en este caso Ubuntu 14.04.
- Se instalará el conjunto de herramientas denominado LAMP, que incorpora Apache, el servidor web; MySQL como gestor de base de datos y PHP como lenguaje de programación. Teniendo en este caso la infraestructura del servidor montada.
- En cuanto a normas y estándares a cumplir, el código tendrá que cumplir las normas y estándares establecidos por el *World Wide Web Consortium*.
- El sitio web cumplirá el estándar HTML 5 y CSS3 suministrados por W3C en cuanto la validación del código empleados.
- Cumplirá el estándar WAI para determinar si la web cumple con las condiciones de accesibilidad según W3C
- El portal online deberá funcionar correctamente en las versiones de navegadores más usuales que existen en el mercado, como son Firefox, Chrome, Internet Explorer, Opera y Safari.
- Se tendrá en cuenta la Ley de Protección de Datos de acuerdo a la Ley Orgánica 15/1999 de 13 de diciembre.
- En la elección de algún componente se buscará que sea gratuito y que no merme la calidad del producto.

Con respecto a los requisitos operativos, la plataforma cumplirá los siguientes aspectos.

- Respecto al interfaz de usuario deber ser sencillo e intuitivo, siguiendo un diseño coherente.
- La gestión de la información será correcta y estructurada.
- En la página principal se detalla con claridad, la misión del sitio. Además dará una visión de lo que ofrece, así como sus características y como está organizada.
- El contenido se dispondrá de forma que ofrezca credibilidad y confianza para el usuario.

- El portal dispondrá de un catálogo de productos para que el usuario pueda elegir entre ellos.
- El usuario encontrará un sistema de búsqueda de productos para que la compra le resulta más fácil.
- El cliente podrá seleccionar todos los productos que quiera y en la cantidad que desee.
- La plataforma dispondrá de alguna herramienta para modificar o eliminar algún producto de los seleccionados en la compra.
- El usuario podrá incluir más usuarios en la compra, introduciendo el correo electrónico de cada uno de ellos.
- El sistema enviará a cada usuario un detalle del pedido y la parte proporcional que pagará. Además se le enviará un enlace para realizar el pago correspondiente a la compra.
- La plataforma proveerá de un acceso a plataforma de pago virtual para poder pagar la compra.
- El sistema gestionará que los usuarios han pagado su respectiva parte para avisar a los componentes de la compra que el pago se ha realizado.
- La plataforma administrará adecuadamente los eventos que se producen durante la gestión de un pedido.

3.2. Definición de los requisitos de funcionamiento

Ajustándose a lo establecido en los apartados de motivación y objetivos del proyecto, la plataforma deberá cumplir unos requisitos de funcionamiento, los cuales detallamos a continuación.

La aplicación contará a grandes rasgos con tres formularios. En el primero el usuario podrá elegir los artículos que vea conveniente, formalizando la compra resultante de esta acción.

El segundo, será la parte inicial a la que está dirigida la motivación de este trabajo y consistirá en solicitar al usuario los distintos correos electrónicos de los diferentes clientes que forman la compra en común.

Por último, el tercer formulario, recepcionará el enlace enviado anteriormente a cada correo electrónico respectivamente, para confirmar el pago parcial y pasar el testigo a la pasarela de pago virtual.

3.3. Casos de uso

Con el diagrama de casos de uso se representa la forma como los diferentes actores actúan con el sistema en desarrollo. Partimos de los diferentes agentes y de las diferentes acciones en el sistema que serían los casos de uso. A continuación mostraremos los diagramas de casos de uso para definir el conjunto de funcionalidades que cumpliría a partir de los requerimientos establecidos [12].

1. Gestión de venta.

El usuario podrá realizar diferentes acciones para que la compra sea lo más sencilla posible. Para ello el usuario podrá realizar las siguientes acciones.

- Seleccionar los productos.
- Eliminar productos.
- Ver carrito.
- Buscar producto.

Lo podemos ver en el diagrama de la imagen 5.

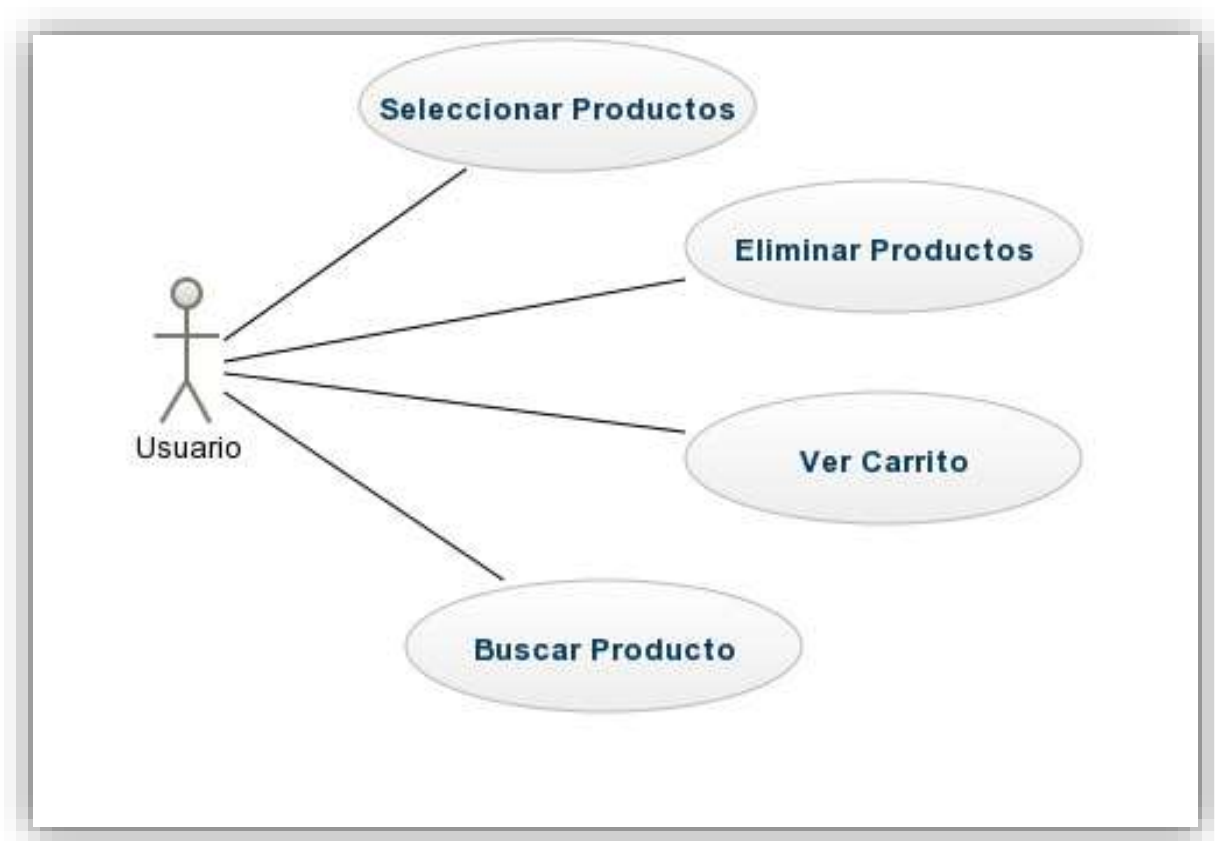


Imagen 5: Caso de Uso - Gestión de venta

2. Confirmación del pedido.

Una vez realizado el pedido y tras verificar que es correcto, el usuario podrá confirmar el pedido.

Tras realizar el paso anterior se mostrará un formulario para poder seleccionar los clientes que podrán compartir la compra. Podemos observarlo en el diagrama representado en la imagen 6.

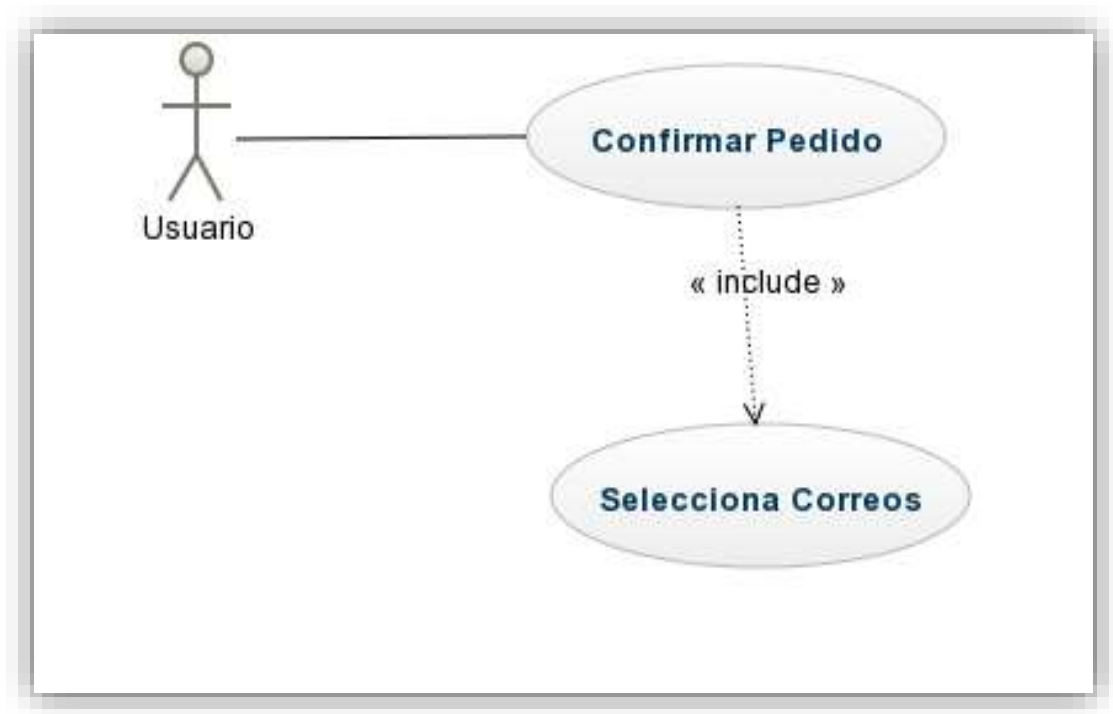


Imagen 6: Caso de Uso - Confirmación del pedido

3. Confirmación de pago.

Una vez confirmado el pedido, el sistema crea los correos electrónicos para posteriormente enviarlos a cada cliente, donde se indica un enlace para confirmar el pago.

Con este enlace el usuario podrá confirmar su pago. Para ello el sistema crea un acceso independiente para cada pago de cada cliente. No obstante todos los accesos estarán relacionados con el mismo pedido. El resultado lo vemos en la imagen 7.

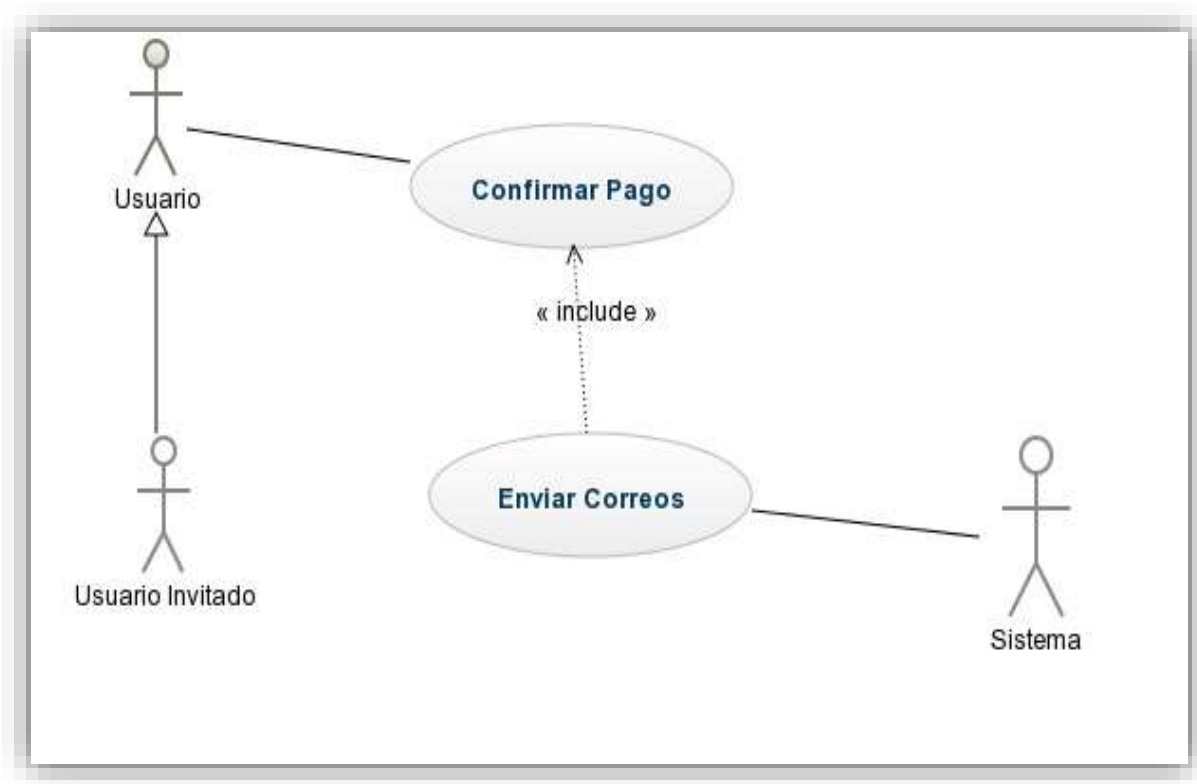


Imagen 7: Caso de Uso - Confirmación de pago

4. Pago del pedido.

Una vez confirmado el pago, el sistema ofrecerá al usuario el acceso a un terminal de pago para pagar su parte del pedido. El esquema de este caso de uso sería la imagen 8.

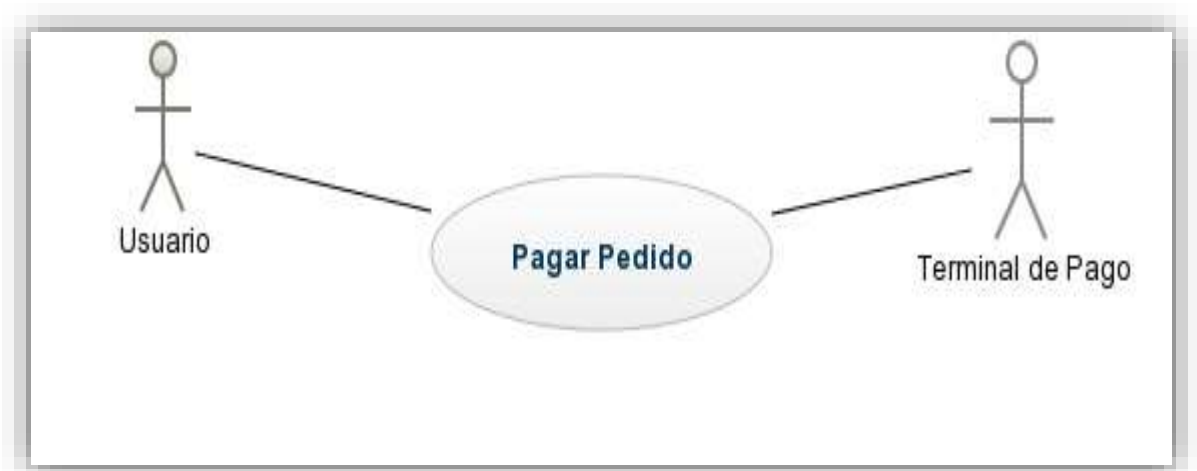


Imagen 8: Caso de Uso - Pago del pedido

5. Confirmación del terminal de pago.

Una vez el usuario ha pagado mediante el terminal de pago virtual, éste mismo enviará al sistema la confirmación del pago correcto o en su caso si ha habido algún error en este pago.

Después de que cada usuario haya hecho su pago respectivo, el sistema comprobará si todos los usuarios de cada pedido han realizado el desembolso, para poder enviar el pedido. Lo podemos ver en el diagrama de la imagen 9.

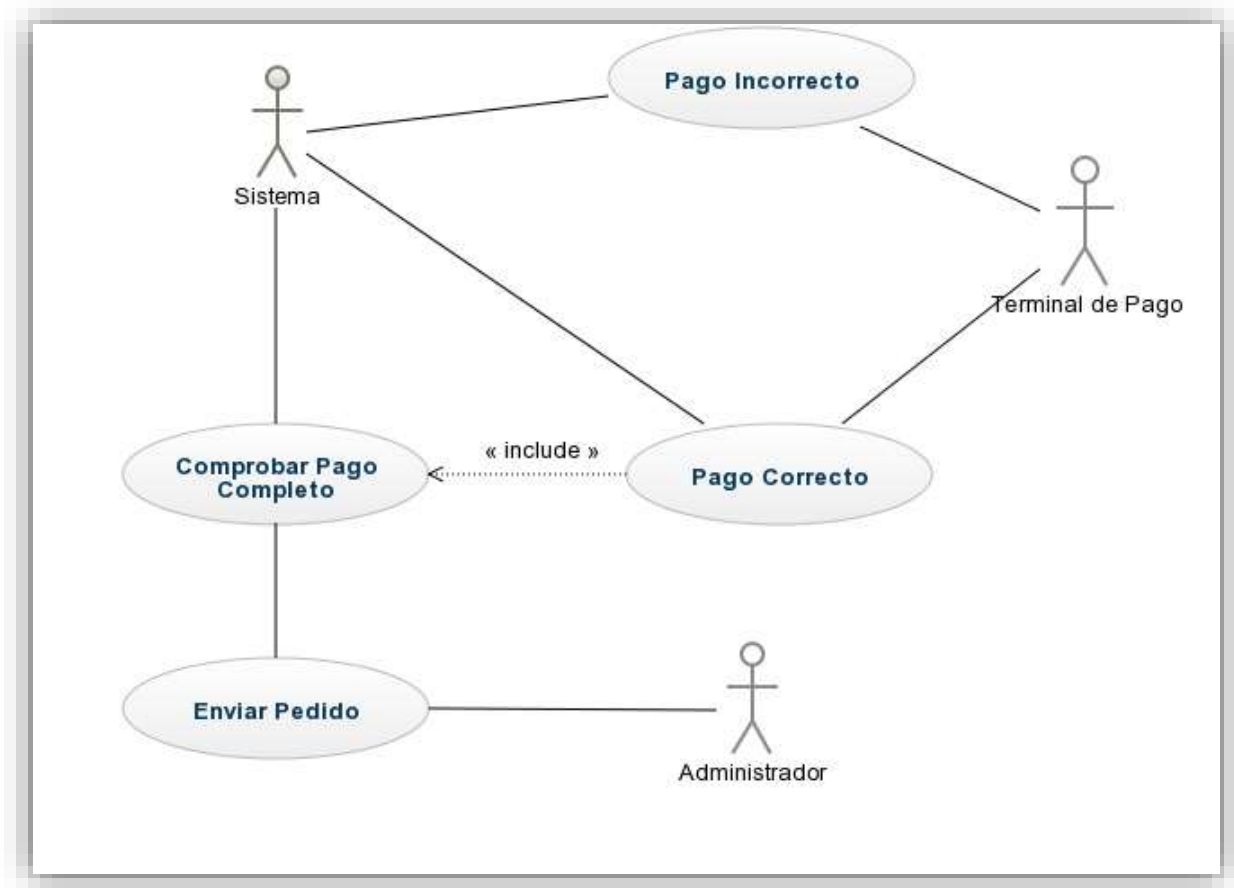


Imagen 9: Caso de Uso - Confirmación del TPV

6. Administración de la base de datos.

El administrador gestionará la base de datos con los diferentes casos de uso para este proceso. En este caso son Añadir producto, Eliminar producto y Actualizar pedidos. Podemos observarlo en el diagrama representado en la imagen 10.

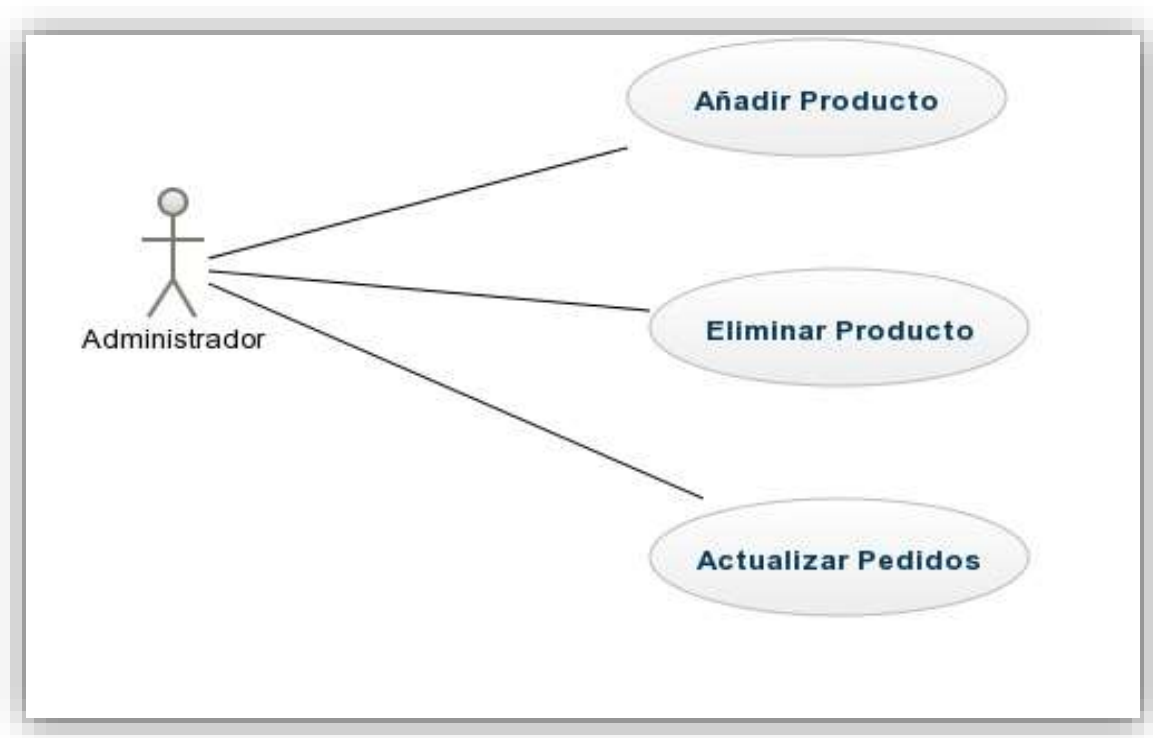


Imagen 10: Caso de uso - Administración

7. Gestión de eventos.

El sistema gestionará eventos en la plataforma enviando información al administrador. El esquema de este caso de uso sería la imagen 11.

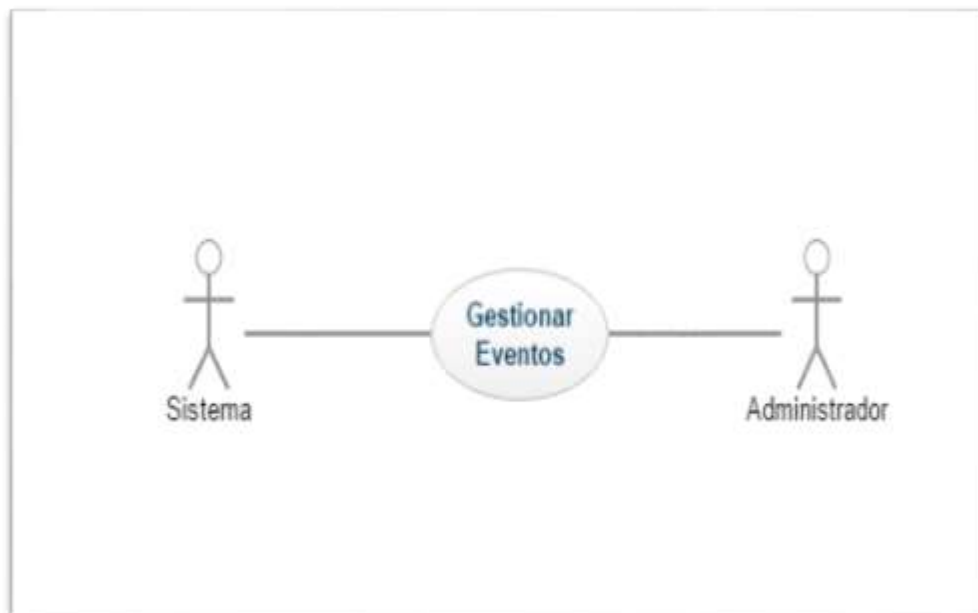


Imagen 11: Caso de Uso - Gestión de Eventos

3.4. Interfaces de usuario

En este apartado pasaremos a detallar los interfaces de usuario por los cuales el usuario se relacionará con el sistema. Para ello vamos a diseñar los prototipos correspondientes para cada evento en el cual iteran el usuario con el sistema.

Igual que en un edificio no se comienza a construir sin unos planos, un proyecto no se puede comenzar a desarrollar sin tener claro el qué y el cómo se quiere llegar a conseguir. Un error muy común es comenzar a escribir código en cuanto tienes una idea, para que al momento siguiente te des cuenta de que eso que has programado no sirve para nada porque no has caído en algún problema que se podría presentar [13]. Por ello la importancia de analizar el sistema y crear uno prototipos adecuados al sistema que pretendemos crear.

La especificación de los principios generales de la interfaz de usuario sería.

- Tanto el acceso a la aplicación como su uso se realizará mediante cualquier navegador web.
- Los mensajes de error serán mostrados por pantalla en un lenguaje sencillo y fácilmente de interpretar.
- En caso de que el cliente no haya introducido un campo correctamente la aplicación avisará debidamente, utilizando para ello los instrumentos que posee las herramientas que utilizaremos.

Partiremos de la pantalla inicial, que tendrá el aspecto que vemos en la imagen 12.

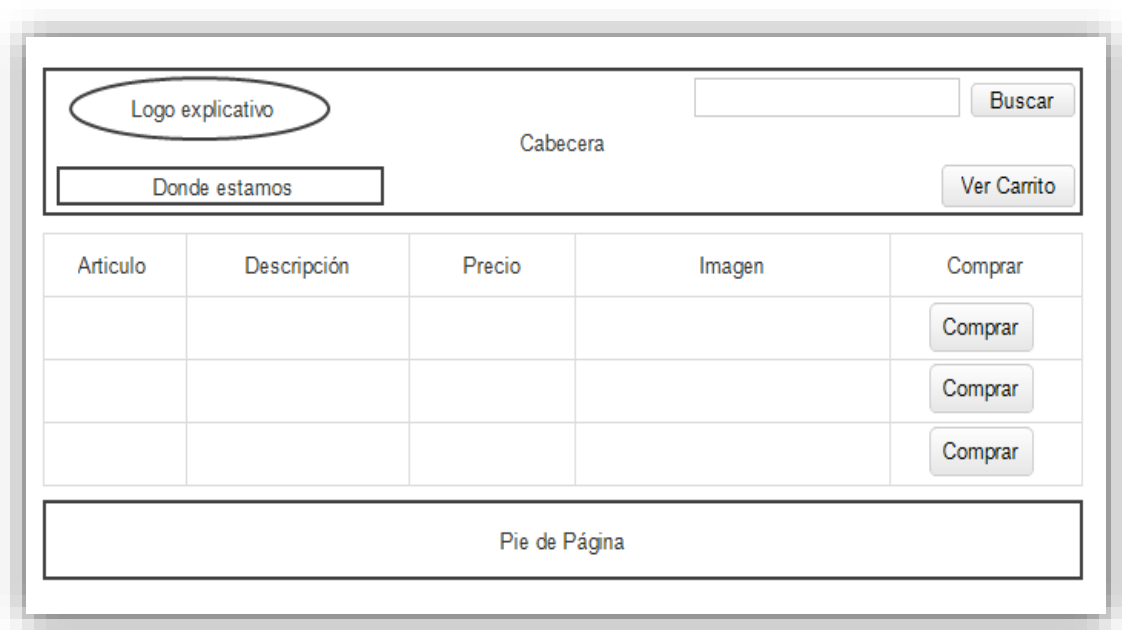


Imagen 12: Prototipo - Página principal

En esta pantalla tenemos, arriba una cabecera fija para mostrar información sobre la empresa y el logo. A la izquierda tenemos un buscador de los productos para que nos muestre los productos disponibles según los parámetros introducidos. Debajo tenemos un enlace al carrito de compras donde podemos ver el estado actual de nuestra compra, es decir, los productos que han escogido los clientes.

En el centro encontramos una sección con una tabla donde estará el catálogo de los productos que vende la tienda. Para cada uno de ellos, tendrá una descripción, una imagen y el precio respectivo. Finalmente cada uno tendrá un botón para comprar el producto y añadirlo al carrito.

Finalmente debajo encontramos un Pie de página, donde podemos colocar diferente información como por ejemplo el contacto de la empresa. Aunque en nuestro caso, éste no es uno de los objetivos que pretendemos con este proyecto.

El siguiente prototipo sería el interfaz para gestionar el carrito de la compra, después de que el usuario haya elegido los artículos. El resultado del prototipo sería la imagen 13.

Logo explicativo

Cabecera

Donde estamos

Producto	Precio	Cantidad		Subtotal
		<input type="text"/>	Actualizar	Eliminar
		<input type="text"/>	Actualizar	Eliminar
		<input type="text"/>	Actualizar	Eliminar

Total

Confirmar Pedido

Pie de Página

Imagen 13: Prototipo - Estado del pedido

En este caso, la apariencia es similar a la primera pantalla de la aplicación. De este modo tenemos una tabla similar al otro prototipo. Sin embargo, ésta tiene una intención de gestionar los productos que el usuario ha seleccionado.

La tabla está compuesta por cada producto escogido. Y por cada uno de ellos tenemos su nombre, su precio, la cantidad y el coste que tendría cada producto, que este caso es llamado subtotal.

Por otro lado, la tabla incluye por cada producto dos botones. Uno de ellos se utilizará para cambiar el número de artículos seleccionados, en este caso “Actualizar”. El otro botón servirá para quitar el producto, si el cliente se ha equivocado en la compra, este sería “Eliminar”. Finalmente, abajo en la derecha localizamos un botón para confirmar el pedido y acceder al nuevo formulario.

Consecutivamente después de confirmar el pedido, nos encontraríamos con el siguiente interfaz ilustrado en la imagen 14.

Producto	Precio	Cantidad	Subtotal

Total

Imagen 14: Prototipo - Confirmación del pedido

Este prototipo muestra una gran similitud con el anterior. Sin embargo han desaparecido las funcionalidades anteriores de los botones porque en este punto ya no son necesarias. Y solamente encontramos el botón “Comprar” para finalizar la gestión de la compra.

En este punto, en cualquier tienda virtual accederíamos a la plataforma virtual de pago, pero en nuestro caso ponemos en marcha el sistema que queremos implementar, es decir el pago colaborativo.

En los siguientes prototipos de interfaces, mostramos como sería la gestión del pago entre los diferentes usuarios. El siguiente mostraría como gestionar la solicitud de correos de los restantes clientes que pagarán la compra, lo podemos observar en la imagen 15.

Logo explicativo

Cabecera

Donde estamos

Pedido

Producto	Precio	Cantidad	Subtotal

Total

Email:

Pie de Página

Imagen 15: Prototipo - Confirmación de Email

La funcionalidad sería realmente sencilla, el usuario pulsaría el botón “Añadir más correos” para que se pueda incluir en el formulario otro input para añadir otro correo electrónico. Cada vez que se pulsa el botón haría la misma operación y cada vez que se pulsa “Eliminar correos” haría lo contrario; es decir, quitar un input de los incluidos anteriormente.

Finalmente un botón “Enviar” para confirmar los correos y para que el sistema los envíe a cada usuario. Esta funcionalidad la podemos ver en la imagen 16.

Prototipo de un formulario para añadir emails. El formulario contiene tres campos de texto etiquetados como "Email:". Debajo de los campos hay dos botones: "Añadir más correos" y "Eliminar correos".

Imagen 16: Detalle Prototipo - Añadir email

El prototipo mostrado en la imagen 17 representa el enlace que recibe cada usuario en su correo electrónico. En él se muestra los datos del pedido junto con el importe que debería pagar el propio usuario.

Mediante el botón "Pagar" se accede a la plataforma de pago virtual para que realice el pago correspondiente mediante su tarjeta de crédito.

Prototipo de una pantalla para pagar un pedido. La pantalla está dividida en secciones: "Logo explicativo" (destacado con un óvalo), "Cabecera" (que contiene un botón "Donde estamos"), "Nº Pedido" (con un campo de texto), "Importe" (con un campo de texto), "Descripción" (con un campo de texto grande), y "Pie de Página" (que contiene los botones "Pagar" y "Volver").

Imagen 17: Prototipo - Pagar pedido

Los dos últimos prototipos de interfaces que podemos ver a continuación muestra la gestión de la aplicación al recibir la respuesta de la plataforma de pago tras realizar el cliente su pago respectivo.

El primero, sería la gestión si el pago de la compra ha sido correcto. En este caso el sistema verificará en qué estado está el pago de los demás clientes que pagan el pedido y mostrará al usuario dos posibles situaciones. Si es el último indicará que ya han pagado todos y que el pedido será enviado. Y en otro caso, indicará que faltan usuarios por realizar su pago correspondiente. Representado en la imagen 18.

El otro prototipo, muestra como gestionaría la plataforma tras un error en el pago, en este caso, mostrando un mensaje. En este caso sería la imagen 19.

Logo explicativo

Cabecera

Donde estamos

Nº Pedido

Importe

Estado del pedido

Volver

Pie de Página

Detailed description: This is a wireframe of a user interface for a payment confirmation screen. It features a header area with a 'Logo explicativo' (circled) and the word 'Cabecera'. Below the header is a box labeled 'Donde estamos'. The main content area contains three input fields: 'Nº Pedido', 'Importe', and 'Estado del pedido'. A 'Volver' button is positioned below these fields. At the bottom, there is a footer box labeled 'Pie de Página'.

Imagen 18: Prototipo - Pago Correcto

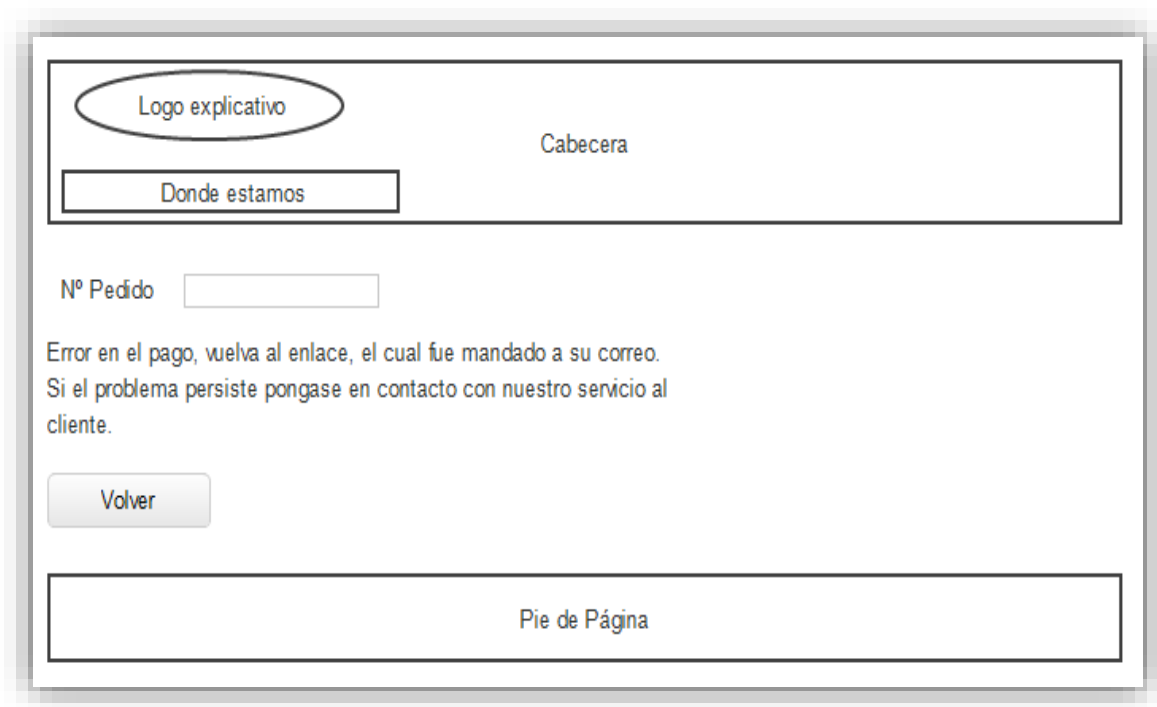


Imagen 19: Prototipo - Pago Incorrecto

3.5. Especificación del plan de pruebas

Para comprobar las funcionalidades de los casos de uso se llevarán a cabo las siguientes pruebas.

- Se escogerá varios productos del catálogo de la tienda virtual para realizar varios pedidos de prueba.
- Se dispondrá de varios correos para poder enviar su parte correspondiente del pago del pedido de prueba y comprobar la recepción y confirmación de éste.
- Se harán distintas pruebas de pago mediante una plataforma de pago virtual por medio del acceso que nos suministra de prueba, utilizando para ello los números de tarjetas de crédito que tienen para realizar las distintas pruebas.
- En todo momento se comprobará el funcionamiento de la aplicación accediendo a la base de datos correspondiente y verificando que los cambios se han realizado oportunamente.
- Se estudiará de forma oportuna la usabilidad de la plataforma, siguiendo los diez principios de usabilidad formulados por Jakob Nielsen [14]. Para ello se irá verificando cada uno de ellos.

Capítulo 4: Diseño del Sistema

Partiendo del análisis que acabamos de realizar deberemos obtener las especificaciones necesarias para realizar el proyecto con éxito. En esta fase deberemos obtener la definición de un modelo arquitectónico del sistema, la identificación de cada subsistema con sus requisitos para facilitar la comprensión de éste, los casos de uso anteriormente definidos y que ahora deberemos revisar; y finalmente los requisitos de implantación.

4.1. Arquitectura

Vamos a definir la arquitectura del sistema que nos permitirá identificar los componentes del mismo. Si hablamos a grandes rasgos de la arquitectura del sistema, podemos definir dos partes bien diferenciadas.

Por un lado, está la propia tienda virtual, donde los clientes interactúan con ella realizando las diferentes compras. Y por otro lado está la parte que corresponde a la parte de administración de la tienda online, donde el administrador se encarga de las tareas de actualizar los productos y comprobar el estado de los pedidos.

Estas dos partes interactúan con la parte fundamental del sistema y en este caso es la base de datos. En esta parte es donde se almacenan todos los datos, además de recuperarlos, modificarlos o eliminarlos.

Tampoco nos podemos olvidar de una parte relativamente importante, que son los procesos que se ejecutan de manera organizada según algunas condiciones definidas y que pueden consultar y modificar la base de datos.

Así pues, podemos identificar los sistemas que mostramos en el diagrama de componentes en la imagen 20.

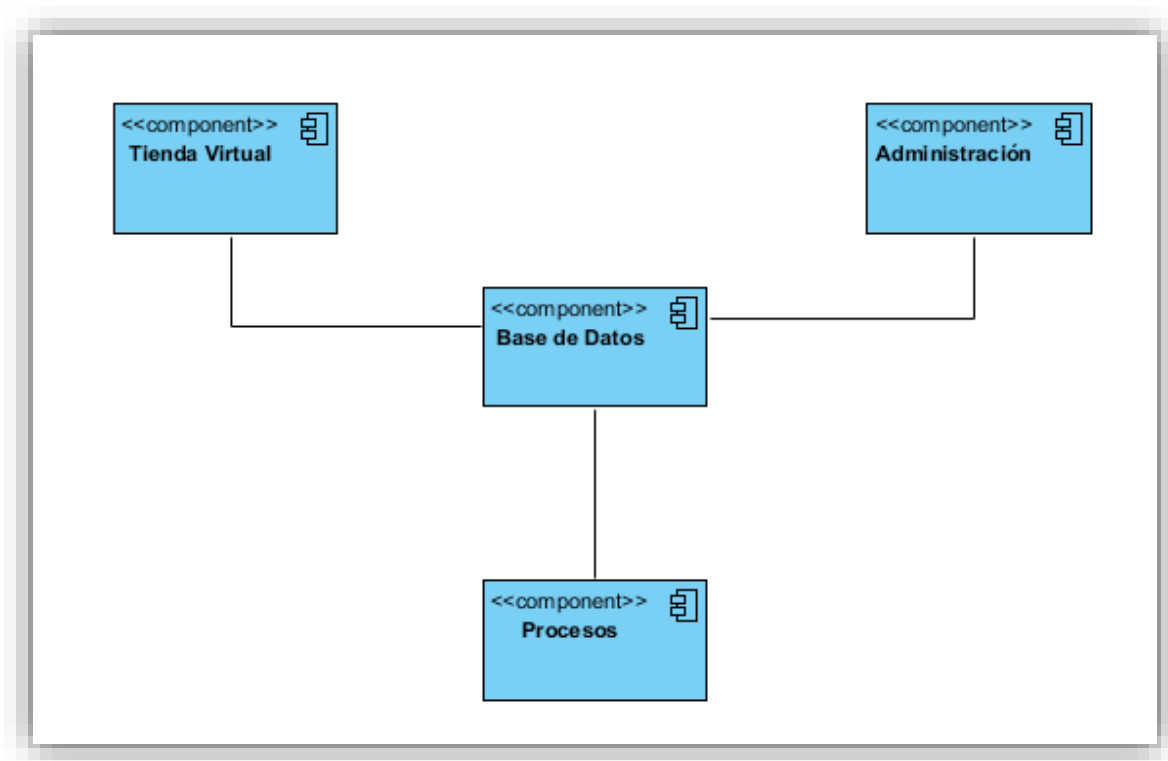


Imagen 20: Arquitectura

Pasaremos a detallar las funciones y con quien se relaciona cada componente. En primer lugar, la tienda virtual ofrece todo lo necesario para que el cliente pueda realizar una compra, esto implica que se relaciona directamente con la base de datos para consultarla y modificarla según lo que haya solicitado el usuario.

Además, tenemos la administración del sistema, en ella los administradores de la tienda gestiona lo que se mostrará en la plataforma. Su relación está directamente relacionada con la base de datos, ya que se dedicará a la inserción y actualización de los productos de la tienda.

También existe la base de datos que almacena todos los datos que maneja tanto la tienda virtual como la parte de la administración de la tienda. Este componente está relacionado con los demás porque todos acceden a ésta para consultar y actualizar.

Por último, tenemos el sistema de procesos que se ejecutan de manera automática según se cumplan determinadas condiciones como por ejemplo el envío de correos a los clientes o la comprobación del estado del pedido. Su implicación está relacionada con la base de datos, donde actualiza los contenidos y verifica el estado de ellos.

4.2. Identificación de subsistemas

Algunos de los sistemas del diagrama anterior están compuesto a su vez por subsistemas. El siguiente paso será identificar los subsistemas que tendremos que desarrollar para el proyecto que estamos realizando.

Después de un estudio de los sistemas podemos dividir el sistema Tienda Virtual en tres subsistemas, los cuales serían Gestión Compra, Gestión Pedido y Gestión Pago. Como podemos ver en la imagen 21.

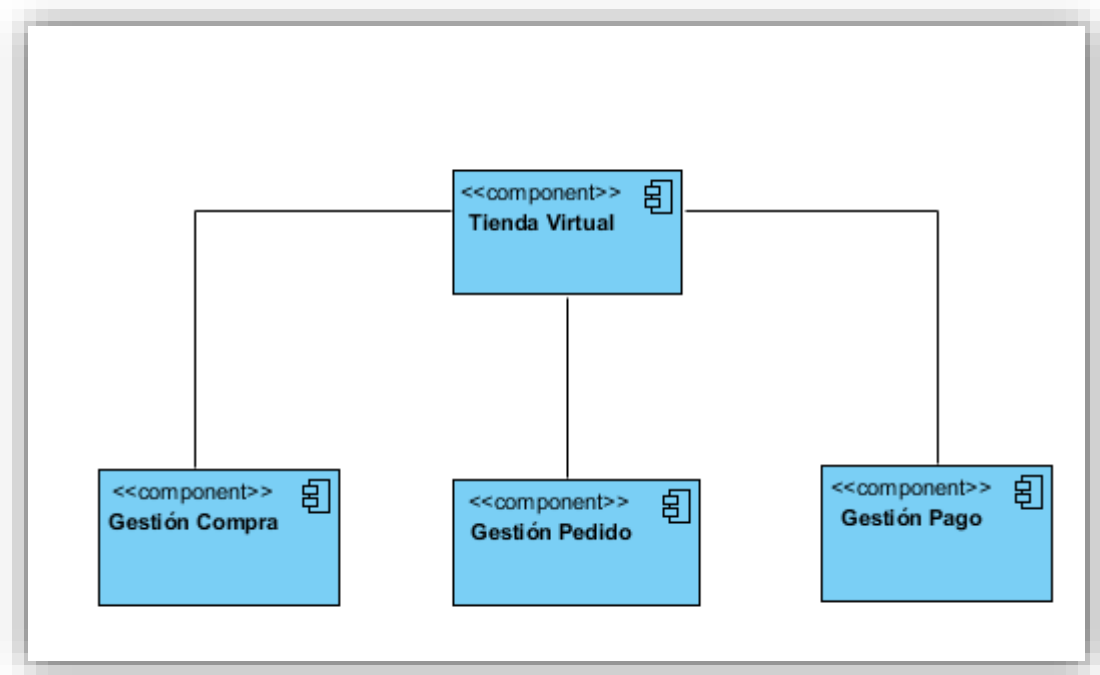


Imagen 21: Subsistemas de Tienda Virtual

4.3. Revisión de casos de uso

A continuación se relacionan los casos de uso que se desarrollaron en la fase de análisis con los sistemas y subsistemas identificados.

- Caso de uso: Gestión de Ventas.

Se corresponde al subsistema de Gestión Compra, donde el usuario realiza todo lo necesario para elegir los productos de la tienda virtual. A su vez está relacionado con los sistemas Base de Datos y Procesos; el primero por la relación con los productos contenidos en la base de datos y el segundo porque se realizan procesos automáticos durante la gestión de la compra.

- Caso de uso: Confirmación de Pedido.

Este caso se corresponde con el subsistema de Gestión Pedido, donde se gestiona el pedido después de que el usuario haya seleccionado los productos. Está relacionado directamente con el sistema Base de Datos porque en ésta se almacenan y gestionan los datos del pedido.

- Caso de uso: Confirmación de Pago.

Se corresponde con el subsistema de Gestión Pedido, donde se tramita la gestión de los envíos de los correos y la confirmación del pago. Por ello también depende del subsistema Gestión Pago. A su vez está relacionado con los sistemas Base de Datos porque se gestiona todo en ella. Y con el sistema de Procesos porque en este caso se realizan procesos automáticos.

- Caso de uso: Pagar pedido.

Este caso se corresponde con el subsistema de Gestión Pago, donde gestiona el pago de cada usuario. Está también ligado al sistema Base de Datos porque desde ahí se controla los pagos de cada usuario.

- Caso de uso: Confirmación TPV.

Se corresponde con el subsistema de Gestión Pago, donde se controla y gestiona la información recibida desde el terminal de pago. Por supuesto, está relacionado con el sistema Base de Datos porque se actualiza datos en ella mediante el sistema de Procesos que comprueba automáticamente el estado global del pago de los diferentes usuarios.

- Caso de uso: Administración.

En este caso se corresponde con el sistema de Administración, donde se gestionan el estado de los productos de la tienda virtual.

- Caso de uso: Gestión de eventos.

Se corresponde con el sistema Procesos, encargado de administrarlos cada cierto tiempo.

4.4. Estructura de la base de datos

La base de datos tendrá una estructura simple con tres entidades y dos relaciones entre ellas que nos servirá para poder trabajar con el sistema y almacenar la información necesaria. Esta organización se puede ver reflejada en el diagrama de entidad-relación mostrado en la imagen 22.

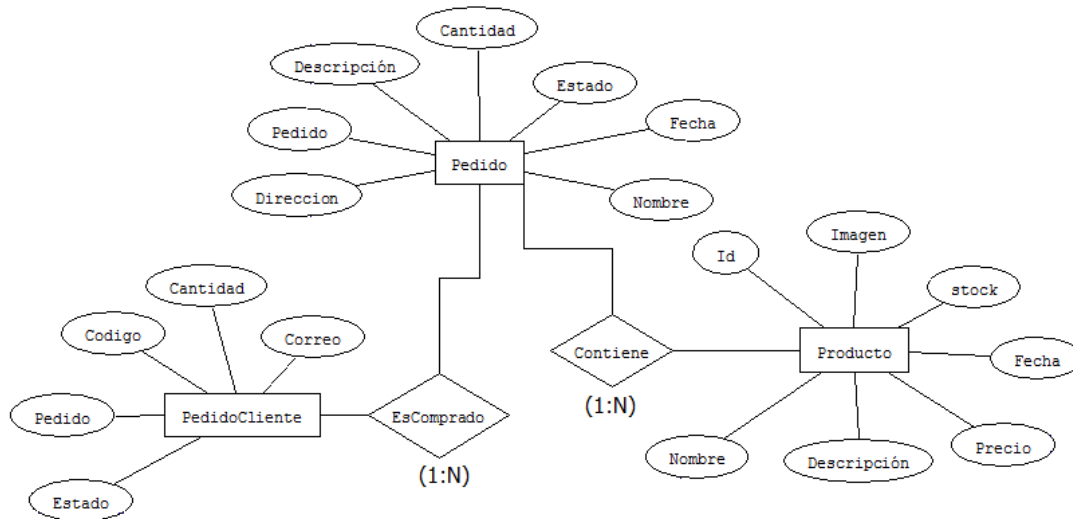


Imagen 22: Diagrama E/R

4.5. Análisis de componentes del sitio web

Después de un estudio pormenorizado de las diferentes alternativas que podemos utilizar como herramientas en nuestra plataforma web, hemos decidido que los componentes a utilizar son los mostrados en la tabla 1.

Componente	Paquete	Versión Prevista	Licencia
Sistema Operativo	Ubuntu	14.04	GPL
Base de Datos	MySQL	5.7.9	GPL
Servidor Web	Apache	2.2.14	Apache Software License
Herramientas de depuración	Firebug	2.0.13	BSD
Interprete de Scripts	PHP	7.0.2	PHP License
Gestor de Base de Datos	PhpMyAdmin	4.5.3.1	GPL
Entorno de desarrollo	Sublime Text	2.0.2	Software propietario – Free
Herramientas UML	GenMyModel	Online	Evaluación

Tabla1: Componentes del sitio

Capítulo 5: Desarrollo

Ahora ya podemos pasar a la construcción del sistema basándonos en el trabajo anteriormente realizado ya que tenemos la información necesaria sobre lo que se ha de desarrollar, las herramientas que utilizaremos para ello, los entornos que manejaremos y demás elementos.

5.1. Planificación de las actividades de desarrollo e integración de sistema

Con respecto a la secuenciación del proyecto en su gran mayoría podemos considerar que es secuencial, en gran parte debido a que los roles de analista y programador están ocupados por la misma persona. En la imagen 23 vemos el diagrama de Gantt con la secuenciación de las actividades.

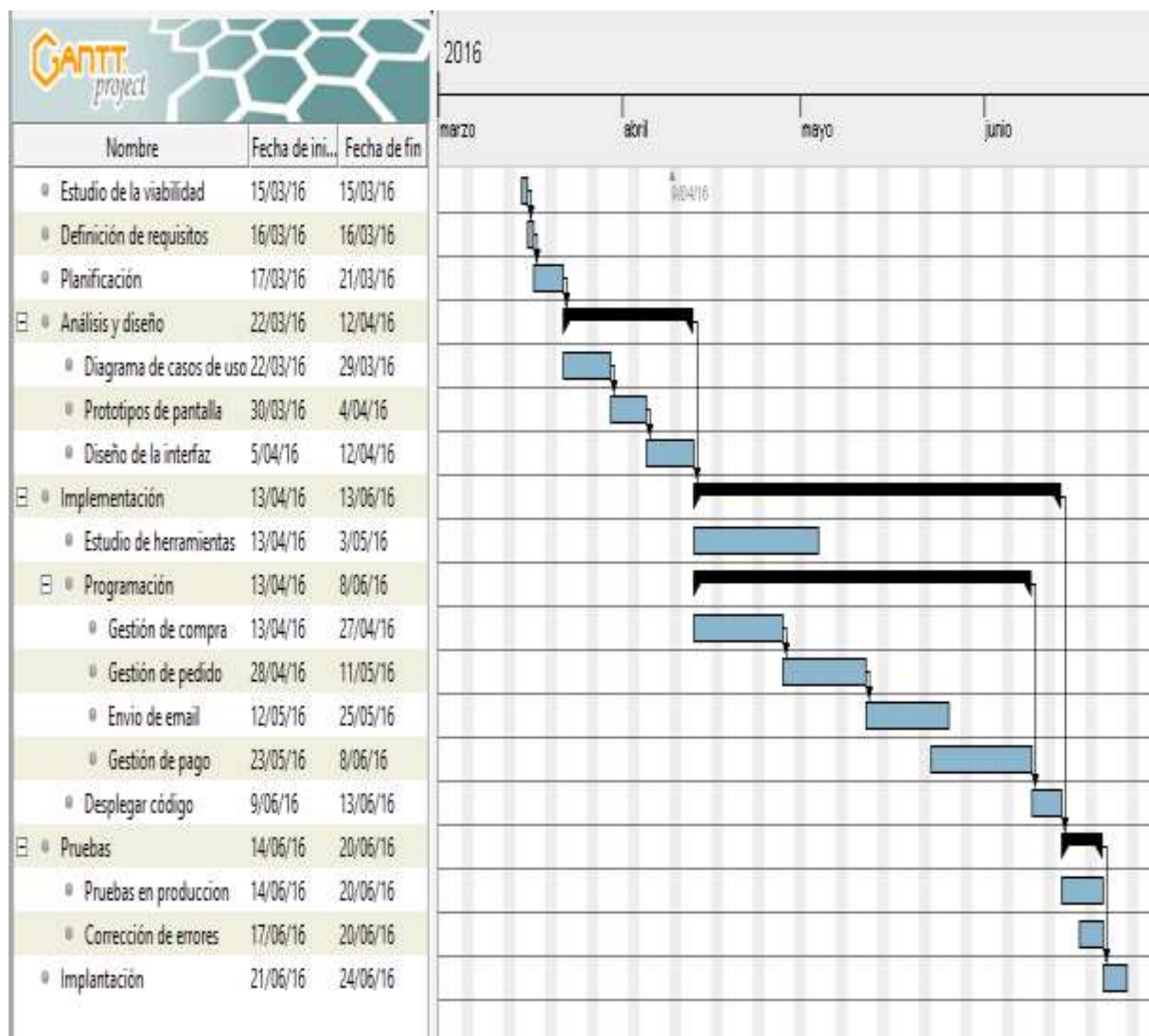


Imagen 23: Diagrama de Gantt

Sin embargo, algunas de las actividades pueden ser solapadas. Alguna porque se puede avanzar cuando la tarea está en punto intermedio. En este caso, cuando el estudio de herramientas está avanzado se puede inicializar la etapa de implementación con las herramientas que son ya conocidas.

Por otro lado, tenemos otras actividades que pueden ser desarrolladas simultáneamente. En particular, es el caso de las pruebas en producción y corrección de errores porque estas tareas se desarrollan prácticamente en paralelo. Además con este solapamiento podemos ganar tiempo en la implantación.

Así mismo, realizando un cálculo de la dedicación de las tareas y el tiempo que aproximadamente se tiene para la realizando del proyecto, tenemos unos 120 días con una dedicación de 3 horas diarias. Nos supone un total de aproximadamente 360 horas para la ejecución del trabajo.

5.2. Desarrollo del proyecto

En primer lugar se han instalado todas las herramientas necesarias mencionadas en los puntos anteriores para poder realizar el proyecto. Todo ello sobre una distribución de Ubuntu 14.04.

Para el desarrollo del proyecto se ha utilizado para editar el código Sublime Text 2 [15], el cual nos ofrece un interfaz fácil de usar. Además posee un sistema de ayuda con diferentes colores para identificar los elementos que estamos proponiendo. En conjunto es una aplicación bastante útil para lo que queremos proponer.

Con todos los elementos instalados se han codificado los diferentes sistemas y subsistemas expuesto en la parte de diseño, utilizando para ello la arquitectura definida en este mismo punto.

Tras el proceso de desarrollo y basándonos en los prototipos diseñados anteriormente, los cuales han resultado de gran ayuda; el resultado de la página principal sería el que podemos ver en la imagen 24.

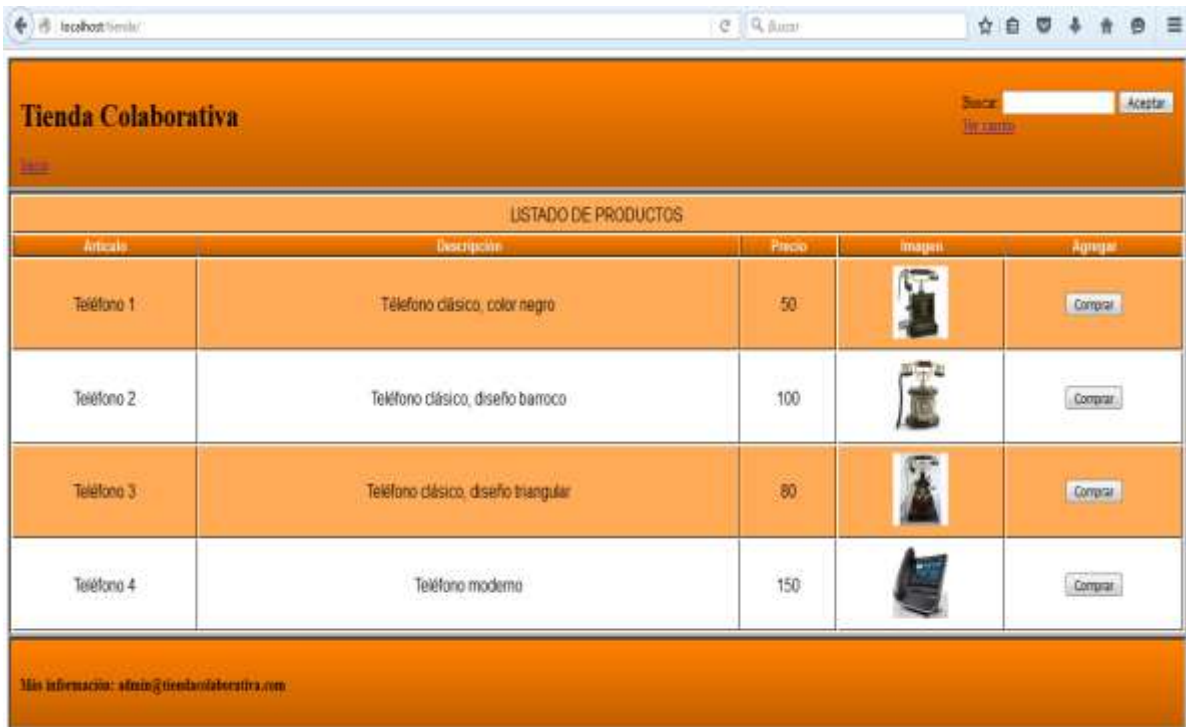


Imagen 24: Página principal de la aplicación

En las siguientes imágenes podremos ver los diferentes interfaces de la aplicación en funcionamiento. En la imagen 25 vemos el carrito de compra después de haber realizado una compra. En esta parte, además podemos modificar el número de productos, así como eliminarlos.



Imagen 25: Interfaz - Carrito de compras

El interfaz representado en la imagen 26, se solicita los correos de los usuarios que participan en el pago. A su vez se pide la dirección y nombre para poder enviar el pedido una vez confirmado el pago por la plataforma.

Tienda Colaborativa

Inicio > Carrito > ConfirmarPedido > DatosCompra

Datos de la Compra

LISTADO DE SU COMPRA			
Producto	Precio	Cantidad	Subtotal
Producto 3	80	2	160

TOTAL: 160

Introduzca los correos de los pagadores y dirección de envío:

Nombre:

Dirección:

Email:

Email:

Más información: admin@tiendacolaborativa.com

Imagen 26: Interfaz - Envío de correos

Del mismo modo, una vez enviado los correos, la plataforma nos emplaza a un nuevo interfaz que podemos ver en la imagen 27.

Tienda Colaborativa

Inicio

Su compra es la siguiente:

Producto: Producto 3
Precio: 80
Cantidad: 2

Mail enviado correctamente a

Mail enviado correctamente a

Más información: admin@tiendacolaborativa.com

Imagen 27: Interfaz - Confirmación de email

En el módulo se gestiona los envíos de correos a cada usuario, un ejemplo de envío de estos correos lo podemos ver en la imagen 28. En ellos nos emplaza con un enlace individual mediante un código aleatorio para cada uno, con el cual se accede de nuevo a la plataforma.



Imagen 28: Información recibida por email

En el enlace correspondiente lo podemos ver en la imagen 29, donde nos ofrece la opción de pagar el pedido o volver a la página de inicio de la web.

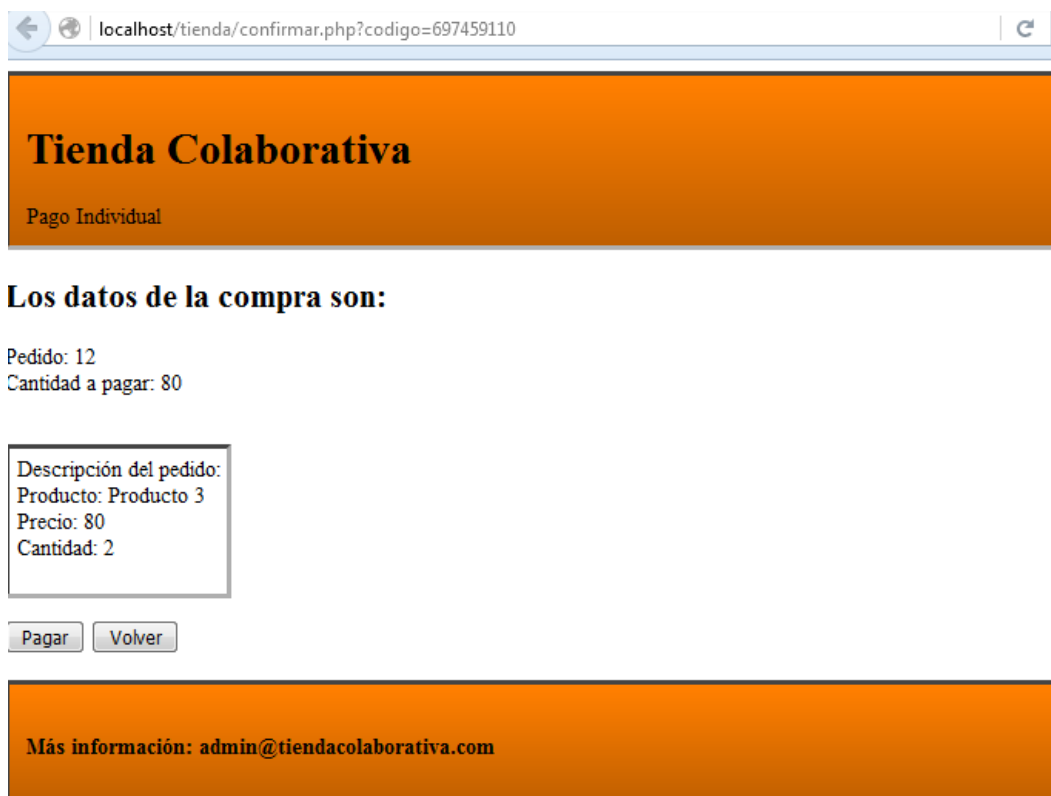


Imagen 29: Interfaz - Confirmación de pago

Una vez pulsado el botón de pagar, accedemos a la plataforma de pago, donde se le indica el coste del pago y el número del pedido al cual se refiere. En la imagen 30 tenemos los dos pasos realizados durante el proceso de pago. En este caso hemos utilizado la plataforma de Liberbank por su facilidad de uso.



Imagen 30: Proceso de pago en el Terminal

Finalmente, solamente queda recepcionar la notificación que venga desde el terminal de pago. En este caso, nos encontramos dos posibilidades. En el primer caso, es que el pago se ha realizado correctamente. En este caso el usuario se encontrará con dos posibilidades, o que todos los usuarios hayan pagado su parte, que en este caso saldría el interfaz mostrado en la imagen 31.



Imagen 31: Interfaz - Pago Correcto 1

O en el caso de que todavía quede algún usuario por participar en su propio pago, el interfaz que se mostrará será la imagen 32, donde se le explica la situación actual.

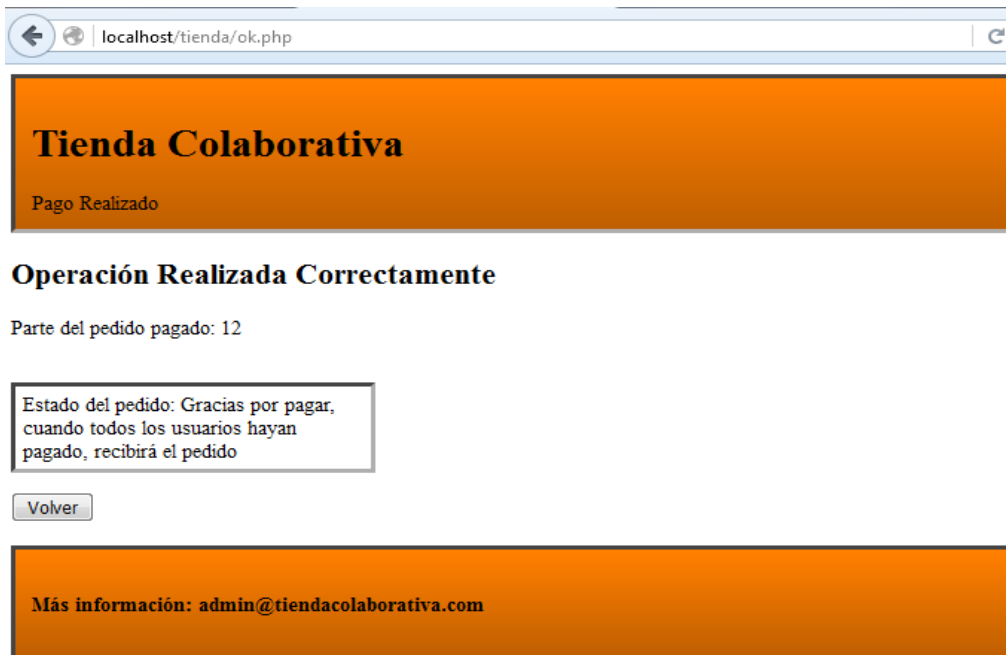


Imagen 32: Interfaz - Pago Correcto 2

Por el contrario si ha habido algún error en el pago en el terminal, entonces se le mostrará el interfaz visualizado en la imagen 33, que se puede ver a continuación.



Imagen 33: Interfaz - Pago Incorrecto

5.3. Documentación

La documentación para este proyecto, principalmente será esta misma memoria, ya que durante su desarrollo se ha detallado todos los elementos de éste. Y servirá como consulta ante cualquier necesidad.

Por otro lado, se podía diseñar un manual para que puede ser consultado por el usuario. Sin embargo, podemos ver que la aplicación es bastante intuitiva de utilizar, por lo que se ha decidido no realizarlo.

Partiendo de lo expuesto en el párrafo anterior, podríamos realizar un manual para que pueda ser usado por un posible administrador de la plataforma. No obstante, se ha cambiado por una explicación pertinente del uso de la plataforma y de la gestión de la propia base de datos.

Capítulo 6: Implantación

Desde el punto de vista tecnológico el sistema web parte de un entorno de desarrollo donde se instalan los elementos lo más parecido a los que finalmente se usarán. En este entorno se desarrolla el software para nuestra tienda virtual.

Una vez realizado todas las funcionalidades necesarias para nuestra web. Se actuará mediante un test de pruebas, donde podamos depurar cualquier error producido durante la etapa de desarrollo.

Asimismo en un entorno de preproducción, donde el ambiente será el real donde la aplicación estará alojada, se realizarán los cambios oportunos para que el software funcione correctamente en el nuevo ambiente, como puede ser la base de datos, que en este caso se puede realizar un sencillo script para instalarla en el nuevo contexto.

Finalmente en un entorno de producción se comprueba que la aplicación está totalmente libre de fallos y que los clientes pueden acceder perfectamente sin tener ningún problema. Para ello se han realizan diferentes test o pruebas que detallaremos en el punto siguiente.

6.1. Automatización de tareas

Para automatizar las tareas de mantenimiento y gestión de la aplicación utilizaremos las herramientas que nos suministra el sistema operativo, en este caso Ubuntu. Para ello utilizaremos el programador de tareas que posee Ubuntu, el cron, un programador sencillo pero muy eficiente.

Para esto, se creará un fichero PHP donde se almacenará todo el código necesario para la automatización de las tareas y este lo ejecutaremos todos los días una vez. El fichero del cron tendrá el aspecto mostrado en la imagen 34.

```
GNU nano 2.2.6 Archivo: /tmp/crontab.6kxuRb/crontab
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
@ 0 * * * php /var/www/html/tienda/tareas.php
```

Imagen 34: Cron del sistema

6.2. Pruebas realizadas

- **Chequeo del código:** para este caso hemos utilizado una herramienta online que nos comprueba si hay algún error en el código PHP. La herramienta utilizada ha sido la proporcionada por la página “*phpcodechecker*” [16]. El resultado ha sido positivo y no ha encontrado ningún error. Lo podemos ver en la imagen 35.

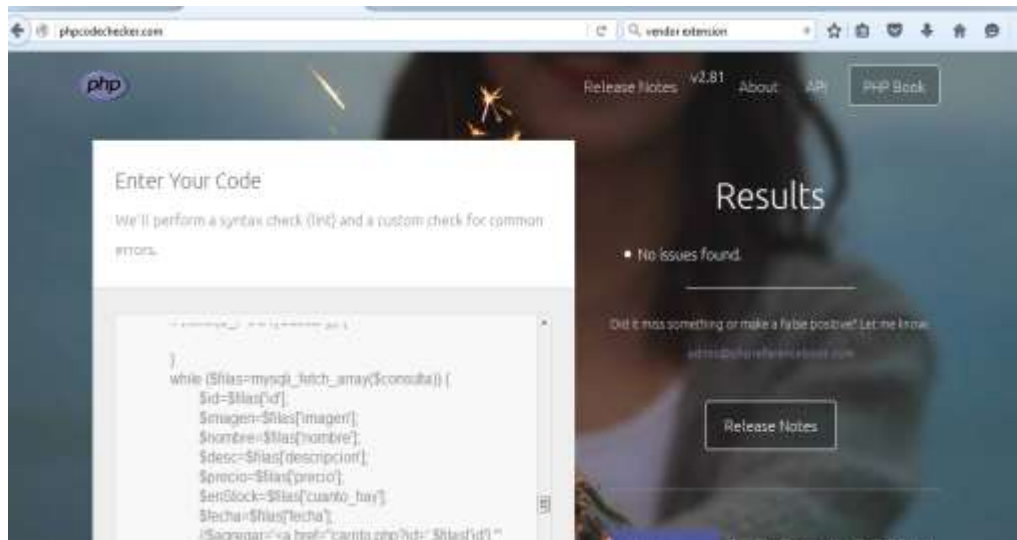


Imagen 35: Resultado del test del código

- **Chequeo CSS:** en este caso para verificar que las hojas de estilo en cascada (*cascading style sheets*) aplicadas está correctamente aplicadas se ha usado el validador que proporciona W3C [17]. El resultado lo podemos observar en la imagen 36.



Imagen 36: Resultado del test CSS

Como podemos ver el test no ha encontrado ningún error en nuestro código en lo relacionado con las hojas de estilo. Sin embargo, nos ofrece ocho advertencias que se refieren a los prefijo –moz y –webkit utilizados en algunas propiedades para que puedan ser reconocidos por navegadores antiguos de las diferentes versiones de Firefox, Chrome, Safari y Opera respectivamente. Estas advertencias lo vemos en la imagen 37.

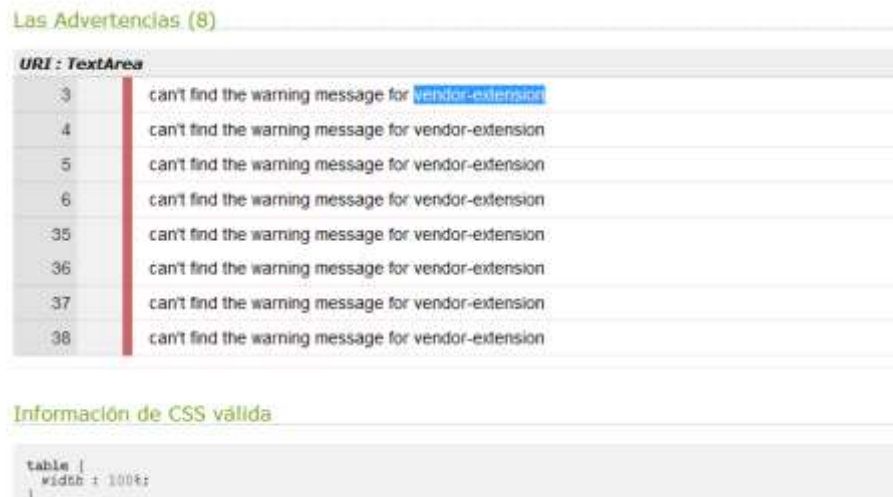


Imagen 37: Detalle de advertencia CSS

- **Chequeo responsive:** para ello hemos hecho varias redimensiones de la plataforma. Para un tamaño que ocupa toda la pantalla, ya hemos visto en las capturas anteriores que el diseño ha sido exitoso. Faltaría comprobar para otras dimensiones, si los objetos de la plataforma se adaptan a la dimensión correspondiente. Cabe destacar en este caso, que el resultado también ha sido aceptable como podemos ver en las imágenes 38 y 39.



Imagen 38: Tamaño pequeño de la aplicación



Imagen 39: Tamaño mediano de la aplicación

- **Test de navegadores:** para ello se ha probado directamente sobre cada uno de los navegadores. En este caso se ha instalado los diferentes navegadores en un sistema operativo posible de un cliente.

La prueba se ha realizado con éxito en todos los navegadores. Sólo en Internet Explorer, algunos elementos de los fondos no se han visualizado porque el propio navegador no los tiene implementados. Sin embargo la funcionalidad ha sido correcta en todos ellos. En las imágenes desde la 40 hasta la 43 podemos ver como la visualización de la plataforma es totalmente correcta en cada uno de los navegadores.



Imagen 40: Visualización en Internet Explorer



Imagen 41: Visualización en Mozilla Firefox



Imagen 42: Visualización en Google Chrome



Imagen 43: Visualización en Opera

- **Test de usabilidad:** para ello se analizarán siguiendo los diez principios de Jakob Nielsen [14].
 - ✓ **Principio 1. “Visibilidad del estatus del sistema”:** podemos decir que el usuario está permanentemente informado sobre lo que está pasando cuando interactúa en la plataforma.
 - ✓ **Principio 2. “Consistencia entre el sistema y el mundo real”:** se ha usado un interfaz intuitivo, por lo que el usuario se sentirá cómodo y no encontrará problemas. Son elementos que un usuario que está relativamente acostumbrado a comprar en Internet le resulta habituales.
 - ✓ **Principio 3. “El usuario es libre y tiene el control”:** en ningún momento el usuario tiene algún problema para navegar por la web, porque en la parte superior está la ruta que el usuario está siguiendo. De esta forma siempre podrá encontrar una salida para volver al principio.
 - ✓ **Principio 4. “Consistencia y estándares”:** la web tiene consistencia porque usa la misma estructura en toda la web, así el usuario solamente ve modificada la parte donde se incluye los contenidos.
 - ✓ **Principio 5. “Prevención de errores”:** durante toda la etapa de ciclo de vida de este software se ha desarrollado para prevenirlos y que no ocurran durante el uso del cliente.
 - ✓ **Principio 6. “Mejor reconocer que memorizar”:** en este caso el usuario tiene siempre toda la información a mano, gracias a la ruta completa que encontrará en la parte superior. Por lo que no tendrá que memorizar aparentemente nada en principio.
 - ✓ **Principio 7. “Flexibilidad y eficiencia de uso”:** en este principio, no puede haber mucha flexibilidad, ya que el sentido y funcionamiento de la plataforma es secuencial siguiendo unos pasos (compra,

confirmación, envío de email y pago). Sin embargo con el enlace de arriba se ofrece la posibilidad de volver hacia atrás.

- ✓ **Principio 8. “Diseño estético y minimalista”**: aquí podemos decir que cumple con este principio, porque los datos son precisos y objetivos. Y no son adornados con un gran número de información.
- ✓ **Principio 9. “Ayuda al usuario a reconocer, diagnosticar y recuperarse de los errores”**: igualmente que lo mencionado anteriormente, se ha intentado hacer un ciclo de vida correcto para que no surjan errores.
- ✓ **Principio 10. “Ayuda y documentación”**: en este caso, no tenemos documentación lo que puede ser una parte que no cumple del principio. Sin embargo pensamos que no es necesario, por la facilidad de uso de su contenido. Por otro lado con respecto a la ayuda. En la parte superior tenemos un buscador de productos que puede resultar mucho más fácil el uso de la aplicación.

En definitiva, la aplicación cumple en su gran mayoría con los principios aportados por Jakob Nielsen. Por lo que podemos decir que la aplicación es bastante usable para un cliente que la utilice.

Capítulo 7: Mantenimiento

El estándar IEEE 1219 [IEEE,1993] define el mantenimiento del software como la modificación de un producto software después de haber sido entregado a los clientes con el fin de corregir defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio en el entorno [18].

Para ello podemos definir cuatro tipos de mantenimiento: correctivo, adaptivo, perfectivo y preventivo.

Por un lado el mantenimiento correctivo tiene como objetivo localizar y eliminar los posibles defectos de los programas. Podemos considerar como defecto una característica del sistema con el potencial de causar un fallo.

Además, tenemos el mantenimiento adaptativo que consiste en la modificación de un software debido a cambios en el entorno en el cual se ejecuta, ya sea el hardware o el software. Estos cambios pueden afectar desde al sistema operativo, pasando por la propia arquitectura del sistema informático o por el entorno de desarrollo del software.

También tenemos el mantenimiento perfectivo que es debido a los cambios en la especificación, normalmente debidos a posibles cambios en los requisitos del producto, implicando este tipo mantenimiento. Puede ser desde algo tan simple como un cambio de formato hasta algo más complicado como podría ser la incorporación de un nuevo módulo a la aplicación.

Por último, está el mantenimiento preventivo el cual consiste en la modificación del software para mejorar sus propiedades sin alterar las especificaciones funcionales iniciales. La solución pasaría por una revisión constante del software para detectar posibles problemas que surjan en el futuro.

Teniendo en cuenta estos mantenimientos mencionados anteriormente, algunos de ellos serán fácilmente de realizar por la labor realizada durante el diseño y el desarrollo de la aplicación. Sin embargo, siempre habrá que tener un mantenimiento ante cualquier problema que pueda surgir durante la vida del software.

Destacar que el mantenimiento adaptivo que puede interferir en el software. Sobre todo cuando algún componente que usa la aplicación se actualiza por algún problema de éste, hay que verificar que el software sigue funcionando perfectamente.

Capítulo 8: Conclusiones

En estos últimos meses, me ha afanado en seguir el ciclo de vida de esta aplicación. He tenido que adquirir conocimientos que desconocía y otros que aunque los había utilizado alguna vez, no había profundizado al nivel que he necesitado para la realización de este proyecto.

Este trabajo ha resultado más sencillo por los conocimientos adquiridos y consolidados durante los últimos años. Sin embargo, otros he tenido que buscar y consultar información sobre cómo utilizarlos. Un ejemplo claro de esto, ha sido la gestión del protocolo SMTP por el servidor Apache. Aunque puedes encontrar mucha información, incluso de software complementario para realizar tal empeño, el funcionamiento de éste ha sido enormemente arduo.

Asimismo, hay que tener en cuenta que la realización del proyecto por una sola persona, a veces ha sido una tarea de extrema dificultad. En gran parte debido a que a veces la visión que puedo tener sobre un problema puede ser errónea. Mientras que si estuviera en un grupo de trabajo que pueda tener distintos puntos de vista, ayuda a un desarrollo más correcto y efectivo. Por ello el trabajo colaborativo es una idea magnífica que ayuda a la realización de tareas, beneficiando ante posibles ideas desacertadas individuales.

El problema de no tener un equipo trabajando conjuntamente, se ha subsanado por un lado con la ayuda inestimable de mis tutores en todo momento y por otro lado con la búsqueda de información y la ayuda encontrada en algunos foros con problemas similares. Esto ha hecho que alguna vez encontrará como solucionar el problema planteado de otra forma igualmente o incluso más eficiente que la pensada inicialmente.

También cabe destacar, el uso de software libre ha sido gratificante. Por un lado por los conocimientos adquiridos durante su aprendizaje, por otro lado por la imperiosa necesidad de no realizar ningún gasto durante el desarrollo de este proyecto.

En cambio, la realización del proyecto desde cero ha sido una tarea complicada por el gran desarrollo de código. Sin embargo, ha sido estimulante por los conocimientos aprendidos y también por conocer en todo momento cada parte del software y de esta manera poder realizar cualquier modificación sin apenas dificultad.

En todo caso se ha conseguido plenamente los objetivos planificados al principio de este proyecto, tanto los objetivos específicos como los generales.

Por un lado se han cumplimentado todos los objetivos generales, que podemos resumir en: analizar las diferentes webs de venta de artículos, realizar un análisis de las posibles herramientas, diseñar un sitio donde los usuarios puedan compartir la compra e implementar los módulos necesarios para el correcto funcionamiento del sitio web.

Por otro lado también se ha conseguido los objetivos específicos de este proyecto y que han pasado por el diseño completo de la aplicación pudiendo hacer un pago colaborativo entre diferentes usuarios, gestionando en todo momento lo necesario para el funcionamiento correcto de la plataforma.

En relación a trabajos futuros, podrían ir encaminados en dos vías. Por un lado, podríamos mejorar la aplicación con varios módulos. Uno de ellos podría ser la gestión del llamado "backend", para que el administrador pudiera gestionar la aplicación de una forma más amigable como puede ser una aplicación y no tener que trabajar directamente a bajo nivel.

Por otro lado, la otra vía podría ser la utilización de parte de este software y usarlo para la creación de un módulo que pudiera ser manipulado aisladamente por otras aplicaciones. Es decir, un software que tuviera unos parámetros y que a partir de ellos se pudiera añadir a una plataforma web como otra función más.

Bibliografía

- [1] P. Graham, «<http://www.paulgraham.com/web20.html>,» 2005. Obtenido de: <http://www.paulgraham.com/web20.html>.
- [2] w3schools, «w3schools,» Diciembre 2015. Obtenido de: http://www.w3schools.com/browsers/browsers_stats.asp.
- [3] «World Wide Web Consortium,» Obtenido de: <https://www.w3.org/>.
- [4] «Web Accessibility Initiative (WAI),» Obtenido de: <https://www.w3.org/WAI/>.
- [5] GPL, «GPL,» Obtenido de: www.gnu.org/licenses/licenses.es.html.
- [6] «<http://www.4webs.es>,» Obtenido de: <http://www.4webs.es/blog/analisis-tiendas-online-2010-prestashop-oscommerce-magento>.
- [7] «osCommerce: Creating Online Stores Worldwide,» 2016. Obtenido de: <https://www.oscommerce.com/>.
- [8] «PrestaShop - Crea tu tienda online gratis,» 2015. Obtenido de: <https://www.prestashop.com/>.
- [9] «Magento: eCommerce Software & eCommerce Platform,» 2015. Obtenido de: <https://magento.com/>.
- [10] top500, «top500,» Obtenido de: <http://www.top500.org/>.
- [11] w3techs.com, «w3techs.com,» 2016. Obtenido de: <http://w3techs.com/technologies/details/os-linux/all/all>.
- [12] www.infoq.com, «www.infoq.com,» 2016. Obtenido de: <http://www.infoq.com/news/2013/11/genmyuml-designer>.
- [13] M. C. Bañuls, «<https://geekytheory.com>,» 2015. Obtenido de: <https://geekytheory.com/la-importancia-del-prototipado/>.
- [14] J. Nielsen, «10 Usability Heuristics for User Interface Design,» 1995. Obtenido de: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
- [15] «Sublime Text: The text editor you'll fall in love with,» 2015. Obtenido de: <https://www.sublimetext.com/>.

- [16] «PHP Code Checker - Syntax Check for Common PHP,» 2015. Obtenido de: <http://phpcodechecker.com/>.
- [17] «Validador CSS W3C,» 2015. Obtenido de: <https://jigsaw.w3.org/css-validator/>.
- [18] J. A. M. Párraga, «IEEE1219,» 1999. Obtenido de: alarcos.esi.uclm.es/per/fruiz/cur/mso/comple/IEEE1219.pdf.