



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

Gestión de una Ferretería

Alumno: Diego Gábana Bartolomé

Tutor: Luis Ignacio Sebastián Martín

Gestión de una Ferretería

Diego Gábana Bartolomé

15/07/2024

Índice general

Lista de figuras	V
Lista de tablas	VII
Resumen	XIII
Abstract	XV
I Memoria del Proyecto	1
1. Descripción del proyecto	3
1.1. Introducción	3
1.2. Objetivos del trabajo	3
1.2.1. Objetivos Generales	3
1.2.2. Objetivos Específicos	4
1.2.3. Alcance y Limitaciones	4
1.3. Entorno de aplicación	4
2. Metodología	7
2.1. Proceso de Desarrollo	7
2.2. Herramientas utilizadas	8
2.2.1. Lenguajes de Programación	8
2.2.2. Gestor de Bases de Datos	9
2.2.3. Entorno de Desarrollo	9
2.2.4. Servidor Local	10
2.2.5. Sistema Operativo	10
2.3. Arquitectura	11
2.3.1. Arquitectura Lógica	11
2.3.2. Arquitectura Física	12
3. Planificación	13
3.1. Estimación del esfuerzo	13
3.1.1. Estimación del Esfuerzo	13

3.1.2. Ventajas e Inconvenientes del Método	15
3.2. Planificación temporal	16
3.3. Presupuesto económico	21
3.3.1. Hardware y software	21
3.3.2. Recursos humanos	22
3.3.3. Presupuesto total	23
4. Conclusiones	25
4.1. Conclusión	25
4.2. Ampliaciones Futuras	26
II Documentación técnica	27
5. Análisis	29
5.1. Requisitos	29
5.1.1. Actores	29
5.1.2. Requisitos de negocio	29
5.1.3. Reglas de negocio	30
5.1.4. Requisitos de usuario	30
5.1.5. Requisitos funcionales	48
5.1.6. Requisitos de informacion	50
5.2. Atributos de calidad	50
6. Diseño	53
6.1. Diseño de datos	53
6.1.1. Modelo Entidad-Relación(ER)	53
6.1.2. Modelo relacional	53
6.2. Diagramas de clase y de secuencia	58
7. Implementación y arquitectura	59
7.1. Estructura del Proyecto	59
7.2. Descripción de los Componentes	60
7.2.1. Controladores	60
7.2.2. Modelos	61
7.2.3. Vistas	61
7.2.4. Otras Carpetas y Archivos	61
7.3. Detalles de Implementación	62
7.3.1. Controlador de Usuarios	62
7.3.2. Modelo de Usuario	65
7.4. Vistas	66
7.5. Conclusión	67

8. Pruebas	69
8.1. Estrategias de Prueba	69
8.2. Pruebas de Caja Negra	69
8.3. Pruebas de Caja Blanca	83
8.4. Pruebas de Seguridad	83
III Manuales de la Aplicación	85
9. Manual de Instalación	87
9.1. Prerrequisitos Técnicos	87
9.2. Instalación del Software de Terceros	87
9.2.1. Instalación de XAMPP	87
9.2.2. Instalación de Visual Studio Code	88
9.3. Instalación y Configuración del Servidor Web	89
9.3.1. Configuración de Apache	89
9.3.2. Configuración de MySQL	89
9.4. Despliegue de la Aplicación	90
10. Manual de Usuario	91
10.1. Inicio de Sesión	91
10.2. Navegación por el Panel de Usuario	92
10.3. Administrar Cuenta	93
10.4. Administrar productos	95
10.5. Administrar clientes	97
10.6. Administrar tienda	99
10.7. Carrito	99
10.8. Cerrar Sesión	102
11. Manual de Administración	103
11.1. Acceso al Panel de Administración	103
11.2. Gestión de Usuarios	103
11.3. Gestión de la Base de Datos	103
11.4. Configuración del Servidor	104
Bibliografía	105

Índice de figuras

2.1. Modelo espiral	8
2.2. Arquitectura lógica	11
2.3. Arquitectura física	12
3.1. Diagrama de Gantt	21
5.1. Casos de uso Subsistema gestionar usuarios.	35
5.2. Casos de uso Subsistema gestionar productos.	40
5.3. Casos de uso Subsistema gestionar clientes.	41
5.4. Casos de uso Subsistema gestionar tienda.	42
6.1. Diagrama entidad relación.	54
9.1. App Xampp.	88
9.2. App Visual Studio Code.	89
10.1. Iniciar sesión.	92
10.2. Iniciar sesión.	93
10.3. Editar información de la cuenta.	94
10.4. Cambiar contraseña.	94
10.5. Gestionar Productos.	96
10.6. Editar Productos.	96
10.7. Gestionar Clientes.	98
10.8. Editar Clientes.	98
10.9. Gestionar Tienda.	99
10.10 Gestionar Carrito.	101
10.11 Cerrar Sesión.	102
11.1. Importar/exportar base de datos.	104

Índice de cuadros

3.1. Equivalencia de los PFNA	14
3.2. Factores de Complejidad	15
3.3. Planificación temporal	18
3.4. Iteraciones	20
3.5. Detalle de Tareas y Subtareas	21
3.6. Hardware	22
3.7. Software	22
3.8. Horas recursos humanos	23
3.9. Presupuesto recursos humanos	23
3.10. Presupuesto total	23
5.1. Actores	29
5.2. Requisitos de negocio	29
5.3. Reglas de negocio	30
5.4. Requisitos de usuario	31
5.5. CU-01. Solicitar registro	32
5.6. CU-02. Aceptar registro	32
5.7. CU-03. Login	33
5.8. CU-04. Consultar app home	34
5.9. CU-05. Cambiar contraseña	34
5.10. CU-06. Agregar nuevo producto	36
5.11. CU-07. Buscar producto	37
5.12. CU-08. Editar producto	38
5.13. CU-09. Ordenar productos listados	39
5.14. CU-10. Agregar nuevo cliente	39
5.15. CU-11. Buscar cliente	43
5.16. CU-12. Editar cliente	44
5.17. CU-13. Ordenar clientes listados	45
5.18. CU-14. Añadir producto a carrito	45
5.19. CU-15. Gestionar carrito	46
5.20. CU-16. Actualizar stock de un producto (dentro del carrito)	47
5.21. CU-17. Seleccionar método de pago	48
5.22. Requisitos funcionales	49

5.23. Requisitos de eficiencia	49
5.24. Requisitos de usabilidad	49
5.25. Requisitos de mantenibilidad	49
5.26. Requisitos de seguridad	50
5.27. Requisitos de disponibilidad	50
5.28. Requisitos de implementacion	50
5.29. Requisitos de informacion	51
6.1. Tabla entidad clientes	55
6.2. Tabla entidad facturas	55
6.3. Tabla entidad productos	55
6.4. Tabla entidad usuarios	56
6.5. Tabla entidad ventas	56
6.6. Tabla relación tener	56
6.7. Tabla relación asociar	57
6.8. Tabla relación realizar	57
6.9. Tabla relación efectuar	57
6.10. Tabla relación contener	57
8.1. Prueba de caja negra - Registrar usuario	70
8.2. Prueba de caja negra - Iniciar Sesión	71
8.3. Prueba de caja negra - Cerrar Sesión	72
8.4. Prueba de caja negra - Fallo al iniciar Sesión	73
8.5. Prueba de caja negra - Buscar Producto	74
8.6. Prueba de caja negra - Añadir Producto al Carrito	75
8.7. Prueba de caja negra - Finalizar Compra	76
8.8. Prueba de caja negra - Actualizar Datos de Usuario	77
8.9. Prueba de caja negra - Eliminar Producto del Carrito	78
8.10. Prueba de caja negra - Cambiar Cantidad de Producto en el Carrito	79
8.11. Prueba de caja negra - Cambiar contraseña	80
8.12. Prueba de caja negra - Fallo Contraseña Nueva Cambiar contraseña	81
8.13. Prueba de caja negra - Fallo Contraseña Nueva Cambiar contraseña	82

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido fundamentales en la realización de este trabajo de fin de grado. En primer lugar, a mi novia, por su amor, paciencia y constante apoyo. Tu comprensión ha sido una fuente inagotable de motivación, y tu presencia ha sido crucial para superar este proyecto. Gracias por estar siempre a mi lado. A mi familia, por ser mi mayor fortaleza. A mis padres, por sus sacrificios, consejos y por siempre creer en mí. A mis hermanos, por su compañía y palabras de ánimo. Este logro es tanto mío como suyo, y estoy profundamente agradecido por su apoyo incondicional. Finalmente, a mi tutor, Luis Ignacio, que ha aceptado ayudarme y por su guía y orientación a lo largo de este proceso. Gracias por compartir su conocimiento y experiencia conmigo.

Resumen

En el presente documento se recoge el desarrollo de un sistema integral para la gestión de una ferretería, facilitando la gestión de inventarios, ventas y clientes. Con ello se pretende mejorar la eficiencia y organización del negocio, gracias al uso de tecnologías como PHP, MySQL y HTML/CSS en un entorno MVC, garantizando una estructura robusta y escalable. Gracias a este sistema los usuarios podrán iniciar sesión fácilmente así como gestionar productos y clientes o realizar ventas. La aplicación también incluye un carrito de la compra y la generación de facturas en PDF, garantizando una experiencia de usuario completa y segura.

Palabras claves: Gestión ferretería, inventario, clientes.

Abstract

This document contains the development of a comprehensive system for the management of a hardware store, facilitating the management of inventories, sales and customers. This is intended to improve the efficiency and organization of the business, thanks to the use of technologies such as PHP, MySQL and HTML/CSS in an MVC environment, ensuring a robust and scalable structure. Thanks to this system, users will be able to easily log in and manage products and customers or make sales. The application also includes a shopping cart and PDF invoice generation, ensuring a complete and secure user experience.

Keywords: Management hardware store, inventory, customers.

Parte I

Memoria del Proyecto

Capítulo 1

Descripción del proyecto

1.1. Introducción

En esta memoria desarrollaremos una aplicación de gestión para una ferretería, concebida con el propósito de optimizar y facilitar los procesos internos de este tipo de establecimientos. El impulso de este proyecto surge de la necesidad de modernizar y automatizar los procesos de gestión de inventarios, ventas y clientes en el contexto específico de una ferretería.

A raíz de las necesidades y particularidades del sector ferretero, surge esta aplicación tratando de ofrecer soluciones específicas que aborden los retos comunes a los que se enfrentan este tipo de negocios en su día a día. Además, incorpora funcionalidades innovadoras que buscan mejorar la eficiencia operativa y la experiencia del cliente. La implantación de esta aplicación está alineada con los objetivos estratégicos de la ferretería, que busca modernizar sus procesos y mejorar la satisfacción de sus clientes.

A lo largo de esta memoria se presentarán en detalle los diferentes aspectos del desarrollo de la aplicación de gestión para una ferretería. Para ello, abordaremos cuestiones como el diseño y la arquitectura del sistema, las funcionalidades implementadas, los retos encontrados y las soluciones propuestas. Con ello, pretendemos ofrecer una visión completa del proceso de desarrollo y sus resultados.

1.2. Objetivos del trabajo

1.2.1. Objetivos Generales

El principal objetivo de este trabajo es desarrollar una aplicación para la gestión de una ferretería, con el propósito de modernizar y optimizar los procesos internos de este tipo de establecimiento, tratando de optimizar tiempo y dinero en los procesos. Esta aplicación es una herramienta que abarca desde la gestión de inventario hasta el registro de ventas, así como la administración de clientes, con el fin de mejorar la eficiencia operativa y la experiencia del cliente.

1.2.2. Objetivos Específicos

- Diseñar e implementar un sistema de gestión de inventario que permita controlar el stock de los productos en la ferretería.
- Crear un sistema de ventas que incluya la búsqueda rápida de productos, la generación de facturas y el registro de pagos.
- Tener un sistema de gestión de clientes para poder tener controladas las facturas de cada compra realizada.
- Evaluar y realizar pruebas funcionales, con el objetivo de identificar posibles errores, mejoras y ajustes.

1.2.3. Alcance y Limitaciones

Las funcionalidades propias de la aplicación, se limitan a las descritas anteriormente, centrándose en la gestión del inventario, ventas y clientes. La principal finalidad de la aplicación es mejorar la eficiencia en el día a día del negocio, facilitando el control de stock, la gestión de ventas y la administración de la base de datos de clientes. Es importante señalar que, la aplicación cubre estas áreas fundamentales (las elegidas para la ferretería de una localidad concreta), no puede no abordar aspectos más específicos o personalizados que podrían variar según cada negocio en particular.

Las limitaciones de la aplicación incluyen la dependencia de un entorno tecnológico específico, como el uso de PHP y MySQL. Esto puede requerir conocimientos técnicos para su implementación y mantenimiento. Además, la aplicación, en su versión actual, no incluye funcionalidades avanzadas como análisis predictivo de inventario o integración con plataformas de ventas en línea, lo cual podría considerarse en futuras ampliaciones del proyecto.

1.3. Entorno de aplicación

El entorno de aplicación de este proyecto se sitúa en el ámbito de la gestión de pequeños y medianos negocios, particularmente ferreterías, que requieren una solución eficiente para manejar sus operaciones diarias. Existen diversas aplicaciones tecnológicas en el mercado que ofrecen funcionalidades similares, como sistemas de gestión de inventario, aplicaciones de punto de venta (POS) y software de gestión de relaciones con clientes (CRM).

Las aplicaciones de trabajo y tecnología existentes incluyen soluciones empresariales como SAP Business One, Odoo y QuickBooks, que ofrecen una amplia gama de herramientas para la gestión empresarial. Estas aplicaciones tienen la ventaja de estar altamente integradas y soportar múltiples módulos que van desde contabilidad, hasta gestión de recursos humanos. Sin embargo, presentan inconvenientes como costes elevados de implementación y mantenimiento, además de una considerable curva de aprendizaje debido a su complejidad.

Otra solución popular es el uso de plataformas de código abierto como ERPNext y Dolibarr, que permiten una mayor personalización y flexibilidad. Estas plataformas son más asequibles y permiten a los usuarios adaptarlas a sus necesidades específicas. Éstas requieren conocimientos técnicos para su configuración y mantenimiento, lo que puede suponer un obstáculo para las pequeñas empresas con recursos limitados.

El enfoque del TFG se justifica por la necesidad de ofrecer una solución más asequible y especializada, diseñada específicamente para las necesidades de una ferretería. A diferencia de las soluciones genéricas comerciales y de código abierto, esta aplicación se centra en resolver problemas específicos observados durante mi experiencia laboral en una ferretería, ofreciendo una interfaz simplificada y funcionalidades directamente alineadas con las operaciones diarias del negocio. Este enfoque no sólo hace que el sistema sea más fácil de adoptar y utilizar, sino que también permite futuras mejoras y personalizaciones basadas en los comentarios directos de los usuarios.

Capítulo 2

Metodología

En este capítulo se detallarán las metodologías y herramientas que se han utilizado para plantear el trabajo.

2.1. Proceso de Desarrollo

Para el desarrollo de este proyecto, se ha utilizado el modelo de desarrollo espiral. Este modelo es conocido por su enfoque iterativo que combina elementos del modelo de prototipos y del modelo en cascada, permitiendo un desarrollo más flexible y la incorporación de cambios y mejoras de manera continua.

El modelo espiral se compone de varias fases que se repiten en iteraciones, cada una de las cuales incluye planificación, evaluación de riesgos, desarrollo, y planeamiento. A lo largo del proyecto, hemos llevado a cabo un total de cuatro iteraciones, cada una de las cuales ha permitido refinar y mejorar el sistema en función de los comentarios recibidos y los análisis realizados.

Las iteraciones han sido estructuradas de la siguiente manera:

- **Primera Iteración:** En esta fase inicial, se realizó un análisis preliminar de los requisitos del sistema y se desarrolló un prototipo básico para evaluar la viabilidad del proyecto.
- **Segunda Iteración:** Basándonos en los comentarios obtenidos del prototipo inicial, se ampliaron y detallaron los requisitos. Se mejoró el diseño y se desarrolló una versión más completa del sistema.
- **Tercera Iteración:** Se enfocó en la integración de funcionalidades adicionales y en la resolución de problemas identificados en las etapas anteriores. También se llevaron a cabo pruebas más exhaustivas.
- **Cuarta Iteración:** En esta fase final, se realizaron los últimos ajustes y optimizaciones. Se llevaron a cabo pruebas de validación y se preparó el sistema para su implementación final.

Este enfoque iterativo ha permitido una mayor flexibilidad y adaptación a los cambios, asegurando que el sistema final cumpla con las expectativas y requisitos del usuario.

A continuación se muestra un 2.1 diagrama que refleja el modelo de desarrollo en espiral:



Figura 2.1: Modelo espiral

2.2. Herramientas utilizadas

En este apartado se detallan las tecnologías concretas que se han utilizado para desarrollar la aplicación, incluyendo los lenguajes de programación, el gestor de base de datos, las herramientas utilizadas en la planificación y generación de documentación y el sistema operativo empleado. Además, se expondrán los criterios por los que se han seleccionado estas herramientas, explicando sus ventajas e inconvenientes.

2.2.1. Lenguajes de Programación

Para el desarrollo de la aplicación, se ha optado por utilizar **PHP** y **HTML**. PHP es un lenguaje de programación ampliamente utilizado en el desarrollo web, especialmente para crear aplicaciones dinámicas y robustas. HTML se ha utilizado para estructurar el contenido de las páginas web.

■ Ventajas de PHP:

- Sencillo de aprender e integrar con HTML.
- Gran cantidad de bibliotecas y frameworks disponibles.

- Amplia comunidad y abundante documentación.

■ **Inconvenientes de PHP:**

- Menor velocidad de ejecución comparado con lenguajes compilados.
- Puede ser propenso a errores de seguridad si no se codifica correctamente.

2.2.2. Gestor de Bases de Datos

El gestor de bases de datos seleccionado para el proyecto es **MySQL**. MySQL es una base de datos relacional.

■ **Ventajas:**

- Alta fiabilidad y rendimiento.
- Soporte para transacciones ACID.
- Amplio soporte y documentación.

■ **Inconvenientes:**

- Puede requerir configuraciones avanzadas para optimización en casos de uso específicos.
- No es tan escalable como algunas alternativas NoSQL.

2.2.3. Entorno de Desarrollo

Para el desarrollo del proyecto se ha utilizado **Visual Studio Code** como editor de código. Visual Studio Code es un editor de código fuente desarrollado por Microsoft que incluye soporte para depuración, control de versiones Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

■ **Ventajas:**

- Interfaz intuitiva y personalizable.
- Amplia variedad de extensiones y plugins.
- Soporte para múltiples lenguajes de programación.

■ **Inconvenientes:**

- Requiere configuración inicial para un rendimiento óptimo.
- Puede ser pesado en términos de consumo de recursos para proyectos muy grandes.

2.2.4. Servidor Local

Para simular un entorno de servidor local, se ha utilizado **XAMPP**, que incluye **Apache** como servidor web. XAMPP es una distribución de Apache fácil de instalar que contiene MySQLy PHP.

▪ **Ventajas:**

- Instalación y configuración sencillas.
- Ideal para el desarrollo y prueba de aplicaciones web en local.
- Incluye todas las herramientas necesarias en un solo paquete.

▪ **Inconvenientes:**

- No recomendado para entornos de producción debido a posibles vulnerabilidades de seguridad.
- Puede consumir una cantidad significativa de recursos del sistema.

2.2.5. Sistema Operativo

El desarrollo del proyecto se ha llevado a cabo en **Windows**, un sistema operativo ampliamente utilizado que proporciona un entorno amigable para desarrolladores y una amplia compatibilidad con diferentes herramientas y software.

▪ **Ventajas:**

- Amplio soporte para software y hardware.
- Interfaz de usuario amigable y fácil de usar.
- Buen soporte para desarrollo de aplicaciones web y bases de datos.

▪ **Inconvenientes:**

- Coste de licencia.
- Puede ser menos estable que algunas alternativas de código abierto.

La elección de estas herramientas viene dada por la facilidad de uso, robustez y la capacidad de cumplir con los requisitos del proyecto de mejor manera.

2.3. Arquitectura

2.3.1. Arquitectura Lógica

La arquitectura lógica de nuestro sistema se organiza en varias capas, cada una de las cuales desempeña un papel crucial en el funcionamiento de la aplicación. A continuación, se describe cada una de estas capas:

Capa de Presentación (Cliente): Esta capa es la interfaz de usuario que interactúa con el sistema. Está compuesta por el navegador web del cliente, que utiliza HTML y CSS para presentar la información y la funcionalidad al usuario. El navegador web se encarga de mostrar las páginas web y de gestionar la interacción del usuario.

Capa de Aplicación (Servidor): La capa de aplicación se encuentra en el servidor y es responsable de procesar las solicitudes del cliente, ejecutar la lógica del negocio y generar respuestas dinámicas. En nuestro caso, utilizamos XAMPP como entorno de desarrollo que incluye Apache como servidor web y PHP como lenguaje de programación del lado del servidor. Apache recibe las solicitudes HTTP del navegador web y las pasa a los scripts PHP que ejecutan la lógica del negocio.

Capa de Acceso a Datos: Esta capa maneja la comunicación entre la aplicación y la base de datos. Utiliza PHP para ejecutar consultas SQL en la base de datos MySQL, recuperar datos, almacenarlos y actualizarlos según sea necesario. La capa de acceso a datos es necesaria para mantener la integridad y la eficiencia del almacenamiento de datos.

Capa de Datos (Base de Datos): La capa de datos está compuesta por la base de datos MySQL, que almacena toda la información necesaria para la aplicación. MySQL es un sistema de gestión de bases de datos relacional que permite el almacenamiento estructurado y la recuperación eficiente de los datos. La base de datos se organiza en tablas que contienen los registros de datos, y se accede a ella mediante consultas SQL.

A continuación se muestra un diagrama que refleja la arquitectura lógica 2.2:

Cliente
Navegador web HTML CSS
Servidor
XAMPP Apache PHP
Acceso a datos
MySQL

Figura 2.2: Arquitectura lógica

2.3.2. Arquitectura Física

La arquitectura física del sistema describe cómo los componentes del sistema están distribuidos físicamente y cómo se comunican entre sí. A continuación, se detalla cada uno de estos componentes:

Cliente: El cliente es un ordenador o dispositivo con un navegador web instalado. Este dispositivo puede ser un ordenador de escritorio o un portátil. El cliente interactúa con el sistema a través de la interfaz de usuario proporcionada por el navegador web.

Servidor: El servidor es una máquina configurada con XAMPP, que incluye Apache, PHP y MySQL. Este servidor puede ser un servidor físico dedicado o una máquina virtual en la nube. El servidor se encarga de procesar las solicitudes del cliente, ejecutar la lógica de la aplicación y acceder a la base de datos para recuperar y almacenar información.

A continuación se muestra un diagrama que refleja la arquitectura física 2.3:

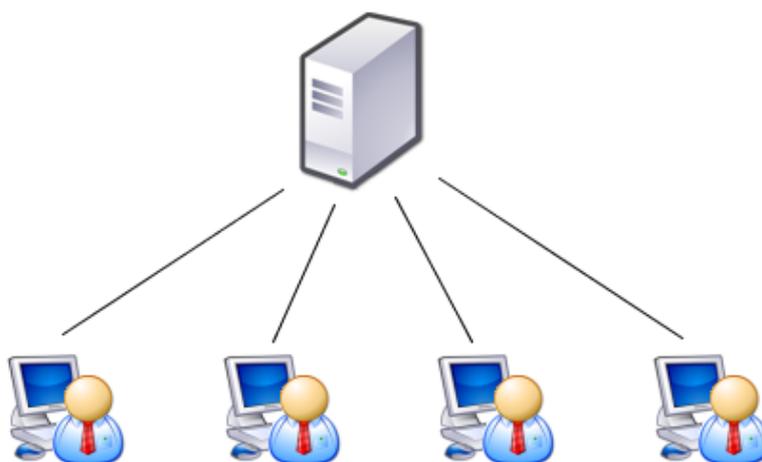


Figura 2.3: Arquitectura física

Capítulo 3

Planificación

En este capítulo se abordarán las cuestiones relativas a la planificación del trabajo.

3.1. Estimación del esfuerzo

Para la planificación temporal y de costes utilizaremos un método que consiste en la estimación de puntos de casos de uso y método de Albretch con puntos de función en el que hay que hacer un análisis previo de los requisitos funcionales de la aplicación.

Para estimar el esfuerzo por casos de uso, primero necesitamos asignar puntos de caso de uso a cada uno de los 25 casos de uso que hemos identificado. Luego, podemos utilizar esta información para calcular el esfuerzo total y la planificación temporal. Los puntos que vamos a asignar a cada caso de uso van a depender de su complejidad y tamaño.

3.1.1. Estimación del Esfuerzo

Determinación de los puntos de función sin ajustar (PFNA)

En el cuadro 3.1 se muestra la determinación de los puntos de función sin ajustar (PFNA):

Cálculo del Factor de Ajuste (FA)

En el cuadro ?? se muestra la cálculo del Factor de Ajuste (FA):

Calculamos el factor de ajuste (FA) aplicando la siguiente fórmula basada en la suma total de los factores de complejidad (FC):

$$FA = 0,65 + (0,01 \times FC)$$

$$FA = 0,65 + (0,01 \times 26) = 0,65 + 0,23 = 0,88$$

Dominio	Complejidad	Peso por complejidad	Nº total de funciones	Total
Entradas de usuario	Baja	x3	8	24
Entradas de usuario	Media	x4	5	20
Entradas de usuario	Alta	x6	0	0
Salidas externas	Baja	x4	0	0
Salidas externas	Media	x5	1	5
Salidas externas	Alta	x7	0	0
Consultas externas	Baja	x3	7	21
Consultas externas	Media	x4	6	24
Consultas externas	Alta	x6	4	24
Ficheros lógicos internos	Baja	x7	0	0
Ficheros lógicos internos	Media	x10	3	30
Ficheros lógicos internos	Alta	x15	0	0
Ficheros lógicos externos	Baja	x5	0	0
Ficheros lógicos externos	Media	x7	0	0
Ficheros lógicos externos	Alta	x10	0	0
Puntos de función sin ajustar (PFNA)				148

Cuadro 3.1: Equivalencia de los PFNA

Cálculo de los puntos de función ajustados (PFA)

A partir de este valor, se calculan los puntos de función ajustados (PFA):

$$PFA = PFNA \times FA = 148 \times 0,88 = 126,72$$

Cálculo del esfuerzo estimado

Para finalizar la estimación, calculamos la duración del proyecto utilizando el esfuerzo personal en horas. Establecemos la equivalencia de 3 horas por punto de función ajustado:

$$\text{Duración del proyecto estimada} = PFA \times 3$$

$$\text{Duración del proyecto estimada} = 126,72 \times 3 = 380,16 \text{ horas}$$

Factores de complejidad	Peso
Comunicación de datos	3
Prestaciones (Rendimiento)	2
Velocidad (Frecuencia) de las transacciones	2
Diseño de la eficiencia del usuario final	3
Complejidad del proceso lógico interno	2
Facilidad de instalación	1
Localizaciones múltiples	1
Funciones distribuidas	0
Gran uso de la configuración	2
Actualizaciones online de datos	2
Reusabilidad	1
Facilidad de operación	2
Facilidad de cambio	2
Factor de Complejidad Total	23

Cuadro 3.2: Factores de Complejidad
??

3.1.2. Ventajas e Inconvenientes del Método

Ventajas

- Estándar Internacional: El método de Albrecht está bien estandarizado y es reconocido internacionalmente, lo que proporciona una base confiable y consistente para la estimación.
- Precisión Técnica: Al considerar factores técnicos como entradas, salidas, consultas, archivos internos y externos, ofrece una estimación más precisa del esfuerzo técnico requerido.
- Adaptabilidad: Puede ser ajustado para diferentes tipos de proyectos a través del uso de factores de ajuste que toman en cuenta la complejidad y otras características del proyecto.
- Comparabilidad: Facilita la comparación entre diferentes proyectos y organizaciones debido a su estandarización y la claridad de sus métricas.

Inconvenientes

- **Complejidad del Proceso:** La evaluación y categorización de todos los elementos técnicos puede ser compleja y consumir mucho tiempo.
- **Necesidad de Experiencia:** Requiere evaluadores con experiencia y conocimiento profundo en el método para obtener estimaciones precisas.
- **Rigidez:** Puede ser menos flexible para ajustarse a cambios en los requisitos o al descubrimiento de nuevos elementos durante el desarrollo.
- **Costoso en Términos de Recursos:** El proceso detallado de contar puntos de función y aplicar factores de ajuste puede requerir una inversión significativa en términos de tiempo y recursos.

3.2. Planificación temporal

La planificación temporal del proyecto tendrá en cuenta los objetivos y requisitos de la aplicación, así como la estimación del esfuerzo. Se distingue en tareas de análisis, diseño, implementación y pruebas, organizadas en las posibles iteraciones del ciclo de vida del proyecto. Además, se considerará el marco temporal para la entrega del proyecto.

1. Análisis

Objetivo: Comprender y documentar los requisitos del sistema.

Duración: 57 horas (15 % del total estimado)

- **Recolección de Requisitos (19 horas)**
 - Entrevistas con stakeholders
 - Revisión de documentación existente
- **Análisis de Requisitos (19 horas)**
 - Definición de casos de uso
 - Modelado de procesos
- **Revisión y Aprobación de Requisitos (19 horas)**
 - Revisión con stakeholders
 - Aprobación de documentos de requisitos

2. Diseño

Objetivo: Crear una arquitectura y diseño detallado del sistema.

Duración: 85 horas (22.5 % del total estimado)

- **Diseño de Datos (28 horas)**
 - Creación del modelo entidad-relación
 - Definición del diccionario de datos
- **Diseño de Interfaz de Usuario (28 horas)**
 - Mockups y wireframes
 - Prototipos de alta fidelidad
- **Diseño de Arquitectura (29 horas)**
 - Definición de la arquitectura del sistema
 - Diseño de componentes y módulos

3. Implementación

Objetivo: Desarrollar y codificar los componentes del sistema.

Duración: 143 horas (37.5 % del total estimado)

- **Configuración del Entorno de Desarrollo (10 horas)**
 - Instalación de herramientas
 - Configuración de repositorios
- **Desarrollo de Componentes Principales (95 horas)**
 - Codificación de módulos y funcionalidades
 - Integración de componentes
- **Desarrollo de Interfaz de Usuario (38 horas)**
 - Implementación de la UI
 - Pruebas de usabilidad

4. Pruebas

Objetivo: Verificar y validar que el sistema cumpla con los requisitos.

Duración: 66 horas (17.5 % del total estimado)

- **Pruebas (66 horas)**
 - Creación de casos de prueba
 - Ejecución de pruebas unitarias

5. Documentación y Entrega

Objetivo: Documentar el sistema y preparar la entrega final.
Duración: 29 horas (7.5 % del total estimado)

- **Documentación del Código (9 horas)**
 - Comentarios y documentación inline
 - Manual de referencia para desarrolladores

- **Documentación del Usuario (10 horas)**
 - Manuales de usuario
 - Guías de instalación y configuración

- **Preparación y Entrega (10 horas)**
 - Compilación de entregables
 - Revisión final y presentación

Resumen de la Planificación Temporal

En el cuadro 3.3 se muestra un resumen de la Planificación Temporal:

Fase	Duración (horas)	Porcentaje del Total
Análisis	57	15 %
Diseño	85	22.5 %
Implementación	143	37.5 %
Pruebas	66	17.5 %
Documentación y Entrega	29	7.5 %
Total	380	100 %

Cuadro 3.3: Planificación temporal

Iteraciones del Ciclo de Vida del Proyecto

El proyecto se dividirá en varias iteraciones, cada una de las cuales incluirá tareas de análisis, diseño, implementación y pruebas para un subconjunto de los requisitos. Esta aproximación iterativa e incremental asegura que el proyecto se desarrolle de manera controlada y se ajusten los entregables según el feedback recibido.

Iteración 1: Análisis y Diseño Inicial

Duración: 4 semanas

Actividades: Recolección y análisis de requisitos, diseño de datos y arquitectura inicial.

Iteración 2: Desarrollo Inicial y Pruebas

Duración: 4 semanas

Actividades: Desarrollo de componentes principales, pruebas.

Iteración 3: Desarrollo Avanzado y Pruebas

Duración: 4 semanas

Actividades: Desarrollo de la interfaz de usuario, pruebas.

Iteración 4: Desarrollo, Pruebas del Sistema y Documentación

Duración: 4 semanas

Actividades: Desarrollo, pruebas del sistema, documentación del código y del usuario.

Iteración 5: Revisión Final y Entrega

Duración: 2 semanas

Actividades: Documentación, preparación de la entrega final, revisión y presentación del proyecto.

Resumen de Iteraciones

En el cuadro 3.4 se muestra un resumen de las iteraciones:

Iteración	Duración (semanas)	Actividades Principales
Iteración 1: Análisis y Diseño Inicial	4	Recolección y análisis de requisitos, diseño de datos y arquitectura inicial
Iteración 2: Desarrollo Inicial y Pruebas	4	Desarrollo de componentes principales, pruebas

Iteración 3: Desarrollo Avanzado y Pruebas	4	Desarrollo de la interfaz de usuario, pruebas
Iteración 4: Desarrollo , Pruebas del Sistema y Documentación	4	Pruebas del sistema, documentación del código y del usuario
Iteración 5: Docuemntación, Revisión Final y Entrega	2	Documentación, preparación de la entrega final, revisión y presentación

Cuadro 3.4: Iteraciones

Detalle de Tareas y Subtareas

En el cuadro 3.5 se muestra un detalle de tareas y subtareas:

Fase	Tarea	Duración (horas)	Iteración
Análisis	Recolección de Requisitos	19	Iteración 1
	Análisis de Requisitos	19	Iteración 1
	Revisión y Aprobación de Requisitos	19	Iteración 1
Diseño	Diseño de Datos	28	Iteración 1
	Diseño de Interfaz de Usuario	28	Iteración 1
	Diseño de Arquitectura	29	Iteración 1
Implementación	Configuración del Entorno de Desarrollo	10	Iteración 2
	Desarrollo de Componentes Principales	101	Iteración 2 y 3
	Desarrollo de Interfaz de Usuario	38	Iteración 3
Pruebas	Pruebas	30	Iteración 2
	Pruebas de Usuario	30	Iteración 4 y 5
Documentación y Entrega	Documentación del Código	9	Iteración 4
	Documentación del Usuario	10	Iteración 4

Preparación y Entrega	10	Iteración 5
-----------------------	----	-------------

Cuadro 3.5: Detalle de Tareas y Subtareas

Conclusión

Esta planificación temporal distribuye de manera equitativa el esfuerzo estimado de 380 horas a lo largo de 18 semanas, con iteraciones bien definidas que permiten un desarrollo controlado y la obtención de feedback continuo. Este enfoque garantiza que se cumplan los objetivos del proyecto dentro del plazo establecido.

El correspondiente diagrama de Gantt 3.1 permite observar de forma gráfica la distribución temporal de las tareas:

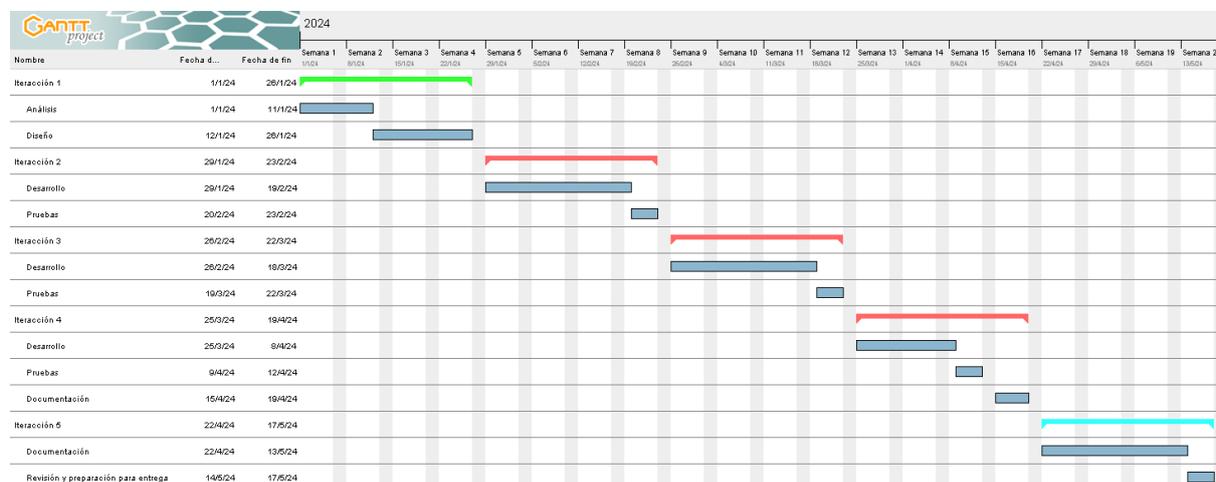


Figura 3.1: Diagrama de Gantt

3.3. Presupuesto económico

3.3.1. Hardware y software

En el cuadro 3.6 se muestra un resumen del presupuesto hardware:

Aclaraciones

Fibra 600GB: se usa 8h al día durante 20 días al mes, es decir, un 22,22 % del mes. Si lo multiplicamos por 4 meses de uso nos da un 88.88 %. Si hacemos un 88.88 % de 38,30€ = 34,04€

En el cuadro 3.7 se muestra un resumen del presupuesto software:

Componente	Precio	Tiempo de vida	Porcentaje de uso	Tiempo de uso	Coste real en proyecto
Ordenador	800,00 €	48 meses	8,3 %	4 meses	66,40 €
Raton y teclado	50,00 €	48 meses	8,3 %	4 meses	4,15 €
Monitor	150,00 €	48 meses	8,3 %	4 meses	12,45 €
Router	0 €			4 meses	0,00 €
Fibra 600 GB	38.30 €/mes		22,22 % del mes	4 meses	34,04 €
Total					117,04 €

Cuadro 3.6: Hardware

Componente	Precio	Tiempo de vida	Porcentaje de uso	Tiempo de uso	Coste real en proyecto
Windows 11	0,00 €			4 meses	0,00 €
Office 365	69,99 €	12 meses	10 %	4 meses	6,90 €
Visual Studio Code	0.00 €			4 meses	0,00 €
XAMPP	0,00 €			4 meses	0,00 €
Google Chrome	0,00 €			4 meses	0,00 €
StarUML	0,00 €			4 meses	0,00 €
Total					6,90 €

Cuadro 3.7: Software

3.3.2. Recursos humanos

El proyecto es llevado a cabo por una sola persona que asume todos los roles necesarios. En este apartado, se describen brevemente estos roles y sus costes en el mercado actual.

Los roles establecidos son: ingeniero analista (para tareas de análisis y diseño), ingeniero *fullstack* (para tareas de implementación), y el ingeniero de pruebas (para tareas de diseño y ejecución de pruebas).

Cada rol tiene una jornada laboral semanal de 40 horas. Con esta base, calculamos el salario por hora para cada rol (los salarios mostrados son netos).

Dado que cada rol se encarga de documentar su trabajo, las horas de documentación se distribuyen proporcionalmente según el número total de horas.

En el cuadro 3.8 se muestra un resumen de las horas de recursos humanos:

En el cuadro 3.9 se muestra un resumen del presupuesto de las horas de recursos humanos:

3.3. Presupuesto económico

Rol	Horas de Documentación	Horas Totales
Analista	15,52 horas	92,52 horas
Programador <i>fullstack</i>	17,76 horas	232,76 horas
Ingeniero de Pruebas	6,72 horas	54,72 horas

Cuadro 3.8: Horas recursos humanos

Rol	Salario Mensual	Salario por Hora	Horas	Coste Total
Analista	1965,60 €	12,28 €	92,52	1136,10 €
Programador <i>fullstack</i>	2066,50 €	12,90 €	232,76	3002,60 €
Ingeniero de Pruebas	1793,50 €	11,20 €	54,72	612,86 €
Total				4751,56 €

Cuadro 3.9: Presupuesto recursos humanos

3.3.3. Presupuesto total

Como resultado de nuestra estimación, podemos ver el cuadro 3.10 en el que se muestra la suma de hardware, software y RRHH.

	Hardware	Software	RRHH	Total
Coste total estimación	117,04 €	6,90 €	4751,56 €	4875,50 €

Cuadro 3.10: Presupuesto total

Capítulo 4

Conclusiones

4.1. Conclusión

La realización de este Proyecto Fin de Grado ha sido una experiencia profundamente enriquecedora y significativa. Este proyecto me ha permitido alcanzar los objetivos académicos propuestos, lo cual provoca un gran impacto personal, al dar respuesta a una necesidad real observada en mi entorno.

Durante un largo período de tiempo, desarrollé labores de trabajo en una ferretería, detectando la desorganización y las dificultades para llevar un control del stock. Este reto me inspiró a crear una aplicación que pudiera facilitar la gestión diaria de una ferretería. A lo largo del desarrollo, me enfrenté a varios retos técnicos y de diseño, que fui superando poco a poco hasta dar lugar a este documento.

Inicialmente, intenté desarrollar la aplicación con Salesforce, pero pronto me di cuenta de que no ofrecía las ventajas que buscaba. Finalmente, opté por utilizar PHP, HTML, y otras tecnologías que me permitieron construir una solución más ajustada a las necesidades específicas de la ferretería. Este cambio de rumbo fue crucial para el éxito del proyecto y me enseñó la importancia de la flexibilidad y la adaptabilidad en el desarrollo de software.

El TFG ha sido una oportunidad para aplicar y expandir mis conocimientos adquiridos durante la carrera, permitiéndome trabajar en aspectos prácticos que no había explorado anteriormente. Desarrollar esta aplicación me ha enseñado a enfrentar problemas del mundo real, a gestionar mejor mi tiempo y a mejorar mis habilidades de programación y diseño de sistemas.

En conclusión, este proyecto no solo ha cumplido con los objetivos técnicos y académicos establecidos, sino que también ha sido un viaje personal lleno de aprendizaje y crecimiento. Agradezco la oportunidad de haber podido desarrollar una solución que ojalá pudiera ver hacer una diferencia real en la gestión de negocios locales.

4.2. Ampliaciones Futuras

A continuación, se presentan algunas ideas de ampliaciones futuras que podrían añadir valor significativo a la aplicación desarrollada:

- **Integración de análisis de datos para la previsión de stock:** Implementar un sistema de análisis de datos que permita prever la necesidad de stock basándose en patrones históricos de ventas. Esto ayudaría a mantener niveles óptimos de inventario y a evitar tanto el exceso como la falta de productos.
- **Sistema de alertas para productos de baja rotación:** Crear un sistema de alertas que notifique al usuario sobre productos que llevan mucho tiempo sin venderse. Esto podría ayudar a tomar decisiones sobre promociones o liquidaciones.
- **Recepción de pedidos de fabricantes:** Añadir una sección para gestionar la recepción de pedidos de fabricantes. Permitiría escanear el código de barras de los productos recibidos e introducir la cantidad nueva, actualizando automáticamente el stock en la base de datos.
- **Integración con otros sistemas de gestión:** Implementar APIs para integrar la aplicación con otros sistemas de gestión empresarial, como contabilidad y recursos humanos, para proporcionar una solución más completa y unificada.
- **Optimización de la interfaz de usuario:** Continuar mejorando la interfaz de usuario basándose en el feedback de los usuarios, para asegurar que sea lo más intuitiva y eficiente posible.

Estas ampliaciones no solo mejorarán la funcionalidad de la aplicación, sino que también proporcionarán herramientas avanzadas para la gestión eficiente y efectiva de una ferretería, adaptándose a las necesidades cambiantes del mercado y de los usuarios.

Parte II

Documentación técnica

Capítulo 5

Análisis

5.1. Requisitos

5.1.1. Actores

En el cuadro 5.1 se muestran los diferentes actores de la app:

ID	Nombre	Descripción
ACT-01	Administrador	Usuario registrado que previamente ha iniciado sesión y tiene permisos ilimitados
ACT-02	Usuario registrado	Usuario registrado que previamente ha iniciado sesión

Cuadro 5.1: Actores

5.1.2. Requisitos de negocio

En el cuadro 5.2 se muestran los diferentes requisitos de negocio de la app:

ID	Requisito
RN-01	Crear un sistema que sirva para optimizar las ventas, stock y facturas de una ferretería
RN-02	Mejorar la trazabilidad entre clientes, ventas de productos y facturas
RN-03	Crear y guardar una factura de cada venta

Cuadro 5.2: Requisitos de negocio

5.1.3. Reglas de negocio

En el cuadro 5.3 se muestran las diferentes reglas de negocio de la app:

ID	Regla
RgN-01	La aplicación no dejará acceder a ningún usuario registrado que no lo haya aprobado el administrador
RgN-02	La aplicación no puede tener más de un administrador
RgN-03	La aplicación debe generar una factura por cada compra
RgN-03	La aplicación debe llevar un registro del stock de cada producto

Cuadro 5.3: Reglas de negocio

5.1.4. Requisitos de usuario

En el cuadro 5.4 se muestran los diferentes requisitos de usuario de la app:

ID	Nombre	Descripción
RU-01	Iniciar sesion	El usuario inicia sesion en la aplicación
RU-02	Cerrar sesión	El usuario cierra sesion en la aplicación
RU-03	Registrar usuario	El usuario se registra como nuevo usuario en la aplicación
RU-04	Consultar app home	El usuario accede a la página principal de la aplicación
RU-05	Gestionar cuenta de usuario	El usuario accede a su página de perfil para poder ver/modificar sus datos
RU-06	Cambiar contraseña	El usuario cambia la contraseña de acceso a la aplicación
RU-07	Agregar nuevo producto	El usuario agrega un nuevo producto a la base de datos
RU-08	Buscar producto	El usuario busca un producto mediante código de barras o nombre en la base de datos
RU-09	Editar producto	El usuario edita un producto de la aplicación
RU-10	Ordenar productos listados	El usuario ordena los productos que aparecen en la lista de la página de productos
RU-11	Agregar nuevo cliente	El usuario agrega un nuevo cliente a la base de datos
RU-12	Buscar cliente	El usuario busca un cliente mediante código de barras o nombre en la base de datos
RU-13	Editar cliente	El usuario edita un cliente de la aplicación

ID	Nombre	Descripción
RU-14	Ordenar clientes listados	El usuario ordena los clientes que aparecen en la lista de la página de productos
RU-15	Añadir producto a carrito	El usuario añade los productos que quiera al carrito
RU-16	Gestionar carrito	El usuario accede a la página del carrito
RU-17	Actualizar stock de producto	El usuario modifica las unidades de un producto añadido al carrito
RU-18	Seleccionar método de pago	El usuario selecciona el método de pago (efectivo o tarjeta)
RU-19	Buscar cliente por DNI o teléfono	El usuario busca un cliente por DNI o teléfono para asociarle la compra
RU-20	Seleccionar cliente	El usuario selecciona un cliente buscado anteriormente
RU-21	Finalizar compra	El usuario finaliza la compra
RU-22	Gestionar productos	El usuario accede a la página de productos
RU-23	Gestionar clientes	El usuario accede a la página de clientes
RU-24	Gestionar tienda	El usuario accede a la página de la tienda
RU-25	Gestionar carrito	El usuario accede a la página del carrito

Cuadro 5.4: Requisitos de usuario

Modelo de casos de uso

En la figura 5.1 se muestra el caso de uso del subsistema gestionar usuario:

En la figura 5.2 se muestra el caso de uso del subsistema gestionar usuario:

En la figura 5.3 se muestra el caso de uso del subsistema gestionar usuario:

En la figura 10.9 se muestra el caso de uso del subsistema gestionar usuario:

Especificación de casos de uso

En el cuadro 5.5 se muestra el caso de uso CU-01. Solicitar registro:

Nombre e ID del CU	CU-01. Solicitar registro
Actor	Usuario

Descripción	El usuario envía sus datos de usuario al sistema, generando una solicitud de alta que deberá ser gestionada por un administrador.
Precondiciones	PRE-1. El usuario no está identificado en el sistema.
Postcondiciones	POST-1. La solicitud queda almacenada en el sistema.
Flujo normal	<p>FN1 El actor introduce sus datos de usuario.</p> <p>FN2 El sistema comprueba los datos introducidos.</p> <p>FN3 Si los datos son correctos, el sistema almacena la solicitud de registro e informa al administrador para que lo apruebe.</p>
Flujo alternativo 1	<p>FA1 Si los datos son incorrectos se informa al usuario del error y no se procede a almacenar la solicitud.</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p> <p>E2 El usuario o correo electrónico ya existen.</p>
Prioridad	Alta

Cuadro 5.5: CU-01. Solicitar registro

En el cuadro 5.6 se muestra el caso de uso CU-02. Aceptar registro:

Nombre e ID del CU	CU-02. Aceptar registro
Actor	Usuario admin
Descripción	El administrador acepta el registro de un usuario para que pueda acceder a la aplicación
Precondiciones	PRE-1. Un usuario ha solicitado registro en la aplicación.
Postcondiciones	POST-1. El usuario queda registrado y puede acceder a la aplicación
Flujo normal	<p>FN1 El actor comprueba y acepta el usuario preregistrado.</p>
Prioridad	Alta

Cuadro 5.6: CU-02. Aceptar registro

En el cuadro 5.7 se muestra el caso de uso CU-03. Login:

Nombre e ID del CU	CU-03. Login
---------------------------	--------------

Actor	Usuario
Descripción	El introduce sus credenciales para poder acceder a la aplicación
Precondiciones	PRE-1. Un usuario debe haber sido registrado en la aplicación.
Postcondiciones	POST-1. El usuario es reconocido y accede a la aplicación
Flujo normal	<p>FN1 El actor introduce sus credenciales.</p> <p>FN2 El sistema comprueba dichas credenciales comparandolas con la base de datos</p> <p>FN3 Si los datos son correctos, el sistema deja acceder al usuario a la aplicación</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p> <p>E2 El usuario o la contraseña no coinciden con ningún usuario en la base de datos.</p>
Prioridad	Alta

Cuadro 5.7: CU-03. Login

En el cuadro 5.8 se muestra el caso de uso CU-04. Consultar app home:

Nombre e ID del CU	CU-04. Consultar app home
Actor	Usuario
Descripción	El introduce sus credenciales para poder acceder a la aplicación
Precondiciones	PRE-1. Un usuario debe haber sido logueado en la aplicación.
Postcondiciones	POST-1. El usuario es reconocido y accede a la aplicación
Flujo normal	<p>FN1 El actor introduce sus credenciales.</p> <p>FN2 El sistema comprueba dichas credenciales comparandolas con la base de datos</p> <p>FN3 Si los datos son correctos, el sistema deja acceder al usuario a la aplicación</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p> <p>E2 El usuario o la contraseña no coinciden con ningún usuario en la base de datos.</p>
Prioridad	Alta

Cuadro 5.8: CU-04. Consultar app home

En el cuadro 5.9 se muestra el caso de uso CU-05. Cambiar contraseña:

Nombre e ID del CU	CU-05. Cambiar contraseña
Actor	Usuario
Descripción	El usuario modifica su contraseña actual por una nueva contraseña.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema.
Postcondiciones	POST-1. La contraseña del usuario se actualiza en el sistema.
Flujo normal	<p>FN1 El actor accede a la configuración de su cuenta.</p> <p>FN2 El sistema solicita la contraseña actual del usuario como medida de seguridad.</p> <p>FN3 El usuario introduce su contraseña actual.</p> <p>FN4 El sistema verifica que la contraseña introducida coincida con la contraseña almacenada.</p> <p>FN5 Si la contraseña es correcta, el sistema permite al usuario ingresar su nueva contraseña.</p> <p>FN6 El usuario introduce y confirma su nueva contraseña.</p> <p>FN7 El sistema actualiza la contraseña del usuario en la base de datos.</p>
Flujo alternativo 1	<p>FA1 Si la contraseña actual ingresada por el usuario no coincide con la contraseña almacenada, se muestra un mensaje de error y se vuelve al paso 2.</p>
Excepciones	<p>E1 El usuario deja campos requeridos sin rellenar.</p> <p>E2 La nueva contraseña no cumple con los criterios de seguridad establecidos.</p>
Prioridad	Media

Cuadro 5.9: CU-05. Cambiar contraseña

En el cuadro 5.10 se muestra el caso de uso CU-06. Agregar nuevo producto:

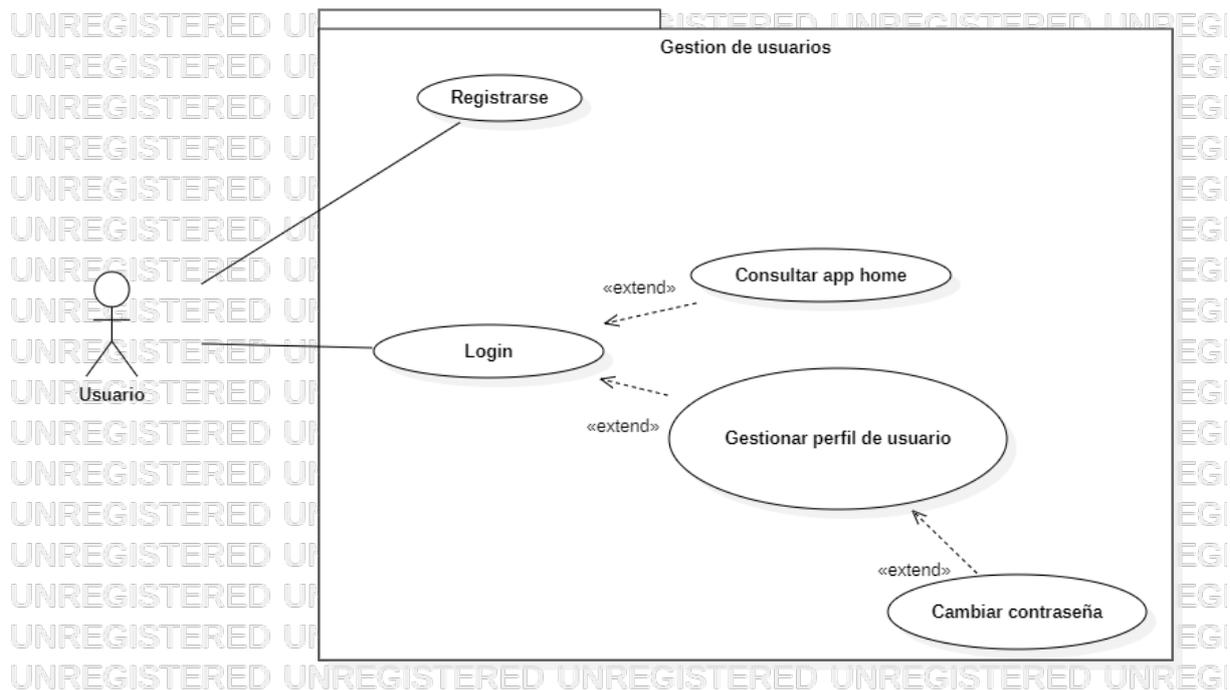


Figura 5.1: Casos de uso Subsistema gestionar usuarios.

Nombre e ID del CU	CU-06. Agregar nuevo producto
Actor	Usuario
Descripción	El usuario añade un nuevo producto al catálogo de la ferretería.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema.
Postcondiciones	POST-1. El nuevo producto se registra en la base de datos del sistema.
Flujo normal	<p>FN1 El actor accede a la opción de agregar nuevo producto en el panel de administración.</p> <p>FN2 El sistema muestra un formulario para ingresar los detalles del nuevo producto.</p> <p>FN3 El usuario administrador completa los campos del formulario (nombre, descripción, precio, cantidad en stock, etc.).</p> <p>FN4 El usuario confirma la creación del nuevo producto.</p> <p>FN5 El sistema verifica que todos los campos obligatorios estén completos y que la información proporcionada sea válida.</p> <p>FN6 Si los datos son válidos, el sistema registra el nuevo producto en la base de datos.</p>

Flujo alternativo 1	FA1 Si algún campo obligatorio no se completa o la información proporcionada no es válida, se muestra un mensaje de error y se vuelve al paso 2.
Excepciones	E1 El usuario deja campos requeridos sin rellenar. E2 Los datos ingresados no son válidos según las restricciones del sistema.
Prioridad	Alta

Cuadro 5.10: CU-06. Agregar nuevo producto

En el cuadro 5.11 se muestra el caso de uso CU-07. Buscar producto:

Nombre e ID del CU	CU-07. Buscar producto
Actor	Usuario
Descripción	El usuario busca un producto en el catálogo de la ferretería utilizando criterios de búsqueda específicos.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema.
Postcondiciones	POST-1. Se muestra una lista de productos que coinciden con los criterios de búsqueda.
Flujo normal	FN1 El actor accede a la opción de búsqueda de productos en el panel de administración o en la interfaz de usuario. FN2 El sistema muestra un formulario o campo de búsqueda donde el usuario puede ingresar los criterios de búsqueda (nombre, categoría, precio, etc.). FN3 El usuario ingresa los criterios de búsqueda y solicita al sistema realizar la búsqueda. FN4 El sistema realiza la búsqueda en la base de datos utilizando los criterios proporcionados. FN5 El sistema muestra una lista de productos que coinciden con los criterios de búsqueda.
Flujo alternativo 1	FA1 Si no se encuentra ningún producto que coincida con los criterios de búsqueda, se muestra un mensaje indicando que no se encontraron resultados.

Excepciones	<p>E1 El usuario no proporciona ningún criterio de búsqueda.</p> <p>E2 No se encuentran productos que coincidan con los criterios de búsqueda.</p>
Prioridad	Media

Cuadro 5.11: CU-07. Buscar producto

En el cuadro 5.12 se muestra el caso de uso CU-08. Editar producto:

Nombre e ID del CU	CU-08. Editar producto
Actor	Usuario
Descripción	El usuario realiza la edición de los detalles de un producto en el catálogo de la ferretería.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema.
Postcondiciones	POST-1. Los detalles del producto se actualizan en la base de datos.
Flujo normal	<p>FN1 El usuario accede a la opción de edición de productos en su perfil de usuario.</p> <p>FN2 El sistema muestra una lista de productos disponibles para editar.</p> <p>FN3 El usuario selecciona el producto que desea editar de la lista.</p> <p>FN4 El sistema muestra un formulario prellenado con los detalles actuales del producto.</p> <p>FN5 El usuario modifica los detalles del producto según sea necesario.</p> <p>FN6 El usuario confirma los cambios y solicita al sistema guardar la actualización.</p> <p>FN7 El sistema valida los datos ingresados por el usuario.</p> <p>FN8 Si los datos son válidos, el sistema actualiza los detalles del producto en la base de datos y muestra un mensaje de confirmación.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide cancelar la edición en cualquier momento, el sistema descarta los cambios y vuelve a la lista de productos.</p>

Excepciones	<p>E1 El usuario deja campos requeridos sin completar.</p> <p>E2 Los datos ingresados por el usuario no son válidos (por ejemplo, un precio negativo).</p>
Prioridad	Media

Cuadro 5.12: CU-08. Editar producto

En el cuadro 5.13 se muestra el caso de uso CU-09. Ordenar productos listados:

Nombre e ID del CU	CU-09. Ordenar productos listados
Actor	Usuario
Descripción	El usuario ordena la lista de productos en función de algún criterio específico, como nombre, precio, disponibilidad, etc.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y ha accedido a la lista de productos.
Postcondiciones	POST-1. La lista de productos se muestra ordenada según el criterio seleccionado por el usuario.
Flujo normal	<p>FN1 El usuario accede a la opción de ordenar productos en la interfaz de usuario.</p> <p>FN2 El sistema muestra una lista desplegable con los criterios de ordenamiento disponibles, como nombre, precio, disponibilidad, etc.</p> <p>FN3 El usuario selecciona un criterio de ordenamiento de la lista desplegable.</p> <p>FN4 El sistema reorganiza la lista de productos según el criterio seleccionado por el usuario y la muestra actualizada.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide cancelar la acción de ordenar productos en cualquier momento, el sistema mantiene el orden actual de la lista y no realiza cambios.</p>
Excepciones	<p>E1 No hay productos disponibles para ordenar.</p> <p>E2 Error en el sistema al procesar la solicitud de ordenamiento.</p>
Prioridad	Media

Cuadro 5.13: CU-09. Ordenar productos listados

En el cuadro 5.14 se muestra el caso de uso CU-10. Agregar nuevo cliente:

Nombre e ID del CU	CU-10. Agregar nuevo cliente
Actor	Usuario
Descripción	El usuario añade un nuevo cliente al sistema introduciendo sus datos personales, como nombre, dirección, número de teléfono, etc.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene los permisos necesarios para agregar clientes.
Postcondiciones	POST-1. El nuevo cliente queda registrado en el sistema y puede ser gestionado y consultado posteriormente.
Flujo normal	<p>FN1 El usuario accede a la opción de agregar nuevo cliente en la interfaz de usuario.</p> <p>FN2 El sistema muestra un formulario con campos para introducir los datos del nuevo cliente, como nombre, dirección, número de teléfono, etc.</p> <p>FN3 El usuario completa el formulario con los datos del nuevo cliente.</p> <p>FN4 El usuario envía el formulario para agregar el nuevo cliente al sistema.</p> <p>FN5 El sistema verifica la validez de los datos ingresados y registra al nuevo cliente en la base de datos.</p> <p>FN6 El sistema muestra un mensaje de confirmación indicando que el nuevo cliente ha sido agregado con éxito.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide cancelar la acción de agregar nuevo cliente en cualquier momento, el sistema descarta los datos introducidos y no realiza ningún registro.</p>
Excepciones	<p>E1 Error en el sistema al procesar la solicitud de registro del nuevo cliente.</p> <p>E2 Campos obligatorios del formulario sin completar.</p>
Prioridad	Alta

Cuadro 5.14: CU-10. Agregar nuevo cliente

En el cuadro 5.15 se muestra el caso de uso CU-11. Buscar cliente:

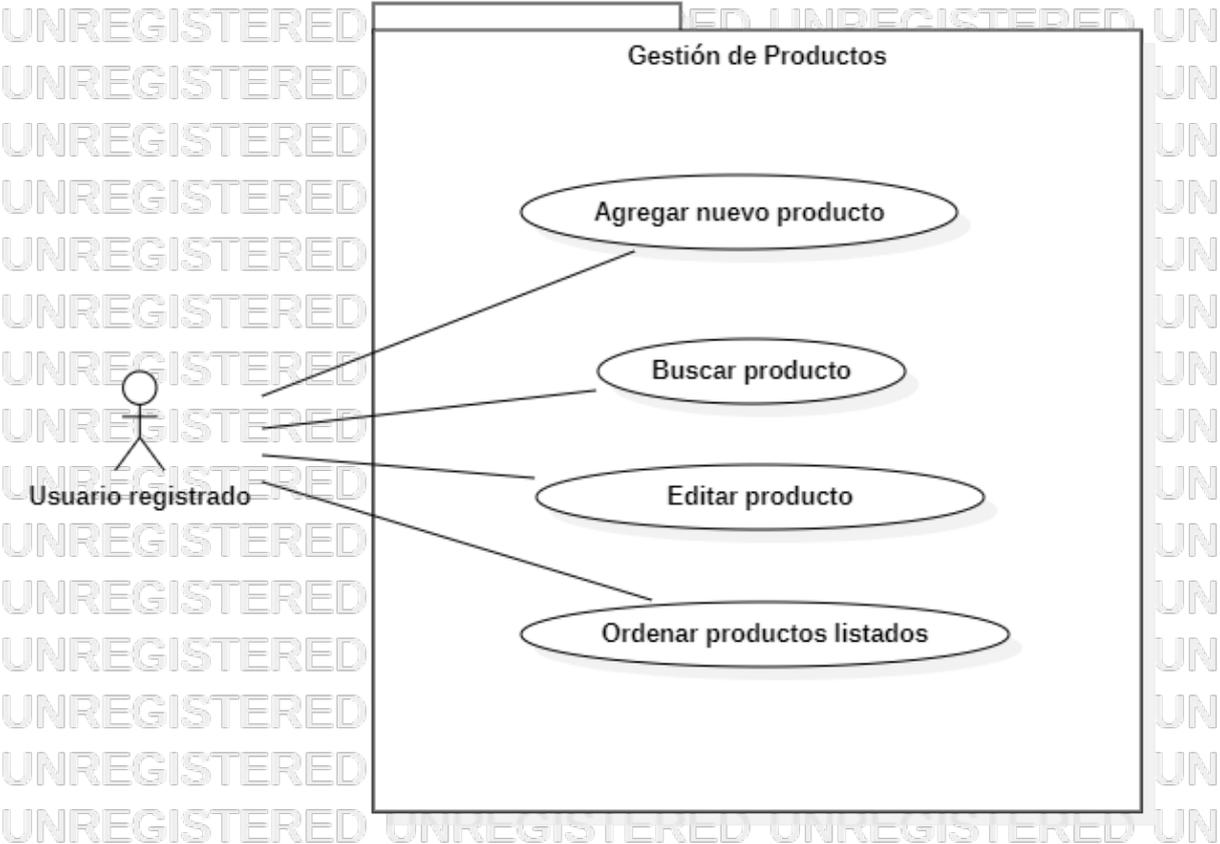


Figura 5.2: Casos de uso Subsistema gestionar productos.

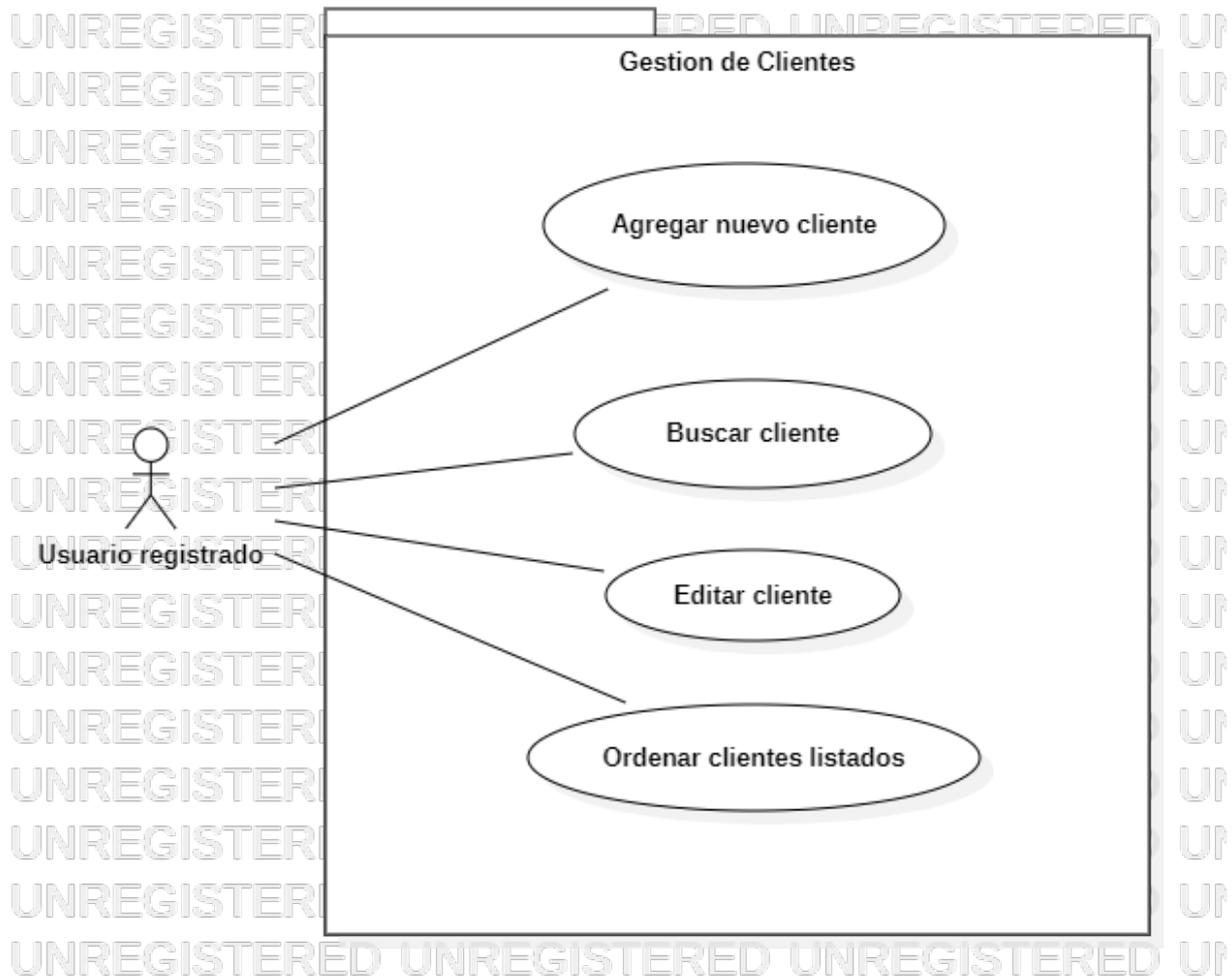


Figura 5.3: Casos de uso Subsistema gestionar clientes.

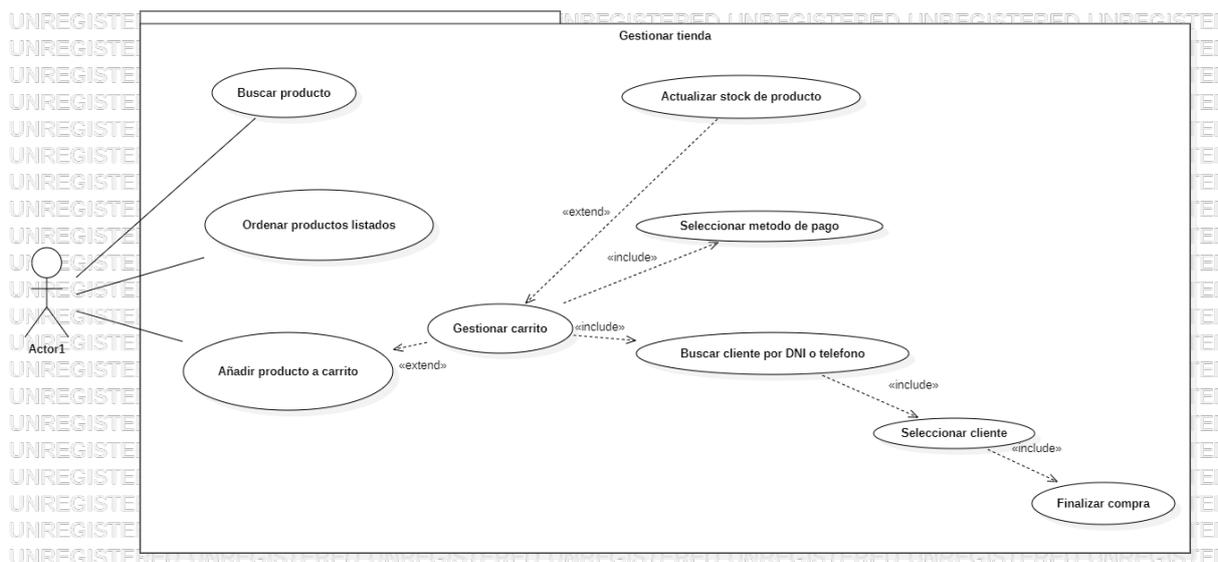


Figura 5.4: Casos de uso Subsistema gestionar tienda.

Nombre e ID del CU	CU-11. Buscar cliente
Actor	Usuario
Descripción	El usuario busca un cliente en el sistema ingresando un criterio de búsqueda, como el nombre, el número de teléfono o el correo electrónico del cliente.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene los permisos necesarios para buscar clientes.
Postcondiciones	POST-1. Se muestran los resultados de la búsqueda que coinciden con el criterio ingresado por el usuario.
Flujo normal	<p>FN1 El usuario accede a la opción de búsqueda de clientes en la interfaz de usuario.</p> <p>FN2 El sistema muestra un campo de búsqueda donde el usuario puede ingresar el criterio de búsqueda, como el nombre, el número de teléfono o el correo electrónico del cliente.</p> <p>FN3 El usuario ingresa el criterio de búsqueda en el campo correspondiente.</p> <p>FN4 El usuario envía la solicitud de búsqueda al sistema.</p> <p>FN5 El sistema realiza la búsqueda en la base de datos y muestra los resultados coincidentes con el criterio de búsqueda ingresado.</p>

Flujo alternativo 1	FA1 Si no se encuentran resultados que coincidan con el criterio de búsqueda ingresado, el sistema muestra un mensaje indicando que no se encontraron resultados.
Excepciones	E1 Error en el sistema al procesar la solicitud de búsqueda de clientes. E2 Campo de búsqueda vacío.
Prioridad	Alta

Cuadro 5.15: CU-11. Buscar cliente

En el cuadro 5.16 se muestra el caso de uso CU-12. Editar cliente:

Nombre e ID del CU	CU-12. Editar cliente
Actor	Usuario
Descripción	El usuario normal edita la información de un cliente existente en el sistema, como su nombre, dirección, número de teléfono, correo electrónico, etc.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene los permisos necesarios para editar clientes.
Postcondiciones	POST-1. Los cambios realizados en la información del cliente quedan guardados en el sistema.
Flujo normal	<p>FN1 El usuario accede a la lista de clientes existentes en el sistema.</p> <p>FN2 El usuario selecciona el cliente que desea editar de la lista.</p> <p>FN3 El sistema muestra la información actual del cliente seleccionado y proporciona campos editables para cada atributo, como nombre, dirección, número de teléfono, correo electrónico, etc.</p> <p>FN4 El usuario modifica los campos de información del cliente según sea necesario.</p> <p>FN5 El usuario confirma los cambios realizados y envía la solicitud de edición al sistema.</p> <p>FN6 El sistema actualiza la información del cliente con los cambios realizados y muestra un mensaje de confirmación al usuario.</p>

Flujo alternativo 1	FA1 Si el usuario decide cancelar la edición, puede salir de la pantalla de edición sin guardar los cambios.
Excepciones	E1 Error en el sistema al procesar la solicitud de edición de cliente. E2 Campos de información del cliente vacíos o con formato incorrecto.
Prioridad	Alta

Cuadro 5.16: CU-12. Editar cliente

En el cuadro 5.17 se muestra el caso de uso CU-13. Ordenar clientes listados:

Nombre e ID del CU	CU-13. Ordenar clientes listados
Actor	Usuario
Descripción	El usuario ordena la lista de clientes existentes en el sistema según algún criterio específico, como nombre, fecha de registro, número de teléfono, etc.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene acceso a la lista de clientes.
Postcondiciones	POST-1. La lista de clientes se muestra ordenada según el criterio seleccionado por el usuario.
Flujo normal	FN1 El usuario accede a la lista de clientes existentes en el sistema. FN2 El usuario selecciona el criterio de ordenamiento deseado, como nombre, fecha de registro, número de teléfono, etc. FN3 El sistema ordena la lista de clientes según el criterio seleccionado y la muestra al usuario.
Flujo alternativo 1	FA1 Si el usuario decide cancelar la ordenación o no selecciona ningún criterio, se muestra la lista de clientes en su estado original sin ordenar.
Excepciones	E1 Error en el sistema al procesar la solicitud de ordenación de clientes.

Prioridad	Media
------------------	-------

Cuadro 5.17: CU-13. Ordenar clientes listados

En el cuadro 5.18 se muestra el caso de uso CU-14. Añadir producto a carrito:

Nombre e ID del CU	CU-14. Añadir producto a carrito
Actor	Usuario
Descripción	El usuario agrega un producto específico al carrito de compras para proceder con la compra posteriormente.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y ha accedido a la lista de productos disponibles.
Postcondiciones	POST-1. El producto seleccionado se agrega al carrito de compras del usuario.
Flujo normal	<p>FN1 El usuario navega por la lista de productos disponibles.</p> <p>FN2 El usuario selecciona el producto que desea agregar al carrito de compras.</p> <p>FN3 El sistema agrega el producto seleccionado al carrito de compras del usuario.</p>
Excepciones	E1 Error en el sistema al procesar la solicitud de agregar producto al carrito.
Prioridad	Alta

Cuadro 5.18: CU-14. Añadir producto a carrito

En el cuadro 5.19 se muestra el caso de uso CU-15. Gestionar carrito:

Nombre e ID del CU	CU-15. Gestionar carrito
Actor	Usuario
Descripción	El usuario puede gestionar los productos agregados a su carrito de compras, permitiéndole eliminar productos o modificar las cantidades.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene al menos un producto en su carrito de compras.
Postcondiciones	POST-1. El usuario ha realizado las modificaciones deseadas en su carrito de compras.

Flujo normal	<p>FN1 El usuario accede a su carrito de compras.</p> <p>FN2 El sistema muestra la lista de productos actualmente en el carrito de compras del usuario, junto con las cantidades y los precios.</p> <p>FN3 El usuario tiene la opción de eliminar productos del carrito o modificar las cantidades de los productos.</p> <p>FN4 El sistema actualiza el carrito de compras del usuario según las modificaciones realizadas.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide cancelar la acción de gestionar el carrito, no se realizan modificaciones en el carrito de compras.</p>
Excepciones	<p>E1 Error en el sistema al procesar la solicitud de gestionar el carrito de compras.</p>
Prioridad	Alta

Cuadro 5.19: CU-15. Gestionar carrito

En el cuadro 5.20 se muestra el caso de uso CU-16. Actualizar stock de un producto (dentro del carrito):

Nombre e ID del CU	CU-16. Actualizar stock de un producto (dentro del carrito)
Actor	Usuario
Descripción	El usuario puede actualizar el stock de un producto dentro de su carrito de compras, permitiéndole verificar si hay suficiente stock disponible antes de finalizar la compra.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y tiene al menos un producto en su carrito de compras.
Postcondiciones	POST-1. El usuario ha verificado y actualizado el stock de los productos en su carrito de compras según la disponibilidad en el sistema.

Flujo normal	<p>FN1 El usuario accede a su carrito de compras.</p> <p>FN2 El sistema muestra la lista de productos actualmente en el carrito de compras del usuario, junto con las cantidades, los precios y el stock disponible.</p> <p>FN3 El usuario tiene la opción de verificar y actualizar el stock de los productos en su carrito antes de finalizar la compra.</p> <p>FN4 El sistema actualiza el stock de los productos en el carrito de compras del usuario normal según las verificaciones y actualizaciones realizadas.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide no actualizar el stock de ningún producto en su carrito, se procede con el proceso de compra sin cambios en el stock.</p>
Excepciones	<p>E1 Error en el sistema al procesar la solicitud de actualizar el stock de un producto en el carrito de compras.</p>
Prioridad	Alta

Cuadro 5.20: CU-16. Actualizar stock de un producto (dentro del carrito)

En el cuadro 5.21 se muestra el caso de uso CU-17. Seleccionar método de pago:

Nombre e ID del CU	CU-17. Seleccionar método de pago
Actor	Usuario
Descripción	El usuario selecciona el método de pago que desea utilizar para realizar la compra de los productos en su carrito.
Precondiciones	PRE-1. El usuario ha iniciado sesión en el sistema y ha agregado al menos un producto al carrito de compras.
Postcondiciones	POST-1. El usuario ha seleccionado un método de pago válido para finalizar la compra.

Flujo normal	<p>FN1 El usuario accede a su carrito de compras.</p> <p>FN2 El sistema muestra la lista de productos en el carrito junto con el monto total a pagar.</p> <p>FN3 El usuario selecciona el método de pago deseado entre las opciones disponibles (por ejemplo, tarjeta de crédito/débito y efectivo).</p> <p>FN4 El sistema verifica la validez del método de pago seleccionado por el usuario.</p> <p>FN5 Si el método de pago es válido, el sistema procede a finalizar la compra.</p>
Flujo alternativo 1	<p>FA1 Si el usuario decide cancelar la compra antes de seleccionar un método de pago, se cancela el proceso y se redirige al usuario de vuelta al carrito de compras.</p>
Excepciones	<p>E1 El usuario intenta seleccionar un método de pago inválido o no disponible.</p> <p>E2 Error en el sistema al procesar la selección del método de pago.</p>
Prioridad	Alta

Cuadro 5.21: CU-17. Seleccionar método de pago

5.1.5. Requisitos funcionales

En el cuadro 5.22 se muestran los diferentes requisitos funcionales de la app:

Requisitos de eficiencia

En el cuadro 5.23 se muestran los diferentes requisitos de eficiencia de la app:

Requisitos de usabilidad

En el cuadro 5.24 se muestran los diferentes requisitos de usabilidad de la app:

Requisitos de mantenibilidad

En el cuadro 5.25 se muestran los diferentes requisitos de mantenibilidad de la app:

ID	Requisito
RF-01	La aplicación mostrará los diferentes campos para poder solicitar información (iniciar sesión, registrarse, editar un producto, etc)
RF-02	La aplicación será capaz de validar los todos los datos introducidos a la hora de rellenar un campo de requisición de información
RF-03	La aplicación será capaz de dar acceso a un usuario o rechazarlo una vez haya introducido los datos de acceso
RF-04	La aplicación será capaz de actualizar la base de datos cada vez que se haga una modificación o venta
RF-05	La aplicación será capaz de generar una factura en pdf con todos los datos de la compra una vez se haya finalizado dicha compra

Cuadro 5.22: Requisitos funcionales

ID	Requisito
RN-01	La aplicación debe tener un tasa de fallo de 1 % como mucho
RN-02	La aplicación debe tener deslimitado el número de accesos de un usuario al día
RN-03	La aplicación debe soportar el uso simultáneo de varios usuarios

Cuadro 5.23: Requisitos de eficiencia

ID	Requisito
RN-01	La aplicación deberá ser compatible con todos los navegadores web
RN-02	La aplicación deberá ser legible y comprensible
RN-03	La aplicación deberá ser de uso fácil

Cuadro 5.24: Requisitos de usabilidad

ID	Requisito
RN-01	La aplicación deberá ser capaz de reiniciarse con facilidad una vez se corrija un error
RN-02	La aplicación deberá estar preparada para poder soportar cambios y mejoras que pida el cliente

Cuadro 5.25: Requisitos de mantenibilidad

Requisitos de seguridad

En el cuadro 5.26 se muestran los diferentes requisitos de seguridad de la app:

ID	Requisito
RNF-SEG-01	Las contraseñas se guardarán cifradas en la base de datos
RNF-SEG-02	La aplicación no permitirá crear 2 cuentas con el mismo email

Cuadro 5.26: Requisitos de seguridad

Requisitos de disponibilidad

En el cuadro 5.27 se muestran los diferentes requisitos de disponibilidad de la app:

ID	Requisito
RNF-DSP-01	La aplicación estará disponible siempre que el usuario quiera disponer de ella, es decir, los 365 días del año

Cuadro 5.27: Requisitos de disponibilidad

Requisitos de implementacion

En el cuadro 5.28 se muestran los diferentes requisitos de implementación de la app:

ID	Requisito
RNF-IMP-01	La aplicación de desarrollará utilizando Visual Studio Code
RNF-IMP-02	La aplicación se desarrollará en el lenguaje de php para el back-end y html para el front-end
RNF-IMP-03	La base de datos de la aplicación se desarrollará en MySql con lenguaje sql

Cuadro 5.28: Requisitos de implementacion

5.1.6. Requisitos de informacion

En el cuadro 5.29 se muestran los diferentes requisitos de información de la app:

5.2. Atributos de calidad

Se analizarán aquí los posibles indicadores de calidad de la aplicación desarrollada, a saber:

Rendimiento: tiene que ver con los tiempos de espera y de ejecución, así como la gestión de la memoria y posibles problemas de concurrencia.

ID	Requisito
RFI-01	Para poder registrar un usuario es necesario introducir username, email y password
RFI-02	Para poder registrar un producto es necesario introducir nombre, descripcion, stock, precio y código de barras
RFI-03	Para poder modificar un producto es necesario introducir nombre, descripcion, stock, precio y/o código de barras
RFI-04	Para poder modificar un usuario es necesario introducir username y/o email
RFI-05	Para poder hacer login con un usuario es necesario introducir email y password
RFI-06	Para poder modificar una contraseña de un usuario es necesario introducir contraseña actual, contraseña nueva y confirmar la contraseña nueva
RFI-07	Para poder buscar un producto es necesario introducir algun texto coincidente(nombre, descripcion, codigo de barras)
RFI-08	Para poder buscar un cliente es necesario introducir algun texto coincidente(nombre, email, dni, teléfono)
RFI-09	Para poder buscar un cliente a la hora de finalizar una compra, es necesario introducir dni o telefono

Cuadro 5.29: Requisitos de informacion

Seguridad: está relacionado con la seguridad del sistema, la privacidad de datos, etc.

Robustez: tiene que ver con el comportamiento, la fiabilidad y la estabilidad de la aplicación ante posibles incidencias en tiempo real.

Capítulo 6

Diseño

Los aspectos de diseño depende mucho del paradigma elegido (programación orientada a objetos o no, por ejemplo).

6.1. Diseño de datos

6.1.1. Modelo Entidad-Relación(ER)

En primer lugar, vamos a hablar del modelo entidad-relación. El modelo entidad-relación te permite visualizar las entidades de la aplicación, sus atributos y las relaciones entre ellas. A continuación se muestra dicho diagrama 6.1.

En este diagrama, las entidades se representan con cuadrados, las relaciones con rombos y los atributos con óvalos.

6.1.2. Modelo relacional

A partir del modelo entidad-relacion se obtiene el modelo relacional, el cual está formado por las siguientes tablas:

Entidad clientes

En el cuadro 6.1 se muestra los datos de la entidad cliente de la app:

Entidad facturas

En el cuadro 6.2 se muestra los datos de la entidad factura de la app:

Entidad productos

En el cuadro 6.3 se muestra los datos de la entidad producto de la app:

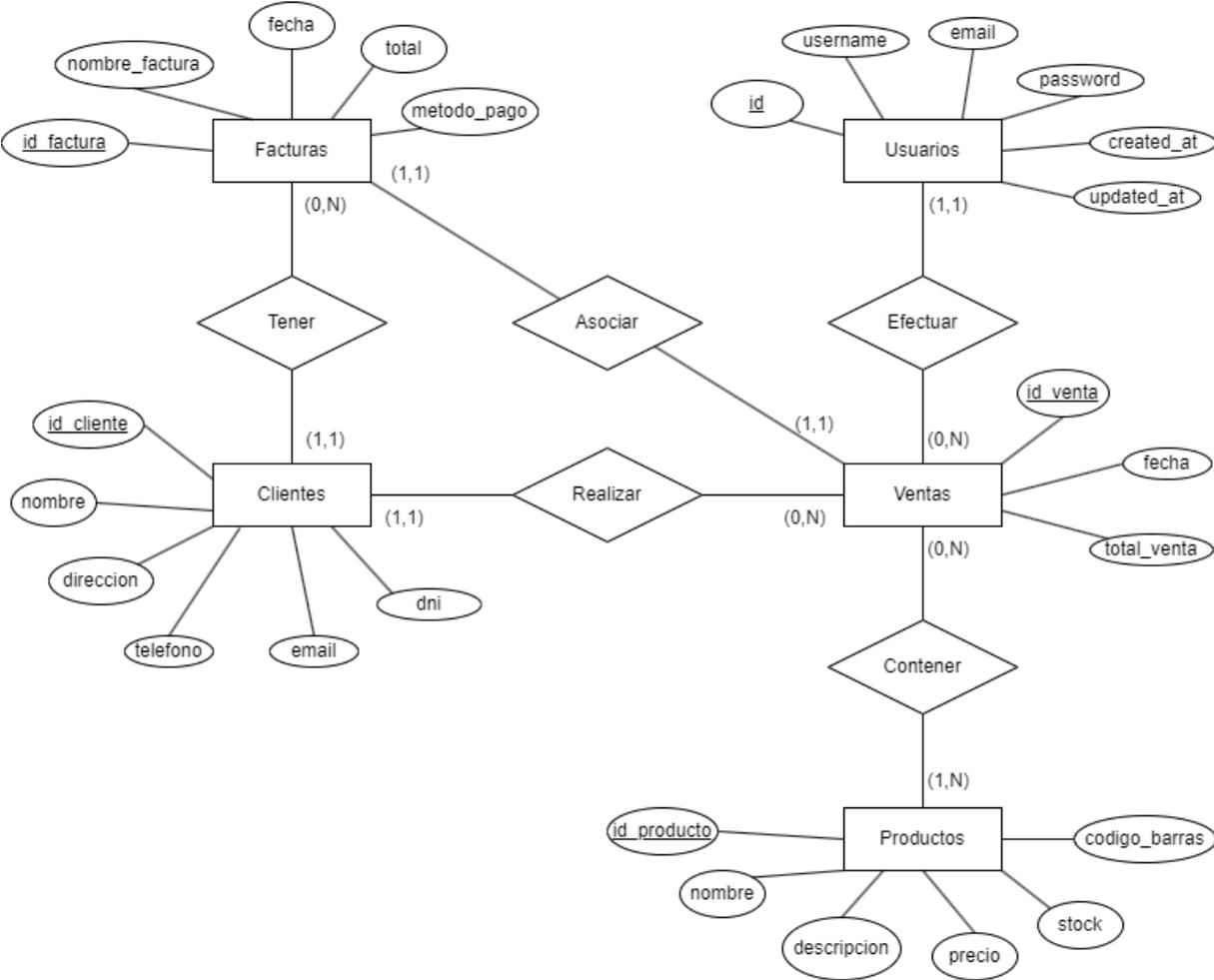


Figura 6.1: Diagrama entidad relación.

ID	Nombre	Definición	Tipo	Restricciones
01.01	id_cliente	Identificador del cliente	int	Primary Key
01.02	nombre	Nombre del cliente	varchar	Not null
01.03	direccion	Dirección del cliente	varchar	Not null
01.04	telefono	telefono de contacto del cliente	varchar	Not null
01.05	email	Email del cliente	varchar	Not null
01.06	dni	Dni del cliente	varchar	Not null

Cuadro 6.1: Tabla entidad clientes

ID	Nombre	Definición	Tipo	Restricciones
02.01	id_cliente	Identificador del cliente	int	Foreign Key
02.02	id_factura	Identificador de la factura	int	Primary key
02.03	fecha	Fecha de la compra	date	Not null
02.04	total	Precio total de la compra	decimal	Not null
02.05	nombre_fac- tura	Nombre de la factura	varchar	Not null
02.06	metodo_pago	Meétodo de pago de la compra	varchar	Not null

Cuadro 6.2: Tabla entidad facturas

ID	Nombre	Definición	Tipo	Restricciones
03.01	id_producto	Identificador del producto	int	Primary Key
03.02	nombre	Nombre del producto	varchar	Not null
03.03	descripcion	Descripción del producto	text	Not null
03.04	precio	Precio del producto	decimal	Not null
03.05	stock	Cantidad de unidades del pro- ducto disponibles para la venta	int	Not null
03.06	codigo_barras	Codigo de barras del producto	varchar	Not null

Cuadro 6.3: Tabla entidad productos

Entidad usuarios

En el cuadro 6.4 se muestra los datos de la entidad usuario de la app:

ID	Nombre	Definición	Tipo	Restricciones
04.01	id	Identificador del usuario	int	Primary Key
04.02	username	Nombre del usuario	varchar	Not null
04.03	email	Email del usuario	varchar	Not null
04.04	password	Contraseña del usuario	varchar	Not null
04.05	created_at	Fecha de creación del usuario	timestamp	Not null
04.06	updated_at	Fecha de la última actualización	timestamp	Not null

Cuadro 6.4: Tabla entidad usuarios

Entidad ventas

En el cuadro 6.5 se muestra los datos de la entidad venta de la app:

ID	Nombre	Definición	Tipo	Restricciones
05.01	id_venta	Identificador del usuario	int	Primary Key
05.012	id-cliente	Nombre del usuario	int	Foreign Key
05.013	fecha	Email del usuario	date	Not null
05.014	total_venta	Contraseña del usuario	decimal	Not null
05.015	id_usuario	Fecha de creación del usuario	int	Not null

Cuadro 6.5: Tabla entidad ventas

Relación tener

En el cuadro 6.6 se muestra los datos de la relación tener de la app:

ID Entidad	Nombre Entidad	Participación	Cardinalidad
E01	Clientes	0	N
E02	Facturas	1	1

Cuadro 6.6: Tabla relación tener

Relación asociar

En el cuadro 6.7 se muestra los datos de la relación asociar de la app:

ID Entidad	Nombre Entidad	Participación	Cardinalidad
E05	Ventas	1	1
E02	Facturas	1	1

Cuadro 6.7: Tabla relación asociar

Relación realizar

En el cuadro 6.8 se muestra los datos de la relación realizar de la app:

ID Entidad	Nombre Entidad	Participación	Cardinalidad
E01	Clientes	0	N
E05	Ventas	1	1

Cuadro 6.8: Tabla relación realizar

Relación efectuar

En el cuadro 6.9 se muestra los datos de la relación efectuar de la app:

ID Entidad	Nombre Entidad	Participación	Cardinalidad
E04	Usuarios	0	N
E05	Ventas	1	1

Cuadro 6.9: Tabla relación efectuar

Relación contener

En el cuadro 6.10 se muestra los datos de la relación contener de la app:

ID Entidad	Nombre Entidad	Participación	Cardinalidad
E03	Productos	0	N
E05	Ventas	1	N

Cuadro 6.10: Tabla relación contener

6.2. Diagramas de clase y de secuencia

Si la programación es orientada a objetos, es necesario especificar los diagramas de clases que se usan.

Los diagramas de secuencia describen la secuencia de pasos con que un usuario interactúa con el sistema, y están relacionados con los casos de uso de la etapa de análisis.

Es interesante también especificar si se ha hecho uso de diseño de patrones.

Capítulo 7

Implementación y arquitectura

En este capítulo se mostrarán los detalles más relevantes en cuanto a la implementación del sistema.

7.1. Estructura del Proyecto

A continuación, se presenta la estructura de directorios y archivos del proyecto, organizada siguiendo el patrón Modelo-Vista-Controlador (MVC):

```
Almacen/  
  .vscode/  
    launch.json  
assets/  
  css/  
    style.css  
config/  
controllers/  
  clienteController.php  
  compraController.php  
  homeController.php  
  productController.php  
  userController.php  
fpdf/  
models/  
  cliente.php  
  factura.php  
  product.php  
  user.php  
public/  
  utils.php  
views/
```

```
client/  
  create.php  
  edit.php  
  index.php  
compras/  
  carrito.php  
  confirmacion.php  
  tienda.php  
home/  
  index.php  
partials/  
  header.php  
product/  
  add.php  
  edit.php  
  list.php  
user/  
  account.php  
  change_password.php  
  login.php  
  logout.php  
  signup.php  
.htaccess  
carrito.jpg  
database.php  
index.php
```

7.2. Descripción de los Componentes

7.2.1. Controladores

Los controladores son los responsables de manejar las solicitudes del usuario, procesar los datos a través de los modelos y devolver las respuestas adecuadas utilizando las vistas.

- **clienteController.php**: Maneja las operaciones relacionadas con los clientes.
- **compraController.php**: Gestiona las compras y el flujo del carrito de compras.
- **homeController.php**: Controla la vista principal del sistema.
- **productController.php**: Gestiona las operaciones relacionadas con los productos.
- **UserController.php**: Maneja las operaciones de usuario, como inicio de sesión y registro.

7.2.2. Modelos

Los modelos representan la estructura de los datos y contienen las funciones para interactuar con la base de datos.

- **cliente.php**: Modelo para la entidad Cliente.
- **factura.php**: Modelo para la entidad Factura.
- **product.php**: Modelo para la entidad Producto.
- **user.php**: Modelo para la entidad Usuario.

7.2.3. Vistas

Las vistas son responsables de la presentación de los datos al usuario. Están organizadas según las entidades y funcionalidades principales del sistema.

- **client/**: Contiene las vistas para la gestión de clientes (crear, editar, listar).
- **compras/**: Incluye las vistas relacionadas con el proceso de compra (carrito, confirmación, tienda).
- **home/**: Contiene la vista principal del sistema.
- **partials/**: Incluye fragmentos de vistas reutilizables, como el header.
- **product/**: Contiene las vistas para la gestión de productos (añadir, editar, listar).
- **user/**: Incluye las vistas para las operaciones de usuario (cuenta, cambio de contraseña, inicio de sesión, registro).

7.2.4. Otras Carpetas y Archivos

- **assets/**: Contiene archivos estáticos como hojas de estilo CSS.
- **config/**: Incluye archivos de configuración necesarios para el sistema.
- **fpdf/**: Contiene la biblioteca FPDF para la generación de facturas en formato PDF.
- **public/**: Incluye utilidades públicas, como scripts compartidos.
- **database.php**: Archivo de conexión a la base de datos.
- **index.php**: Punto de entrada principal del sistema.
- **.htaccess**: Archivo de configuración del servidor.

7.3. Detalles de Implementación

7.3.1. Controlador de Usuarios

El archivo `UserController.php` maneja las operaciones relacionadas con los usuarios, como el inicio de sesión, el registro y la actualización de datos del usuario. Aquí se ejemplifica el método `login`:

```
<?php
require_once 'models/user.php';

class UserController {
    public function index() {
        session_start();

        if (!isset($_SESSION['user_id'])) {
            header('Location: index.php?controller=UserController&action=login');
            exit();
        }

        $userModel = new user();
        $user_info = $userModel->getUserById($_SESSION['user_id']);

        require 'views/user/account.php';
    }

    public function login() {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            if (!isset($_POST['email']) || !isset($_POST['password'])) {
                $error_message = 'Por favor, complete todos los campos.';
                require 'views/user/login.php';
                return;
            }

            $email = $_POST['email'];
            $password = $_POST['password'];

            $userModel = new user();
            $user = $userModel->getUserByEmail($email);

            if ($user && password_verify($password, $user['password'])) {
                session_start();
                $_SESSION['user_id'] = $user['id'];
                header('Location: index.php');
```

```
        } else {
            $error_message = 'Email o contraseña incorrectos.';
            require 'views/user/login.php';
        }
    } else {
        require 'views/user/login.php';
    }
}

public function logout() {
    session_start();
    session_destroy();
    header('Location: index.php?controller=userController&action=login');
}

public function signup() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $username = $_POST['username'];
        $email = $_POST['email'];
        $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

        $userModel = new user();
        $userModel->createUser($username, $email, $password);

        header('Location: index.php?controller=userController&action=login');
    } else {
        require 'views/user/signup.php';
    }
}

public function account() {
    session_start();

    if (!isset($_SESSION['user_id'])) {
        header('Location: index.php?controller=userController&action=login');
        exit();
    }

    $userModel = new user();
    $user_info = $userModel->getUserById($_SESSION['user_id']);

    require 'views/user/account.php';
}
```

```

public function change_password() {
    session_start();

    if (!isset($_SESSION['user_id'])) {
        header('Location: index.php?controller=userController&action=login');
        exit();
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $current_password = $_POST['current_password'];
        $new_password = $_POST['new_password'];
        $confirm_password = $_POST['confirm_password'];

        if ($new_password !== $confirm_password) {
            $error_message = 'La nueva contraseña y la confirmación no coinciden.';
            require 'views/user/change_password.php';
            return;
        }

        $userModel = new user();
        $user = $userModel->getUserById($_SESSION['user_id']);

        if (!$user || !password_verify($current_password, $user['password'])) {
            $error_message = 'La contraseña actual es incorrecta.';
            require 'views/user/change_password.php';
            return;
        }

        $hashed_new_password = password_hash($new_password, PASSWORD_DEFAULT);
        $result = $userModel->updatePassword($_SESSION['user_id'], $hashed_new_password);

        if ($result) {
            $success_message = 'Contraseña actualizada correctamente.';
        } else {
            $error_message = 'Error al actualizar la contraseña.';
        }
    }

    require 'views/user/change_password.php';
}
}
?>

```

Este método verifica las credenciales del usuario y lo redirige a la página principal si la autenticación es exitosa.

7.3.2. Modelo de Usuario

El archivo `user.php` contiene el modelo de usuario, que incluye métodos para interactuar con la base de datos y gestionar la información del usuario.

```
<?php
require_once 'database.php';

class user {
    public function getUserById($id) {
        $conn = obtenerConexion();
        $query = $conn->prepare('SELECT * FROM users WHERE id = ?');
        $query->bind_param('i', $id);
        $query->execute();
        $result = $query->get_result();
        cerrarConexion($conn);
        return $result->fetch_assoc();
    }

    public function getUserByEmail($email) {
        $conn = obtenerConexion();
        $query = $conn->prepare('SELECT * FROM users WHERE email = ?');
        $query->bind_param('s', $email);
        $query->execute();
        $result = $query->get_result();
        cerrarConexion($conn);
        return $result->fetch_assoc();
    }

    public function createUser($username, $email, $password) {
        $conn = obtenerConexion();
        $query = $conn->prepare('INSERT INTO users (username, email, password) VALUES
        $query->bind_param('sss', $username, $email, $password);
        $query->execute();
        cerrarConexion($conn);
    }

    public function updatePassword($user_id, $new_password) {
        $conn = obtenerConexion();
        $query = $conn->prepare('UPDATE users SET password = ? WHERE id = ?');
```

```

        $query->bind_param('si', $new_password, $user_id);
        $result = $query->execute();
        cerrarConexion($conn);
        return $result;
    }
}
?>

```

7.4. Vistas

Las vistas presentan la interfaz del usuario. Por ejemplo, la vista de inicio de sesión `login.php`:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
    <header>
        <h1>Ferretería</h1>
    </header>
    <div class="main-container">
        <h1>Login</h1>
        <?php if (!empty($error_message)): ?>
            <p style="color: red;"><?php echo $error_message; ?></p>
        <?php endif; ?>
        <form action="index.php?controller=userController&action=login" method="POST">
            <input type="email" name="email" placeholder="Enter your email" required>
            <input type="password" name="password" placeholder="Enter your Password"
            <input type="submit" value="Submit">
        </form>
        <p>or <a href="index.php?controller=userController&action=signup">SignUp</a></p>
    </div>
</body>
</html>

```

Esta vista muestra el formulario de inicio de sesión y gestiona los mensajes de error.

7.5. Conclusión

La implementación del sistema sigue el patrón Modelo-Vista-Controlador (MVC) para separar responsabilidades, lo que facilita el mantenimiento del código y la escalabilidad. Cada componente está claramente definido, lo que permite una gestión eficaz de la lógica empresarial, la interacción con la base de datos y la presentación de la interfaz de usuario.

Capítulo 8

Pruebas

Este capítulo es crucial para demostrar el correcto funcionamiento del sistema. En él se detallarán las estrategias de prueba utilizadas y los momentos en que se realizaron dichas pruebas. Se incluirán pruebas de caja negra, caja blanca y de seguridad. Además, se presentará una completa batería de pruebas para validar la funcionalidad de la aplicación, de acuerdo con los requisitos establecidos.

8.1. Estrategias de Prueba

Las pruebas realizadas se han dividido en varias categorías:

- **Pruebas de Caja Negra:** Estas pruebas se centran en las entradas y salidas del sistema, sin considerar su funcionamiento interno.
- **Pruebas de Caja Blanca:** Se examina el funcionamiento interno del sistema, verificando rutas específicas y condiciones lógicas.
- **Pruebas de Seguridad:** Se aseguran de que el sistema es seguro frente a posibles ataques y vulnerabilidades.

8.2. Pruebas de Caja Negra

Las pruebas de caja negra se han realizado para validar la funcionalidad del sistema desde el punto de vista del usuario final. A continuación, se presentan algunos ejemplos de estas pruebas.

En el cuadro 8.1 se muestra la prueba de caja negra PCN-01: Registrar Usuario:

PCN-01: Registrar Usuario	
Objetivo de la prueba	Valorar la posibilidad de crear nuevos usuarios desde la aplicación.
Precondiciones	La inexistencia de una cuenta de usuario en el sistema con las mismas credenciales.
Datos de entrada	<ul style="list-style-type: none"> ▪ Email: admin@gmail.com ▪ Nombre y apellidos de usuario: Juan Pérez ▪ Contraseña: admin
Respuesta esperada	La creación de cuenta de usuario en el sistema se ha realizado con éxito.
Resultado final	VÁLIDO.

Cuadro 8.1: Prueba de caja negra - Registrar usuario

En el cuadro 8.2 se muestra la prueba de caja negra PCN-02: Iniciar Sesión:

PCN-02: Iniciar Sesión	
Objetivo de la prueba	Verificar que los usuarios pueden iniciar sesión con credenciales correctas.
Precondiciones	Existencia de una cuenta de usuario registrada con las credenciales proporcionadas.
Datos de entrada	<ul style="list-style-type: none">▪ Email: admin@gmail.com▪ Contraseña: user123
Respuesta esperada	El usuario inicia sesión correctamente y se redirige a la página principal.
Resultado final	VÁLIDO.

Cuadro 8.2: Prueba de caja negra - Iniciar Sesión

En el cuadro 8.3 se muestra la prueba de caja negra PCN-03: Cerrar Sesión:

PCN-03: Cerrar Sesión	
Objetivo de la prueba	Probar la funcionalidad de cerrar sesión.
Precondiciones	El usuario debe estar autenticado en el sistema.
Datos de entrada	<ul style="list-style-type: none"> ▪ Ninguno
Respuesta esperada	El usuario cierra la sesión y es redirigido a la página de inicio de sesión.
Resultado final	VÁLIDO.

Cuadro 8.3: Prueba de caja negra - Cerrar Sesión

En el cuadro 8.4 se muestra la prueba de caja negra PCN-04: Fallo al Iniciar Sesión:

PCN-04: Fallo al Iniciar Sesión	
Objetivo de la prueba	Verificar que los usuarios no pueden iniciar sesión con credenciales incorrectas.
Precondiciones	La no existencia de una cuenta de usuario registrada con las credenciales proporcionadas.
Datos de entrada	<ul style="list-style-type: none"> ▪ Email: xxxxxxxx ▪ Contraseña: xxxxxxxx
Respuesta esperada	El usuario no puede iniciar sesión correctamente y le da la posibilidad de volver a iniciar sesión probando otro usuario y contraseña.
Resultado final	VÁLIDO.

Cuadro 8.4: Prueba de caja negra - Fallo al iniciar Sesión

En el cuadro 8.5 se muestra la prueba de caja negra PCN-05: Buscar Producto:

PCN-05: Buscar Producto	
Objetivo de la prueba	Validar la búsqueda de productos en la sección productos.
Precondiciones	El sistema debe tener productos disponibles en la base de datos.
Datos de entrada	<ul style="list-style-type: none"> ▪ Término de búsqueda: "Tornillo métrica 5"
Respuesta esperada	El sistema muestra una lista de productos que coinciden con el término de búsqueda.
Resultado final	VÁLIDO.

Cuadro 8.5: Prueba de caja negra - Buscar Producto

En el cuadro 8.6 se muestra la prueba de caja negra PCN-06: Añadir Producto al Carrito:

PCN-06: Añadir Producto al Carrito	
Objetivo de la prueba	Probar la adición de productos al carrito de compras.
Precondiciones	El usuario debe haber iniciado sesión y haber seleccionado un producto.
Datos de entrada	<ul style="list-style-type: none"> ▪ Producto: "Tornillo métrica 5" ▪ Cantidad: 10
Respuesta esperada	El sistema añade el producto al carrito de compras del usuario.
Resultado final	VÁLIDO.

Cuadro 8.6: Prueba de caja negra - Añadir Producto al Carrito

En el cuadro 8.7 se muestra la prueba de caja negra PCN-07: Finalizar Compra:

PCN-07: Finalizar Compra	
Objetivo de la prueba	Verificar el proceso de finalización de compra.
Precondiciones	El usuario debe tener productos en el carrito de compras y haber iniciado sesión.
Datos de entrada	<ul style="list-style-type: none"> ▪ Seleccionar cliente: Cliente a hacer la compra ▪ Método de pago: Tarjeta de crédito ▪ Información de envío: Dirección del usuario
Respuesta esperada	El sistema procesa el pago y genera una confirmación de la compra.
Resultado final	VÁLIDO.

Cuadro 8.7: Prueba de caja negra - Finalizar Compra

En el cuadro 8.8 se muestra la prueba de caja negra PCN-08: Actualizar Datos de Usuario:

PCN-08: Actualizar Datos de Usuario	
Objetivo de la prueba	Validar la actualización de los datos de perfil de usuario.
Precondiciones	El usuario debe estar autenticado y tener acceso a su perfil.
Datos de entrada	<ul style="list-style-type: none"> ▪ Nombre: Juan Pérez ▪ Dirección: Calle Falsa 123 ▪ Teléfono: 123456789
Respuesta esperada	Los datos del perfil del usuario se actualizan correctamente en el sistema.
Resultado final	VÁLIDO.

Cuadro 8.8: Prueba de caja negra - Actualizar Datos de Usuario

En el cuadro 8.9 se muestra la prueba de caja negra PCN-09: Eliminar Producto del Carrito:

PCN-09: Eliminar Producto del Carrito	
Objetivo de la prueba	Verificar que los usuarios pueden eliminar productos del carrito de compras.
Precondiciones	El usuario debe haber iniciado sesión y tener productos en el carrito.
Datos de entrada	<ul style="list-style-type: none"> ▪ Producto: "Laptop"
Respuesta esperada	El producto es eliminado del carrito de compras del usuario.
Resultado final	VÁLIDO.

Cuadro 8.9: Prueba de caja negra - Eliminar Producto del Carrito

En el cuadro 8.10 se muestra la prueba de caja negra PCN-10: Cambiar Cantidad de Producto en el Carrito:

PCN-10: Cambiar Cantidad de Producto en el Carrito	
Objetivo de la prueba	Validar la funcionalidad para cambiar la cantidad de un producto en el carrito de compras.
Precondiciones	El usuario debe haber iniciado sesión y tener productos en el carrito.
Datos de entrada	<ul style="list-style-type: none"> ▪ Producto: "Laptop" ▪ Cantidad: 2
Respuesta esperada	El sistema actualiza la cantidad del producto en el carrito y el total correspondiente.
Resultado final	VÁLIDO.

Cuadro 8.10: Prueba de caja negra - Cambiar Cantidad de Producto en el Carrito

En el cuadro 8.11 se muestra la prueba de caja negra PCN-11: Cambiar Contraseña:

PCN-11: Cambiar Contraseña	
Objetivo de la prueba	Verificar que los usuarios pueden cambiar la contraseña de su usuario correctamente.
Precondiciones	La existencia de una cuenta de usuario registrada con las credenciales proporcionadas.
Datos de entrada	<ul style="list-style-type: none"> ▪ Contraseña actual: Admin123. ▪ Nueva contraseña: Cafe123. ▪ Confirmación nueva contraseña: Cafe123.
Respuesta esperada	La contraseña queda cambiada correctamente y almacenada en base de datos cifrada.
Resultado final	VÁLIDO.

Cuadro 8.11: Prueba de caja negra - Cambiar contraseña

En el cuadro 8.13 se muestra la prueba de caja negra PCN-12: Fallo Contraseña Antigua Cambiar Contraseña:

PCN-12: Fallo Contraseña Antigua Cambiar Contraseña	
Objetivo de la prueba	Verificar que los usuarios no pueden cambiar la contraseña de su usuario correctamente si no se introduce bien la contraseña antigua .
Precondiciones	La existencia de una cuenta de usuario registrada con las credenciales proporcionadas.
Datos de entrada	<ul style="list-style-type: none"> ▪ Contraseña actual: xxxxxx. ▪ Nueva contraseña: Cafe123. ▪ Confirmación nueva contraseña: Cafe123.
Respuesta esperada	La contraseña no se cambia correctamente.
Resultado final	VÁLIDO.

Cuadro 8.12: Prueba de caja negra - Fallo Contraseña Nueva Cambiar contraseña

En el cuadro 8.13 se muestra la prueba de caja negra PCN-13: Fallo Contraseña Antigua Cambiar Contraseña:

PCN-13: Fallo Contraseña Antigua Cambiar Contraseña	
Objetivo de la prueba	Verificar que los usuarios no pueden cambiar la contraseña de su usuario correctamente si no se introduce la misma contraseña en nueva contraseña y en la confirmación.
Precondiciones	La existencia de una cuenta de usuario registrada con las credenciales proporcionadas.
Datos de entrada	<ul style="list-style-type: none"> ▪ Contraseña actual: Admin123. ▪ Nueva contraseña: Tutucaca12. ▪ Confirmación nueva contraseña: Cafe123.
Respuesta esperada	La contraseña no se cambia correctamente.
Resultado final	VÁLIDO.

Cuadro 8.13: Prueba de caja negra - Fallo Contraseña Nueva Cambiar contraseña

8.3. Pruebas de Caja Blanca

Las pruebas de caja blanca se han centrado en verificar el correcto funcionamiento interno del sistema. Se han diseñado casos de prueba para cubrir las rutas más críticas y las condiciones lógicas del código.

- **PCB-01:** Verificación del flujo de registro de usuarios, cubriendo todas las ramas de decisión en el código.
- **PCB-02:** Verificación del proceso de inicio de sesión, asegurando que todas las condiciones de seguridad y validación de credenciales sean correctas.
- **PCB-03:** Prueba de las funciones de validación de entrada para formularios.
- **PCB-04:** Verificación de la lógica de recuperación de contraseña.
- **PCB-05:** Prueba de los algoritmos de búsqueda de productos.
- **PCB-06:** Verificación del cálculo de precios en el carrito de compras.
- **PCB-07:** Prueba de la función de actualización de perfil de usuario.
- **PCB-08:** Verificación de la gestión de stock de productos.
- **PCB-09:** Prueba de las funciones de gestión de usuarios por el administrador.

8.4. Pruebas de Seguridad

Para asegurar que el sistema es robusto frente a posibles ataques y vulnerabilidades, se han realizado las siguientes pruebas de seguridad:

- **PS-01:** Prueba de inyección SQL en el formulario de inicio de sesión.
- **PS-02:** Prueba de fuerza bruta en el sistema de autenticación.

Parte III

Manuales de la Aplicación

Capítulo 9

Manual de Instalación

Este capítulo está dedicado a explicar los prerequisites técnicos necesarios, el sistema operativo y otros detalles para instalar la aplicación desarrollada en el TFG. Además, se proporcionará una guía detallada sobre cómo obtener e instalar el software de terceros necesario. Dado que se trata de una aplicación web, se explicará cómo instalar y configurar los servidores necesarios para que funcione la aplicación y cómo realizar su despliegue.

9.1. Prerrequisitos Técnicos

Antes de proceder con la instalación de la aplicación, asegúrese de cumplir con los siguientes prerequisites técnicos:

- **Sistema Operativo:** Windows 10, macOS, o cualquier distribución de Linux.
- **Espacio en Disco:** Al menos 1 GB de espacio libre.
- **Memoria RAM:** Mínimo 4 GB de RAM.
- **Software de Terceros:**
 - XAMPP (versión 7.4 o superior) que incluye Apache, MySQL y PHP.
 - Visual Studio Code (o cualquier otro editor de código de su preferencia).

9.2. Instalación del Software de Terceros

9.2.1. Instalación de XAMPP

1. Descargar XAMPP:

- Visite la página oficial de XAMPP y descargue la versión correspondiente a su sistema operativo.

2. Instalar XAMPP:

- Ejecute el instalador de XAMPP y siga las instrucciones en pantalla.
- Seleccione los componentes necesarios: Apache, MySQL, PHP.
- Finalice la instalación.

3. Configurar XAMPP:

- Abra el Panel de Control de XAMPP.
- Inicie los módulos de Apache y MySQL.
- Verifique que ambos servicios estén funcionando correctamente accediendo a <http://localhost> en su navegador web.

En la figura 9.1 se muestra la app Xampp:

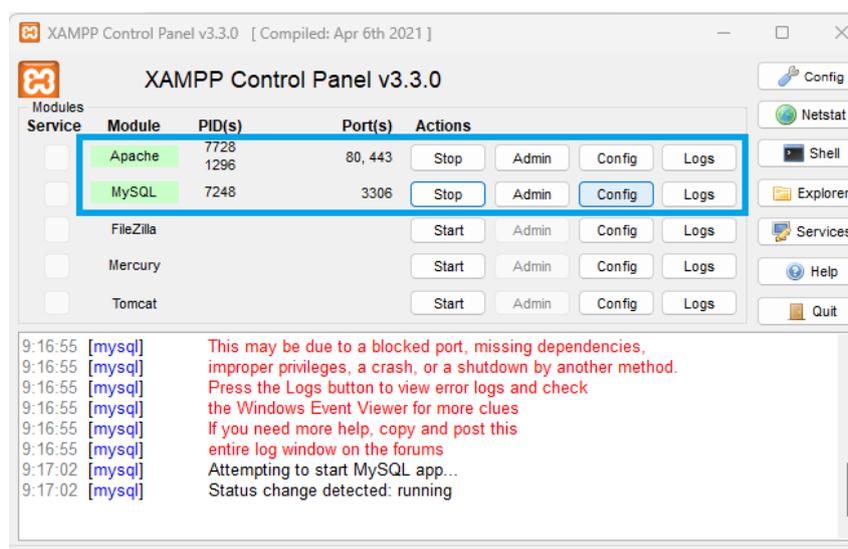


Figura 9.1: App Xampp.

9.2.2. Instalación de Visual Studio Code

1. Descargar Visual Studio Code:

- Visite la página oficial de Visual Studio Code y descargue la versión correspondiente a su sistema operativo.

2. Instalar Visual Studio Code:

- Ejecute el instalador y siga las instrucciones en pantalla.
- Una vez completada la instalación, abra Visual Studio Code.

En la figura 9.2 se muestra la app Visual Studio Code:

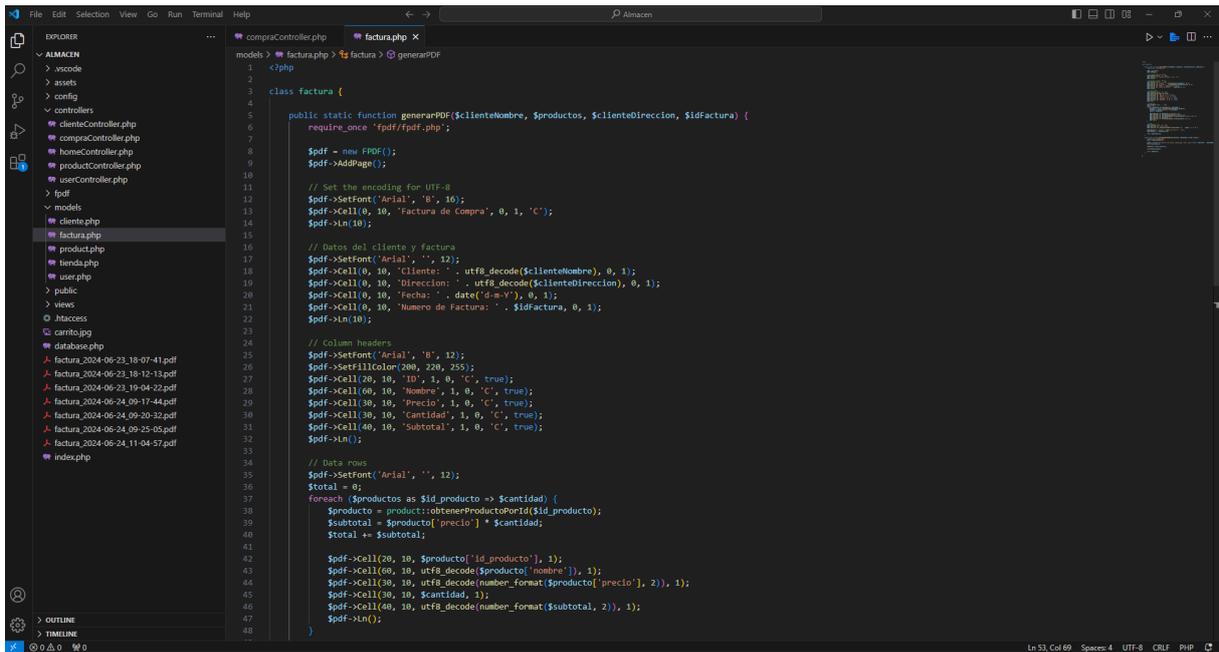


Figura 9.2: App Visual Studio Code.

9.3. Instalación y Configuración del Servidor Web

9.3.1. Configuración de Apache

1. Editar el archivo de configuración de Apache:

- Navegue hasta el directorio de instalación de XAMPP y localice el archivo `httpd.conf` en la carpeta `apache\conf`.
- Abra `httpd.conf` con un editor de texto (puede usar Visual Studio Code).

2. Configurar el DocumentRoot:

- Busque la línea `DocumentRoot 'C:/xampp/htdocs'` y cambie el directorio a la ubicación de su proyecto. Por ejemplo:

```
DocumentRoot "C:/path/to/your/project"  
<Directory "C:/path/to/your/project">
```

3. Guardar los cambios:

- Guarde y cierre el archivo `httpd.conf`.

9.3.2. Configuración de MySQL

1. Crear una nueva base de datos:

- Abra phpMyAdmin accediendo a `http://localhost/phpmyadmin` en su navegador.
- Cree una nueva base de datos para la aplicación. Por ejemplo, `nombre_base_datos`.

2. Importar la estructura de la base de datos:

- Seleccione la base de datos creada y vaya a la pestaña **Importar**.
- Seleccione el archivo SQL con la estructura de la base de datos y haga clic en **Continuar**.

9.4. Despliegue de la Aplicación

1. Copiar los archivos del proyecto:

- Copie todos los archivos de su proyecto en el directorio configurado en el `DocumentRoot` de Apache.

2. Configurar el archivo de conexión a la base de datos:

- Localice el archivo de configuración de la base de datos en su proyecto (por ejemplo, `config.php`).
- Edite los parámetros de conexión para que coincidan con su configuración de MySQL:

```
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "ferreteria";
```

3. Verificar el despliegue:

- Acceda a `http://localhost/tfg1/almacen/index.php` en su navegador.
- Verifique que la aplicación se cargue correctamente y realice una prueba básica para asegurarse de que todo esté funcionando.

Capítulo 10

Manual de Usuario

Este manual está diseñado para proporcionar a los usuarios finales una guía completa y clara sobre cómo utilizar la aplicación. La información contenida aquí está destinada a usuarios no avanzados y podría coincidir con la sección de AYUDA del programa.

10.1. Inicio de Sesión

- **Paso 1:** Abra su navegador web y diríjase a <http://localhost/tfg1/almacen/login.php>.
- **Paso 2:** En la página de inicio de sesión, introduzca su dirección de correo electrónico y contraseña.
- **Paso 3:** Haga clic en el botón "Iniciar Sesión".
- **Nota:** Si no tiene una cuenta, haga clic en "Registrarse" y siga las instrucciones para crear una nueva cuenta. Una vez se haya registrado, espere hasta que el administrador acepte tu registro y puedas acceder a la app con ese usuario y contraseña.

En la figura 10.1 se muestra la pantalla para iniciar sesión en la aplicación:

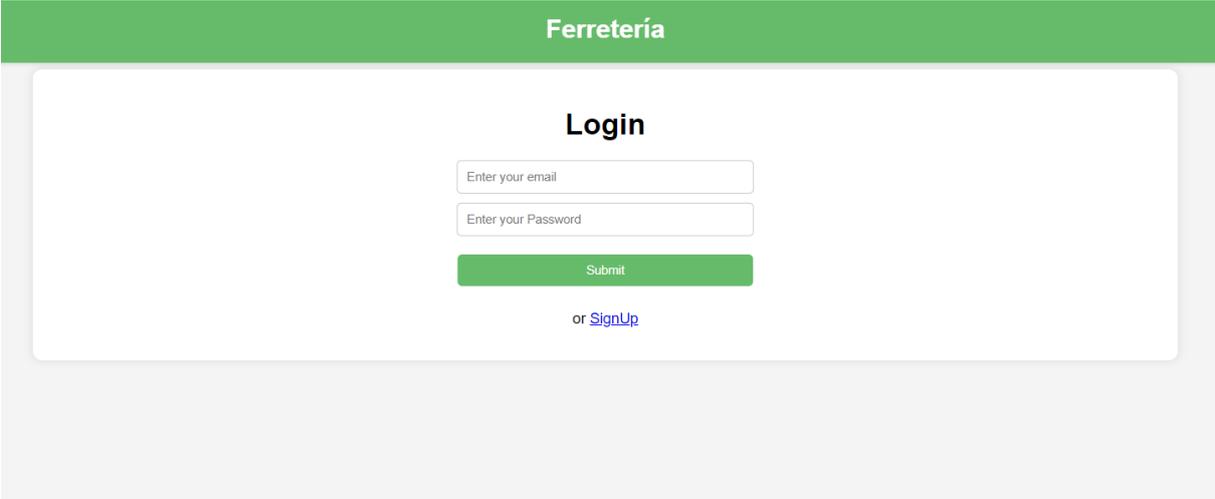
The image shows a web application interface for 'Ferretería'. At the top, there is a green header bar with the word 'Ferretería' in white. Below the header is a white login form. The form is titled 'Login' in bold black text. It contains two input fields: 'Enter your email' and 'Enter your Password'. Below these fields is a green 'Submit' button. Underneath the button, there is a link that says 'or [SignUp](#)'.

Figura 10.1: Iniciar sesión.

10.2. Navegación por el Panel de Usuario

- **Panel Principal:** Una vez iniciado sesión, será redirigido al panel principal donde puede ver y gestionar sus actividades.
- **Menú de Opciones:** Utilice el menú superior para acceder a las diferentes secciones de la aplicación, como "Productos", "Clientes", "Tienda" y "Cuenta" o pulse sobre "Cerrar sesión" para finalizar la sesión.
- **Actualizar Cuenta:** En la sección "Cuenta", puede actualizar su información personal y cambiar su contraseña.
- **Gestionar Carrito:** Dentro de la sección "Tienda", en la parte superior derecha se encuentra un carrito en cual se puede ver los artículos que contiene y se puede finalizar una compra.

En la figura 10.2 se muestra la pantalla principal de la aplicación:



Figura 10.2: Iniciar sesión.

10.3. Administrar Cuenta

En la sección "Cuenta" puede actualizar su información personal y cambiar su contraseña.

- **Editar nombre:** pulsando sobre el botón "Editar" en nombre de usuario, puede editar el nombre de la cuenta y actualizarlo en base de datos. Una vez le de a editar, le saldrá un campo de texto sobre el que podrá escribir su nuevo nombre. Cuando escriba el nombre deseado, pulse sobre el botón guardar y quedará cambiado.
- **Editar correo electrónico:** pulsando sobre el botón "Editar" en correo electrónico, podrá editar el correo electrónico de la cuenta y actualizarlo en base de datos. Una vez le de a editar, le saldrá un campo de texto sobre el que podrá escribir su nuevo correo. Cuando escriba el correo deseado, pulse sobre el botón guardar y quedará cambiado.
- **Editar contraseña:** pulsando sobre el botón "Cambiar contraseña" podrá editar la contraseña de la cuenta y actualizarla en base de datos. Una vez pulsa el botón, le saldrá una ventana nueva donde se le solicitará la contraseña actual, la nueva contraseña y por último, que repita de nuevo la nueva contraseña. Una vez pulse en el botón de cambiar la contraseña, si los datos introducidos son correctos, se cambiará automáticamente la contraseña y se almacenará la nueva en la base de datos.

En la figura 10.3 se muestra la pantalla con la información de la cuenta del usuario:

En la figura 10.4 se muestra la pantalla para cambiar la contraseña de la cuenta:



Figura 10.3: Editar información de la cuenta.



Figura 10.4: Cambiar contraseña.

10.4. Administrar productos

En la sección "Productos" puede ver un listado con todos los productos e información de los mismos, buscar dichos productos por nombre, descripción o código de barras y editarlos.

- **Listar productos:** una vez accede a la página de productos, puede ver una lista con todos los productos de la base de datos, los cuales se pueden ordenar por columnas.
- **Buscar productos:** en la parte superior de la lista de productos se encuentra una barra de búsqueda. En esta barra se puede introducir palabras clave que coincidan con el nombre o la descripción de un producto. También se puede introducir un código de barras con un lector o con el propio teclado.
- **Agregar nuevo producto:** pulsando sobre el botón "Añadir nuevo producto", se abrirá una nueva página con los diferentes campos para poder añadir un nuevo producto. Estos campos son: "Nombre", "Descripción", "Precio" y "Código de barras". Una vez se completan todos los campos y se pulse el botón "Agregar producto", el producto quedará registrado en la base de datos y saldrá en la lista de productos.
- **Editar productos:** para poder editar un producto, en cada fila aparece un botón en la parte derecha de la tabla que pone "Editar". Una vez pulsas dicho botón, se habilitan las celdas para poder editar los diferentes campos de cada producto. Una vez editados, se pulsa el botón "Guardar" y estos cambios se aplican en la base de datos.

En la figura 10.7 se muestra la pantalla para gestionar los productos:

En la figura 10.6 se muestra la pantalla para editar los productos:

Ferretería
Productos Clientes Tienda Cuenta Cerrar sesión

Gestión de Productos

Agregar Nuevo ProductoBuscar

ID ▲	Nombre	Descripción	Precio	Código de Barras	Stock	Acciones
1	Pala de Jardín	Pala de jardín con mango ergonómico	14.49	978020137983	20	Editar Eliminar
2	Rastrillo	Rastrillo de acero para jardín	12.99	978020137984	18	Editar Eliminar
3	Cepillo de Alambre	Cepillo de alambre para limpieza	5.99	978020137985	30	Editar Eliminar
4	Guantes de Trabajo	Guantes de trabajo resistentes al corte	7.99	978020137986	25	Editar Eliminar
5	Taladro Inalámbrico	Taladro inalámbrico con batería recargable	59.99	978020137962	10	Editar Eliminar
6	Caja de Clavos	Caja de clavos de diferentes tamaños	7.49	978020137963	50	Editar Eliminar
7	Llave Inglesa	Llave inglesa ajustable de acero	12.99	978020137964	20	Editar Eliminar
8	Destornillador	Juego de destornilladores de precisión	15.99	978020137965	30	Editar Eliminar
9	Cinta Métrica	Cinta métrica retráctil de 5 metros	5.99	978020137966	40	Editar Eliminar
10	Martillo	Martillo de carpintero con mango de madera	9.99	978020137967	25	Editar Eliminar
11	Sierra Manual	Sierra manual de acero inoxidable	14.99	978020137968	15	Editar Eliminar
12	Tijeras de Podar	Tijeras de podar para jardín	8.99	978020137969	35	Editar Eliminar

Figura 10.5: Gestionar Productos.

Ferretería
Productos Clientes Tienda Cuenta Cerrar sesión

Editar Producto

Nombre:

Descripción:

Precio:

Código de Barras:

Stock:

Actualizar Producto

Figura 10.6: Editar Productos.

10.5. Administrar clientes

En la sección "Clientes" puede ver un listado con todos los clientes e información de los mismos, buscar dichos clientes por nombre, correo, DNI o teléfono y editarlos.

- **Listar clientes:** una vez accede a la página de clientes, puede ver una lista con todos los clientes de la base de datos, los cuales se pueden ordenar por columnas.
- **Buscar clientes:** en la parte superior de la lista de productos se encuentra una barra de búsqueda. En esta barra se puede introducir palabras clave que coincidan con el nombre, correo, DNI, o teléfono.
- **Agregar nuevo cliente:** pulsando sobre el botón "Añadir nuevo cliente", se abrirá una nueva página con los diferentes campos para poder añadir un nuevo cliente. Estos campos son: "Nombre", "Email", "DNI", "Teléfono" y "Dirección". Una vez se completan todos los campos y se pulse el botón "Agregar cliente", el cliente quedará registrado en la base de datos y saldrá en la lista de clientes.
- **Editar clientes:** para poder editar un cliente, en cada fila aparece un botón en la parte derecha de la tabla que pone "Editar". Una vez pulsas dicho botón, se habilitan las celdas para poder editar los diferentes campos de cada cliente. Una vez editados, se pulsa el botón "Guardar" y estos cambios se aplican en la base de datos.

En la figura 10.7 se muestra la pantalla para gestionar los clientes:

En la figura 10.6 se muestra la pantalla para editar los clientes:

Ferretería
 Productos Clientes **Tienda** Cuenta Cerrar sesión

Gestión de Clientes

Agregar Nuevo Cliente

Buscar

ID ▲	Nombre	Email	Dni	Teléfono	Dirección	Acciones
1	Diego Gábana	diego@gmail.com	51434567R	666555999	Dario 4	Editar Eliminar
2	Isabel Castillo	isabel.castillo@gmail.com	51486677F	622445566	Calle Olivo 15	Editar Eliminar
3	Pepe Gotera	pepe@gmail.com	51486596T	652365230	Calle Las Peñas 148	Editar Eliminar
4	Manuel Herrera	manuel.herrera@gmail.com	51486778G	633556677	Avenida Almendros 11	Editar Eliminar
5	Alberto Méndez	alberto.mendez@gmail.com	51486980I	655778899	Calle Haya 18	Editar Eliminar
6	Laura Fernández	laura.fernandez@gmail.com	51486657L	611223344	Calle Mayor 23	Editar Eliminar
7	Carlos Pérez	carlos.perez@gmail.com	51486758M	622334455	Avenida Libertad 45	Editar Eliminar
8	Ana Gómez	ana.gomez@gmail.com	51486859N	633445566	Calle Sol 8	Editar Eliminar
9	Javier Ruiz	javier.ruiz@gmail.com	51486960O	644556677	Calle Luna 17	Editar Eliminar
10	María López	maria.lopez@gmail.com	51486061P	655667788	Calle Estrella 32	Editar Eliminar
11	Pedro Sánchez	pedro.sanchez@gmail.com	51486162Q	666778899	Calle Rio 10	Editar Eliminar
12	Lucía Martín	lucia.martin@gmail.com	51486263R	677889900	Calle Viento 25	Editar Eliminar

Figura 10.7: Gestionar Clientes.

Ferretería
 Productos Clientes **Tienda** Cuenta Cerrar sesión

Editar Cliente

Nombre:

Email:

Dni:

Teléfono:

Dirección:

Actualizar Cliente

Figura 10.8: Editar Clientes.

10.6. Administrar tienda

En la sección "Tienda" puede ver un listado con todos los productos e información de los mismos, buscar dichos productos por nombre, descripción o código de barras y añadirlos al carrito.

- **Listar productos:** una vez accede a la página de productos, puede ver una lista con todos los productos de la base de datos, los cuales se pueden ordenar por columnas.
- **Buscar productos:** en la parte superior de la lista de productos se encuentra una barra de búsqueda. En esta barra se puede introducir palabras clave que coincidan con el nombre o la descripción de un producto. También se puede introducir un código de barras con un lector o con el propio teclado.
- **Añadir productos a carrito:** para poder añadir un producto al carrito, primero hay que poner un número de cantidad de producto. Este número nunca puede ser superior al stock disponible de ese producto. Si ponemos un número y pulsamos al botón "Añadir", se añadirá el producto con la cantidad indicada. En el caso en el que no haya stock suficiente, saldrá un mensaje diciendo que no hay suficiente stock y no se añadirá al carrito.

En la figura 10.9 se muestra la pantalla de la tienda con la opción del carrito:

ID	Nombre	Descripción	Precio	Código de Barras	Stock	Añadir al Carrito
1	Pala de Jardín	Pala de jardín con mango ergonómico	14.49	978020137983	20	1 Añadir
2	Rastrillo	Rastrillo de acero para jardín	12.99	978020137984	18	1 Añadir
3	Cepillo de Alambre	Cepillo de alambre para limpieza	5.99	978020137985	30	1 Añadir
4	Guantes de Trabajo	Guantes de trabajo resistentes al corte	7.99	978020137986	25	1 Añadir
5	Taladro Inalámbrico	Taladro inalámbrico con batería recargable	59.99	978020137962	10	1 Añadir
6	Caja de Clavos	Caja de clavos de diferentes tamaños	7.49	978020137963	50	1 Añadir
7	Llave Inglesa	Llave inglesa ajustable de acero	12.99	978020137964	20	1 Añadir
8	Destornillador	Juego de destornilladores de precisión	15.99	978020137965	30	1 Añadir

Figura 10.9: Gestionar Tienda.

10.7. Carrito

En la sección carrito, se puede ver un listado con los productos añadidos al carrito y finalizar esa compra.

- **Listar productos:** una vez accede a la página de carrito, puede ver una lista con todos los productos seleccionados para la compra, los cuales han sido anteriormente añadidos al carrito.
- **Actualizar cantidad de un producto:** en cada fila se muestra un producto y la cantidad que se va a comprar. Dicha cantidad se puede actualizar con la condición que nunca puede mayor al número de stock de ese producto. Una vez se cambia la cantidad, se actualiza el precio total de la compra.
- **Eliminar producto:** como se menciona en el anterior apartado, en cada fila se muestra un producto. En la parte decha de cada producto, aparece un botón "Eliminar", el cual elimina ese producto del carrito y se actualiza el precio total de la compra.
- **Seleccionar método de pago:** se muestran dos opciones para realizar el pago, por tarjeta de crédito/débito(móvil) o en efectivo, dependiendo de lo que solicite el cliente.
- **Buscar clientes:** se debe introducir palabras clave que coincidan con el nombre, correo, DNI, o teléfono para poder buscar un cliente y asociarle la compra. Una vez se pulsa el botón "Buscar", aparece un listado con los posibles clientes y se selecciona el deseado.
- **Finalizar compra:** una vez realizados y revisados los pasos anteriores, se pulsa sobre el botón "Finalizar compra" y la compra quedará finalizada. Posteriormente se generará una factura que se guardará como PDF con su identificador en la carpeta principal de la app.

En la figura 10.10 se muestra la pantalla del carrito para poder finalizar la compra:

Ferretería
Productos Clientes Tienda Cuenta Cerrar sesión

Carrito de Compras

ID	Nombre	Precio Unitario	Cantidad	Subtotal	Acciones
1	Pala de Jardín	14.49	<input type="text" value="1"/>	14.49	Actualizar Eliminar
3	Cepillo de Alambre	5.99	<input type="text" value="1"/>	5.99	Actualizar Eliminar
4	Guantes de Trabajo	7.99	<input type="text" value="2"/>	15.98	Actualizar Eliminar
5	Taladro Inalámbrico	59.99	<input type="text" value="1"/>	59.99	Actualizar Eliminar

Seleccione método de pago:

Buscar cliente por DNI o teléfono:

[Buscar](#)

Total de la compra: 96.45€

Figura 10.10: Gestionar Carrito.

10.8. Cerrar Sesión

En la barra superior, como se menciona anteriormente, aparece un campo llamado "Cerrar Sesión". Si un usuario logueado pulsa sobre dicho botón, se cerrará la sesión y se le redirigirá hasta la página de login.

En la figura 10.11 se muestra la opción para cerrar sesión:



Figura 10.11: Cerrar Sesión.

Capítulo 11

Manual de Administración

Este manual está dirigido a los administradores de la aplicación y proporciona instrucciones detalladas sobre cómo gestionar y mantener el sistema.

11.1. Acceso al Panel de Administración

- **Paso 1:** Inicie sesión con una cuenta de administrador.
- **Paso 2:** Acceda al panel de administración desde el menú superior.

11.2. Gestión de Usuarios

- **Aceptar o denegar nuevo usuario:**
 - Acceda a "Gestión de Usuarios" y haga clic en "Aceptar Usuario" si lo que quiere es aceptarlo.
 - Acceda a "Gestión de Usuarios" y haga clic en "Denegar Usuario" si lo que quiere es denegarlo.

11.3. Gestión de la Base de Datos

- **Copia de Seguridad:**
 - Acceda a phpMyAdmin desde <http://localhost/phpmyadmin>.
 - Seleccione la base de datos de la aplicación.
 - Vaya a la pestaña "Exportar" y siga las instrucciones para crear una copia de seguridad.
- **Restauración de Datos:**

- Acceda a phpMyAdmin y seleccione la base de datos.
- Vaya a la pestaña "Importar" y cargue el archivo de la copia de seguridad.

Para poder prevenir problemas con la base de datos, siempre que se pueda (recomendación 1 vez a la semana) hay que hacer una copia de la base de datos. Para ello, dentro del enlace <http://localhost/phpmyadmin> acceder a la base de datos de la aplicación y pulsar sobre el boton que pone exportar. Se descargará un archivo en .sql el cuál se podrá volver a importar si algo falla en la base de datos. Para ello, en la figura 11.1 se muestra la opción de importar y exportar en la base de datos:

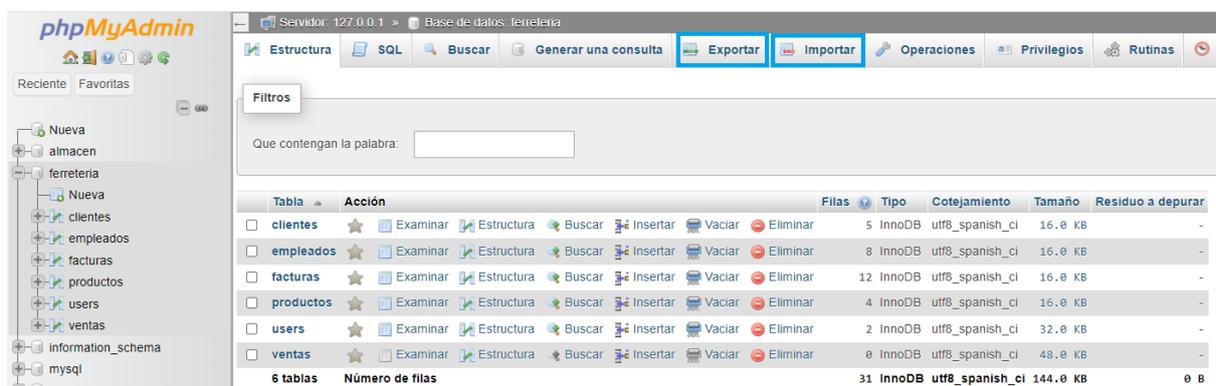


Figura 11.1: Importar/exportar base de datos.

11.4. Configuración del Servidor

- **Configuración de Apache:**
 - Acceda al archivo `httpd.conf` en el directorio de instalación de XAMPP.
 - Realice las configuraciones necesarias, como cambios en el `DocumentRoot`.
- **Configuración de MySQL:**
 - Asegúrese de que el servicio de MySQL esté corriendo.
 - Verifique las configuraciones de usuario y permisos en phpMyAdmin.

Bibliografía

- [1] PHP Documentation Group. *PHP: Hypertext Preprocessor*. Accessed: 2021-10-12. 2021. URL: <https://www.php.net/docs.php>.
- [2] Robin Nixon. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. O'Reilly Media, Inc., 2018.
- [3] W3Schools. *PHP Tutorial*. Accessed: 2021-10-10. 2021. URL: <https://www.w3schools.com/php/>.
- [4] Luke Welling y Laura Thomson. *PHP and MySQL Web Development*. Addison-Wesley Professional, 2009.