Pelis Zamora

Tutor: María Criado Domínguez

Trabajo de fin de ciclo

Desarrollo de aplicaciones web

Contenido

[Tabla de Diagramas 2](#_Toc136194806)

[Tabla de Ilustraciones 2](#_Toc136194807)

[Tabla de Código 2](#_Toc136194808)

[Tabla de Pruebas 2](#_Toc136194809)

[Introducción 4](#_Toc136194810)

[Análisis de la aplicación 4](#_Toc136194811)

[Requisitos 4](#_Toc136194812)

[Requisitos funcionales 4](#_Toc136194813)

[Requisitos no funcionales 4](#_Toc136194814)

[Recursos necesarios para el desarrollo 5](#_Toc136194815)

[Recurso Hardware 5](#_Toc136194816)

[Recurso Software 5](#_Toc136194817)

[Metodología 5](#_Toc136194818)

[Metodología de desarrollo 5](#_Toc136194819)

[Planificación 5](#_Toc136194820)

[Diseño del proyecto 8](#_Toc136194821)

[Diseño de la BBDD 8](#_Toc136194822)

[Modelo E-R 8](#_Toc136194823)

[Modelo Relacional 9](#_Toc136194824)

[Interacción con el usuario 10](#_Toc136194825)

[Casos de uso 10](#_Toc136194826)

[Diseño de la interfaz 23](#_Toc136194827)

[*Wireframes de la tienda* 23](#_Toc136194828)

[Diseño de la arquitectura 30](#_Toc136194829)

[Arquitectura del servidor 30](#_Toc136194830)

[Arquitectura del cliente 31](#_Toc136194831)

[Detalles de implementación 31](#_Toc136194832)

[Desarrollo de la aplicación 31](#_Toc136194833)

[Implementación real 36](#_Toc136194834)

[Fase de pruebas 37](#_Toc136194835)

[Pruebas 37](#_Toc136194836)

[Trabajo Futuro 40](#_Toc136194837)

[Conclusiones 40](#_Toc136194838)

[Bibliografía 40](#_Toc136194839)

# Tabla de Diagramas

[Diagrama 1 D. de Gantt 0](#_Toc136020379)

[Diagrama 2 Modelo E-R 0](#_Toc136020380)

[Diagrama 3 Modelo Relacional 1](#_Toc136020381)

[Diagrama 4 Caso de uso 2](#_Toc136020382)

# Tabla de Ilustraciones

[Ilustración 1 Imagen de la metodología en espiral 4](#_Toc136020864)

[Ilustración 2 Wireframe del Login 15](#_Toc136020865)

[Ilustración 3 Wireframe del Registro 15](#_Toc136020866)

[Ilustración 4 WIreframe del Home 16](#_Toc136020867)

[Ilustración 5 Wireframe del detalle del producto 16](#_Toc136020868)

[Ilustración 6 Wireframe de la vista del carrito 17](#_Toc136020869)

[Ilustración 7 Wireframe de las opciones del usuario 17](#_Toc136020870)

[Ilustración 8 Wireframe del formulario de edición de datos 18](#_Toc136020871)

[Ilustración 9 Wireframe del Historial de pedidos 18](#_Toc136020872)

[Ilustración 10 Wireframe del Historial de pedido 19](#_Toc136020873)

[Ilustración 11 Wireframe de la Confirmación de compra 19](#_Toc136020874)

[Ilustración 12 Wireframe del Formulario de contacto 20](#_Toc136020875)

[Ilustración 13 Wireframe de la vista el inicio del usuario 20](#_Toc136020876)

[Ilustración 14 Wireframe de la vista de las tablas 21](#_Toc136020877)

[Ilustración 15 Mapa de navegación de la tienda 21](#_Toc136020878)

[Ilustración 16 Mapa de navegación del administrador 22](#_Toc136020879)

[Ilustración 17 Estructura del proyecto 23](#_Toc136020880)

# Tabla de Código

[Código 1 Anotaciones de Lombok 23](#_Toc136020881)

[Código 2 Métodos de encriptación y desencriptación 24](#_Toc136020882)

[Código 3 Creación de la tabla de la factura 25](#_Toc136020883)

[Código 4 Creación del cuerpo del mensaje 26](#_Toc136020884)

[Código 5 Código de las anotaciones de JPA 27](#_Toc136020885)

[Código 6 Código de las librerías de Simple Datatables 27](#_Toc136020886)

# Tabla de Pruebas

[Tabla 1 Prueba del inicio de sesión 28](#_Toc136020539)

[Tabla 2 Prueba del registro de un usuario 28](#_Toc136020540)

[Tabla 3 Prueba de la edición de datos de un usuario 28](#_Toc136020541)

[Tabla 4 Prueba del cierre de sesión 28](#_Toc136020542)

[Tabla 5 Prueba de la compra 28](#_Toc136020543)

[Tabla 6 Prueba de la consulta del historial 28](#_Toc136020544)

[Tabla 7 Prueba de la vista del carrito 29](#_Toc136020545)

[Tabla 8 Prueba de la realización de altas y bajas 29](#_Toc136020546)

[Tabla 9 Prueba de añadir y editar datos 29](#_Toc136020547)

[Tabla 10 Prueba de aceptar y cancelar pedidos 29](#_Toc136020548)

[Tabla 11 Prueba de editar datos como empleado 29](#_Toc136020549)

# Introducción

Debido a plataformas de streaming como Netflix o HBO, la tienda Pelis Zamora ha visto que sus ventas se han reducido, por lo tanto, han decidido informatizarse, creando una plataforma de venta de películas en físico en la web, así con esta tienda intentaran luchar contra la competencia que les generan estas grandes plataformas.

La idea principal del proyecto es crear una aplicación web orientada tanto al uso de una tienda desde el punto de vista del cliente como la gestión por parte de los empleados de esta.

Los clientes podrán comprar los productos ofrecidos y los dueños podrán gestionar los usuarios, productos, facturas y albaranes.

El objetivo es crear una tienda en la que los clientes puedan comprar películas catalogadas por categorías, al hacer su compra se generara una factura, la cual mostrara los todos los datos relacionados con la compra del cliente, la aplicación permitirá a los trabajadores de la tienda, que estarán catalogados como un solo administrador, el cual será el dueño y tendrá acceso a todo, y como moderadores, los cuales serán los empleados de la tienda, estos últimos solo podrán aceptar y cancelar pedidos y editar los valores de las tablas.

# Análisis de la aplicación

## Requisitos

### Requisitos funcionales

R1: Como moderador podrá aceptar y cancelar pedidos, y editar las tablas.

R2: Como administrador, poder añadir, eliminar y modificar los datos de las tablas usando la interfaz de la aplicación.

R3: Como cliente, poder realizar compras en la aplicación, poder ver los productos sin tener sesión iniciada, ver su carrito, ver su historial.

R4: Todos los usuarios podrán acceder a su perfil, ver sus datos y poder modificarlos, registrarse e iniciar sesión.

### Requisitos no funcionales

R1: Uso de un lenguaje orientado a objetos

R2: Uso de una arquitectura cliente-servidor

R3: Se podrá usar desde distintos navegadores

# Recursos necesarios para el desarrollo

## Recurso Hardware

* Conexión a internet
* Servidor web
* Ordenador i5 y 8 gb de RAM

## Recurso Software

* Parte del Cliente
  + Navegador
  + IDE eclipse for java and web Developers
  + MariaDB
  + Java 1.8.0\_212
  + XAMPP 3.2.3
* Parte del Administrador
  + Navegador
  + SpringBoot IDE
  + MariaDB
  + Java 17
  + SpringBoot
  + XAMPP 3.2.3

# Metodología

## Metodología de desarrollo

Para el desarrollo del proyecto se ha usado una metodología en espiral, pudiendo repasar todas las partes del proyecto en cada entrega. He utilizado este tipo de metodología ya que me permite volver a partes anteriores del proyecto en caso de que fuera necesario repasar o actualizar partes de él.

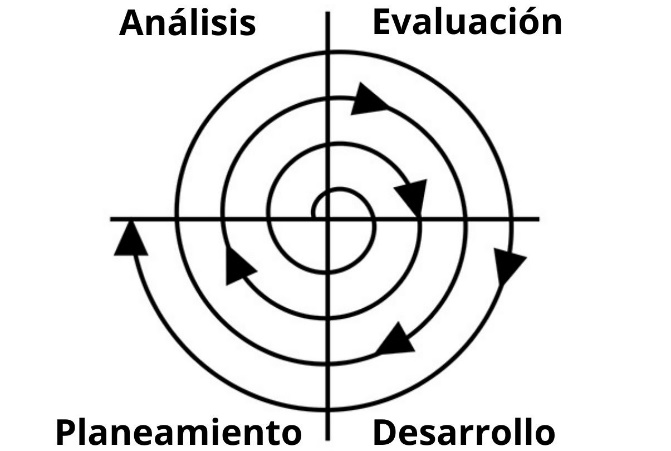


Ilustración Imagen de la metodología en espiral

## Planificación

**1ª Fase (Aproximadamente 35 horas):**

* Análisis e investigación del diseño (5h)
* Planificación de los objetivos (5h)
* Planificación de los medios Hardware y software (5h)
* Realización de los casos de uso principales(5h)
* Realización de los diagramas de base de datos(5h)
* Prototipo de la aplicación (5h)
* Documentación(5h)

**2ª Fase: (45 horas)**

* Creación de la base de datos (5h)
* Primera versión de la aplicación(25h)
  + Pagina inicial y login
  + Gestión de usuarios y de empleados
* Realización de pruebas(5h)
* Corrección de errores(5h)
* Documentación(5h)

**3ª Fase: (40 horas)**

* Segunda versión de la aplicación(20h)
  + Gestión de productos
  + Gestión de ventas
* Corrección de errores (10h)
* Realización de pruebas (5h)
* Documentación(5h)

**4ª Fase: (30 horas)**

* Tercera versión de la aplicación(10h)
  + Gestión de pedidos
* Corrección de errores(10h)
* Realización de pruebas(5h)
* Documentación(5h)

**5ª Fase (35 horas)**

* Ultima versión de la aplicación(10h)
* Análisis de los objetivos cumplidos(5h)
* Últimas pruebas(5h)
* Depuración de código(5h)
* Elaboración de manual(5h)
* Documentación final (5h)

Gráfico

Descripción generada automáticamente

Diagrama D. de Gantt

# Diseño del proyecto

## Diseño de la BBDD

Modelo E-R Diagrama

Descripción generada automáticamente

Diagrama Modelo E-R

### Modelo Relacional

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama Modelo Relacional

## Interacción con el usuario

### Casos de uso

Diagrama

Descripción generada automáticamente

Diagrama Caso de uso

#### Caso de uso del inicio de sesión

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios inicien sesión en el sistema |
| Precondiciones | El usuario debe estar registrado en el sistema |
| Flujo Principal | 1. El usuario ingresa su nombre de usuario y contraseña. 2. El sistema verifica las credenciales ingresadas. 3. El sistema autentica al usuario y muestra la página principal. |
| Flujos Alternativos | A1: Credenciales incorrectas:   1. El sistema muestra un mensaje de error y solicita al usuario que ingrese nuevamente las credenciales. 2. El usuario reintenta ingresar las credenciales. 3. El sistema vuelve al paso 2 del flujo principal.   A2: Usuario no registrado:   1. El sistema muestra un mensaje indicando que el usuario no está registrado y proporciona opciones para registrarse. |
| Postcondiciones | El usuario ha iniciado sesión y tiene acceso a las funciones y datos del sistema. |
| Requerimientos no funcionales | El sistema debe proteger la confidencialidad de las credenciales de los usuarios.  El sistema debe ser capaz de manejar múltiples usuarios concurrentes.  El tiempo de respuesta del sistema para verificar las credenciales debe ser aceptable para los usuarios. |

#### Caso de uso del registro

|  |  |
| --- | --- |
| Actores | Cliente, Sistema |
| Objetivo | Permitir que los usuarios se registren en el sistema |
| Precondiciones | El usuario no debe estar registrado previamente en el sistema |
| Flujo Principal | 1. El usuario accede a la página de registro. 2. El usuario ingresa la información requerida, como nombre, dirección de correo electrónico y contraseña. 3. El sistema verifica que la dirección de correo electrónico no esté previamente registrada en el sistema. 4. El sistema crea una cuenta para el usuario utilizando la información proporcionada. 5. El sistema muestra un mensaje de confirmación de registro exitoso. |
| Flujos Alternativos | A1: Correo electrónico ya registrado:   1. El sistema muestra un mensaje indicando que la dirección de correo electrónico ya está asociada con una cuenta. 2. El usuario puede intentar registrarse con una dirección de correo electrónico diferente o recuperar la contraseña de la cuenta existente. 3. El sistema vuelve al paso 2 del flujo principal.   A2: Campos obligatorios no completados:   1. El sistema muestra un mensaje indicando que algunos campos obligatorios no han sido completados. 2. El usuario completa los campos faltantes. 3. El sistema vuelve al paso 3 del flujo principal.   A3: Error en la conexión o en la creación de la cuenta:   1. El sistema muestra un mensaje de error indicando que no se pudo completar el registro. |
| Postcondiciones | El usuario ha completado el registro y puede iniciar sesión en el sistema utilizando las credenciales proporcionadas. |
| Requerimientos no funcionales | El sistema debe proteger la confidencialidad de la información del usuario durante el registro.  El sistema debe ser capaz de manejar múltiples registros concurrentes.  El tiempo de respuesta del sistema para verificar y crear la cuenta debe ser aceptable para los usuarios. |

#### Caso de uso de la edición de los datos del usuario

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios editen sus propios datos en el sistema |
| Precondiciones | El usuario debe haber iniciado sesión en el sistema |
| Flujo Principal | 1. El usuario accede a la configuración o perfil de su cuenta.  2. El sistema muestra los datos actuales del usuario, como nombre, dirección de correo electrónico, etc.  3. El usuario realiza las modificaciones deseadas en los campos de datos.  4. El sistema valida y actualiza los cambios realizados por el usuario.  5. El sistema muestra un mensaje de confirmación de que los cambios se han guardado correctamente. |
| Flujos Alternativos | A1: Campos no válidos o faltantes:   1. El sistema muestra un mensaje indicando que algunos campos no son válidos o están incompletos. 2. El usuario corrige los campos incorrectos o faltantes. 3. El sistema vuelve al paso 4 del flujo principal.   A2: Error en la conexión o en la actualización de datos:   1. El sistema muestra un mensaje de error indicando que no se pudo completar la actualización de datos. |
| Postcondiciones | Los cambios realizados por el usuario se han guardado correctamente en el sistema. |
| Requerimientos no funcionales | El sistema debe proteger la integridad y confidencialidad de los datos del usuario durante la edición.  El tiempo de respuesta del sistema para validar y guardar los cambios realizados por el usuario debe ser aceptable para los usuarios. |

#### Caso de uso del cierre de sesión

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios cierren su sesión en el sistema |
| Precondiciones | El usuario debe haber iniciado sesión en el sistema |
| Flujo Principal | 1. El usuario accede a la opción de cierre de sesión o cierre de cuenta. 2. El sistema finaliza la sesión del usuario y elimina la información de autenticación. 3. El sistema muestra un mensaje de confirmación de que la sesión se ha cerrado correctamente. |
| Flujos Alternativos | A1: Sesión ya cerrada:   1. El sistema muestra un mensaje indicando que el usuario ya ha cerrado su sesión.   A2: Error en el cierre de sesión:   1. El sistema muestra un mensaje de error indicando que no se pudo completar el cierre de sesión. |
| Postcondiciones | La sesión del usuario se ha cerrado correctamente y no tiene acceso a las funciones y datos del sistema. |
| Requerimientos no funcionales | El sistema debe asegurarse de que los datos y la información del usuario estén protegidos después del cierre de sesión.  El tiempo de respuesta del sistema para cerrar la sesión debe ser aceptable para los usuarios. |

#### Caso de uso de la edición de tablas

|  |  |
| --- | --- |
| Actores | Empleado, Sistema |
| Objetivo | Permitir que los usuarios editen los datos de las tablas en el sistema |
| Precondiciones | El usuario debe tener permisos de edición y acceso a las tablas correspondientes |
| Flujo Principal | 1. El usuario accede a la tabla específica que desea editar. 2. El sistema muestra los datos actuales de la tabla en una interfaz de edición. 3. El usuario realiza las modificaciones deseadas en los campos de datos de la tabla. 4. El sistema valida y actualiza los cambios realizados por el usuario en la tabla. 5. El sistema muestra un mensaje de confirmación de que los cambios se han guardado correctamente. |
| Flujos Alternativos | A1: Campos no válidos o faltantes:   1. El sistema muestra un mensaje indicando que algunos campos no son válidos o están incompletos. 2. El usuario corrige los campos incorrectos o faltantes. 3. El sistema vuelve al paso 4 del flujo principal.   A2: Error en la conexión o en la actualización de datos:   1. El sistema muestra un mensaje de error indicando que no se pudo completar la actualización de datos. |
| Postcondiciones | Los cambios realizados por el usuario se han guardado correctamente en la tabla correspondiente. |
| Requerimientos no funcionales | El sistema debe proteger la integridad y confidencialidad de los datos de la tabla durante la edición.  El tiempo de respuesta del sistema para validar y guardar los cambios realizados por el usuario en la tabla debe ser aceptable para los usuarios. |

#### Caso de uso del envío o cancelación de pedidos

|  |  |
| --- | --- |
| Actores | Empleados o administradores, Sistema |
| Objetivo | Permitir que los usuarios acepten y cancelen pedidos en el sistema |
| Precondiciones | El usuario debe tener los permisos necesarios y acceso a los pedidos correspondientes.  Debe de haber pedidos para enviar o cancelar |
| Flujo Principal | 1. El usuario inicia sesión 2. El sistema muestra la lista de pedidos pendientes para su revisión. 3. El usuario acepta o cancela pedidos. 4. El sistema registra la decisión del usuario y actualiza el estado del pedido en consecuencia. 5. El sistema genera una factura con los datos del pedido. 6. El sistema envía esta factura por correo electrónico al cliente. |
| Postcondiciones | El pedido a cambiado su estado dependiendo del apartado en el que se encontrara, ha enviado o cancelado |
| Requerimientos no funcionales | El sistema debe garantizar la integridad y la consistencia de los datos del pedido al actualizar su estado.  El tiempo de respuesta del sistema para registrar la decisión del usuario y actualizar el estado del pedido debe ser aceptable para los usuarios. |

#### Caso de uso de añadir y editar datos de las tablas

|  |  |
| --- | --- |
| Actores | Administrador, Sistema |
| Objetivo | Permitir que los usuarios añadan y editen datos en las tablas del sistema |
| Precondiciones | El usuario debe tener los permisos necesarios y acceso a las tablas correspondientes |
| Flujo Principal | 1. El usuario accede a la tabla específica en la que desea añadir o editar datos. 2. El sistema muestra la interfaz de la tabla con los datos actuales. 3. El usuario selecciona la opción de añadir nuevos datos o editar datos existentes. 4. El usuario introduce los nuevos datos o realiza las modificaciones deseadas en los campos correspondientes. 5. El sistema valida y registra los cambios realizados por el usuario en la tabla. |
| Flujos Alternativos | A1: Campos no válidos o faltantes:   1. El sistema muestra un mensaje indicando que algunos campos no son válidos o están incompletos. 2. El usuario corrige los campos incorrectos o faltantes. 3. El sistema vuelve al paso 5 del flujo principal. |
| Postcondiciones | Los nuevos datos han sido añadidos o los datos existentes han sido editados correctamente en la tabla correspondiente. |
| Requerimientos no funcionales | El sistema debe proteger la integridad y la consistencia de los datos al añadir o editar registros en las tablas.  El tiempo de respuesta del sistema para validar y registrar los cambios realizados por el usuario en la tabla debe ser aceptable para los usuarios. |

#### Caso de uso de dar de alta y de baja datos de las tablas

|  |  |
| --- | --- |
| Actores | Administrador, Sistema |
| Objetivo | Permitir que los usuarios den de alta y baja campos en las tablas del sistema |
| Precondiciones | El usuario debe tener los permisos necesarios y acceso a las tablas correspondientes.  Debe de haber filas dadas de alta o de baja |
| Flujo Principal | 1. El usuario accede a la vista de la tabla. 2. El usuario selecciona la opción de dar de baja un campo. 3. En caso de dar de baja un campo de dar de baja un campo, este dejara de aparecer para los clientes 4. El usuario accede a la vista de la tabla 5. El usuario selecciona la opción de dar de alta un campo 6. En caso de dar de alta un campo, este empezara a aparecer para los clientes |
| Postcondiciones | El campo ha sido dado de alta o baja correctamente en la estructura de la tabla correspondiente. |
| Requerimientos no funcionales | El sistema debe garantizar la integridad y consistencia de los datos al dar de alta o baja campos en las tablas.  El tiempo de respuesta del sistema para validar y registrar los cambios realizados por el usuario en la estructura de la tabla debe ser aceptable para los usuarios. |

#### Caso de uso de la compra

|  |  |
| --- | --- |
| Actores | Cliente, Sistema |
| Objetivo | Permitir que los usuarios compren productos en el sistema |
| Precondiciones | El usuario debe haber iniciado sesión y tener una cuenta activa |
| Flujo Principal | 1. El usuario navega por el catálogo de productos y selecciona el producto que desea comprar. 2. El usuario selecciona la opción de "Agregar al carrito". 3. El sistema verifica la disponibilidad del producto. 4. Si el producto está disponible y hay suficiente cantidad: 5. El sistema registra el pedido del usuario. 6. El sistema muestra un resumen del pedido y solicita la información de pago y envío. 7. El usuario ingresa la información de pago y envío. 8. El sistema valida la información y procesa el pago. |
| Flujos Alternativos | A1: Producto no disponible o cantidad insuficiente:  a. El sistema no añade el producto al carrito. |
| Postcondiciones | El usuario ha realizado la compra exitosamente y se ha generado un pedido en el sistema. |
| Requerimientos no funcionales | El tiempo de respuesta del sistema para procesar la compra y generar la confirmación del pedido debe ser aceptable para los usuarios. |

#### Caso de uso del historial de pedidos

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios vean su historial de pedidos en el sistema |
| Precondiciones | El usuario debe haber iniciado sesión y tener una cuenta activa |
| Flujo Principal | 1. El usuario accede a su cuenta o perfil en el sistema. 2. El sistema muestra las opciones de navegación del usuario. 3. El usuario selecciona la opción de "Historial de Pedidos". 4. El sistema recupera y muestra una lista de los pedidos previos del usuario. 5. El usuario selecciona un pedido específico para ver los detalles. 6. El sistema muestra los detalles del pedido, como productos, fecha, estado, etc. 7. El usuario puede navegar de regreso al historial de pedidos para ver otros pedidos. |
| Flujos Alternativos | A1: No hay historial de pedidos disponible:   1. No se muestra nada en la vista 2. El usuario vuelve a la página principal o continúa navegando en el sistema |
| Postcondiciones | El usuario ha visualizado su historial de pedidos y los detalles de un pedido específico según su elección. |
| Requerimientos no funcionales | El sistema debe proteger la confidencialidad y privacidad de los datos del historial de pedidos del usuario.  El tiempo de respuesta del sistema para recuperar y mostrar el historial de pedidos debe ser aceptable para los usuarios. |

#### Caso de uso de la solicitud de cancelación

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios soliciten la cancelación de un pedido en el sistema |
| Precondiciones | El usuario debe haber iniciado sesión y tener una cuenta activa |
| Flujo Principal | 1. El usuario accede a su historial de pedidos en el sistema. 2. El sistema muestra una lista de los pedidos previos del usuario. 3. El usuario selecciona el pedido específico que desea cancelar. 4. El usuario selecciona la opción de "Solicitar Cancelación" o similar. 5. El sistema registra la solicitud de cancelación y cambia el estado del pedido. |
| Flujos Alternativos | A1: Pedido ya ha sido enviado o entregado:   1. El sistema muestra un mensaje indicando que el pedido no se puede cancelar porque ya ha sido enviado o entregado. 2. El usuario puede optar por contactar al servicio al cliente para resolver el problema. |
| Postcondiciones | El usuario ha solicitado la cancelación de un pedido y el estado del pedido ha sido actualizado en el sistema. |
| Requerimientos no funcionales | El sistema debe manejar correctamente las solicitudes de cancelación de pedidos según las políticas establecidas.  El tiempo de respuesta del sistema para procesar la solicitud de cancelación y actualizar el estado del pedido debe ser aceptable para los usuarios. |

#### Caso de uso de la solicitud de ver el carrito

|  |  |
| --- | --- |
| Actores | Usuario, Sistema |
| Objetivo | Permitir que los usuarios vean los productos agregados en su carrito de compras |
| Precondiciones | El usuario debe haber iniciado sesión y tener una cuenta activa |
| Flujo Principal | 1. El usuario accede al sistema y navega por el catálogo de productos. 2. El usuario selecciona los productos que desea agregar al carrito. 3. El sistema agrega los productos seleccionados al carrito del usuario. 4. El usuario selecciona la opción del carrito. 5. El sistema muestra una lista de los productos agregados en el carrito, incluyendo detalles como nombre, cantidad, precio, etc. 6. El usuario puede ajustar la cantidad de los productos en el carrito o eliminar productos si es necesario. 7. El usuario puede continuar navegando en el catálogo de productos o proceder al proceso de compra. |
| Flujos Alternativos | A1: Carrito vacío:   1. El usuario puede continuar navegando en el catálogo de productos para agregar productos al carrito. |
| Postcondiciones | El usuario ha visualizado los productos agregados en su carrito de compras y puede realizar ajustes si es necesario. |
| Requerimientos no funcionales | El sistema debe asegurar la consistencia y actualización de los productos en el carrito del usuario.  El tiempo de respuesta del sistema para mostrar el contenido del carrito y realizar ajustes debe ser aceptable para los usuarios. |

Diseño de la interfaz

#### Wireframes de la tienda

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración Wireframe del Login

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración Wireframe del Registro

Diagrama

Descripción generada automáticamente

Ilustración WIreframe del Home

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración Wireframe del detalle del producto

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración Wireframe de la vista del carrito

Texto

Descripción generada automáticamente

Ilustración Wireframe de las opciones del usuario

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Ilustración Wireframe del formulario de edición de datos

Interfaz de usuario gráfica, Texto, Tabla

Descripción generada automáticamente con confianza media

Ilustración Wireframe del Historial de pedidos

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente con confianza media

Ilustración Wireframe del detalle del pedido

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración Wireframe del Formulario de contacto

#### Wireframes del administrador

Tabla

Descripción generada automáticamente

Ilustración Wireframe de la vista el inicio del usuario

Tabla

Descripción generada automáticamente

Ilustración Wireframe de la vista de las tablas

#### Mapa de navegación de la tienda

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

Ilustración Mapa de navegación de la tienda

#### Mapa de navegación del administrador

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración Mapa de navegación del administrador

## Diseño de la arquitectura

### Arquitectura del servidor

El desarrollo de la tienda se dividirá en 2 proyectos separados, uno para la parte del cliente, parte la cual contará con lo necesario para la vista del catálogo, del carrito, la compra de los productos, la administración de los datos del usuario logueado y un formulario de contacto. Esta parte estará desarrollada en JSP y Servlets.

Por otro lado, estará la parte de la administración de la tienda, esta parte contará con lo necesario para administrar los campos de la tienda, pudiendo añadir, editar, y dar de baja los datos que se quiera. Esta parte estará desarrollada usando el framework SpringBoot.

Ambas aplicaciones estarán desarrolladas siguiendo el patrón del modelo-vista-controlador, se relacionarán en que compartirán la misma base de datos.

Diagrama

Descripción generada automáticamente

Ilustración Estructura del proyecto

### Arquitectura del cliente

Del lado del cliente se hará uso de librerías de JavaScript para la paginación, los buscadores y los filtros presentes en las tablas de la aplicación de administración

# Detalles de implementación

## Desarrollo de la aplicación

Para el desarrollo de la aplicación se han usado distintas tecnologías y librerías

#### JSP y Servlets

JavaServer Pages es una tecnología que permite a los desarrolladores incrustar código Java dentro de páginas HTML. Proporciona una forma simplificada de crear contenido web dinámico combinando plantillas HTML estáticas con código Java en el lado del servidor.

Los servlets son clases de Java que se ejecutan en un servidor web y manejan las solicitudes entrantes de los clientes. Proporcionan un modelo de programación en el lado del servidor para construir aplicaciones web.

#### SpringBoot:

Spring Boot es un marco de desarrollo de aplicaciones Java que se basa en el framework Spring. Proporciona un enfoque simplificado y rápido para crear aplicaciones web y servicios RESTful, permitiendo a los desarrolladores centrarse en la lógica del negocio en lugar de ocuparse de la configuración y la infraestructura.

#### Lombok:

Esta librería se ha usado para la generación de los getters y los setters y los constructores de las clases. Esta librería genera esto usando las anotaciones @Data, @AllArgsConstructor y @NoArgsConstructor

Texto

Descripción generada automáticamente

Código Anotaciones de Lombok

#### Jasypt

Librería usada para la encriptación de las claves al guardar en la base de datos y desencriptación al comprobar los valores de las claves al iniciar sesión

Texto

Descripción generada automáticamente

Código Métodos de encriptación y desencriptación

#### IText

Esta librería se ha usado para la creación de las facturas del cliente, generándose algunas partes siguiendo un patrón en todas las facturas como es la cabecera y una parte dinámica según el pedido como es la parte relacionada con este, los datos del pedido y el detalle de cada producto en él.

Texto

Descripción generada automáticamente

Código Creación de la tabla de la factura

#### SpringBoot Mail

Esta librería de Springboot se ha usado para él envió de las facturas una vez se han generado, añadiendo un mensaje y la factura adjunta al correo que se va a enviar.

Texto

Descripción generada automáticamente

Código Creación del cuerpo del mensaje

#### JPA

Se ha usado JPA para las relaciones de la base de datos en los modelos de la aplicación del administrador.

Texto

Descripción generada automáticamente con confianza media

Código Código de las anotaciones de JPA

#### Simple DataTables

Esta librería de JavaScript se ha usado en la parte del administrador para la paginación, filtros y buscador de todas las tablas que se encuentran en el proyecto.



Código Código de las librerías de Simple Datatables

## Implementación real



# Fase de pruebas

## Pruebas

|  |  |
| --- | --- |
| N.º de prueba | 1 |
| Caso de uso a probar | Inicio de sesión |
| Prueba realizada | El usuario tiene que iniciar sesión |
| Situación inicial | No hay ningún usuario con la sesión iniciada |
| Resultado esperado | El usuario iniciará sesión y podrá realizar las funciones que le permiten su rol |
| Resultado obtenido | El usuario iniciará sesión y podrá realizar las funciones que le permiten su rol |

Tabla Prueba del inicio de sesión

|  |  |
| --- | --- |
| N.º de prueba | 2 |
| Caso de uso a probar | Registro de usuario |
| Prueba realizada | El usuario tiene que registrarse en la aplicación |
| Situación inicial | No hay ningún usuario con la sesión iniciada, el email que usara no está en la base de datos |
| Resultado esperado | El usuario se registrará en la aplicación |
| Resultado obtenido | El usuario se encuentra en la base de datos |

Tabla Prueba del registro de un usuario

|  |  |
| --- | --- |
| N.º de prueba | 3 |
| Caso de uso a probar | Editar Datos |
| Prueba realizada | El usuario tiene que editar sus datos |
| Situación inicial | Hay un usuario con la sesión iniciada |
| Resultado esperado | El usuario cambiara sus datos |
| Resultado obtenido | Las contraseñas no coinciden |

Tabla Prueba de la edición de datos de un usuario

|  |  |
| --- | --- |
| N.º de prueba | 4 |
| Caso de uso a probar | Cerrar sesión |
| Prueba realizada | El usuario tiene que cerrar sesión |
| Situación inicial | Hay un usuario con la sesión iniciada |
| Resultado esperado | El usuario cerrara sesión |
| Resultado obtenido | El usuario cierra su sesión |

Tabla Prueba del cierre de sesión

|  |  |
| --- | --- |
| N.º de prueba | 5 |
| Caso de uso a probar | Comprar |
| Prueba realizada | El usuario tiene que comprar en la aplicación |
| Situación inicial | Hay un cliente con la sesión iniciada |
| Resultado esperado | El usuario realizara el proceso de compra correctamente |
| Resultado obtenido | Fallo al controlar el stock del carrito |

Tabla Prueba de la compra

|  |  |
| --- | --- |
| N.º de prueba | 6 |
| Caso de uso a probar | Ver su historial |
| Prueba realizada | El usuario tiene que ver su historial |
| Situación inicial | Hay un cliente con la sesión iniciada y ha realizado compras |
| Resultado esperado | El usuario visualizara su historial de pedidos y su detalle |
| Resultado obtenido | El usuario visualiza el historial de pedidos |

Tabla Prueba de la consulta del historial

|  |  |
| --- | --- |
| N.º de prueba | 7 |
| Caso de uso a probar | Ver el carrito |
| Prueba realizada | El usuario tiene que ver su carrito |
| Situación inicial | Hay un cliente con la sesión iniciada y ha añadido al carrito productos |
| Resultado esperado | El usuario visualizara su carrito |
| Resultado obtenido | El usuario visualiza su carrito |

Tabla Prueba de la vista del carrito

|  |  |
| --- | --- |
| N.º de prueba | 8 |
| Caso de uso a probar | Realización de bajas o altas |
| Prueba realizada | El usuario tiene que dar de alta y de baja filas de las tablas |
| Situación inicial | Hay un administrador con la sesión iniciada |
| Resultado esperado | El usuario dará de baja o de alta correctamente |
| Resultado obtenido | El estado cambia |

Tabla Prueba de la realización de altas y bajas

|  |  |
| --- | --- |
| N.º de prueba | 9 |
| Caso de uso a probar | Añadir o editar datos |
| Prueba realizada | El usuario tiene que añadir datos nuevos y editar los presentes |
| Situación inicial | Hay un administrador con la sesión iniciada |
| Resultado esperado | El usuario añadirá datos y editará los presentes correctamente |
| Resultado obtenido | Error al añadir a la base de datos |

Tabla Prueba de añadir y editar datos

|  |  |
| --- | --- |
| N.º de prueba | 10 |
| Caso de uso a probar | Aceptar y cancelar pedidos |
| Prueba realizada | El usuario tiene que aceptar y cancelar pedidos |
| Situación inicial | Hay un administrador o empleado con la sesión iniciada |
| Resultado esperado | El usuario aceptara y cancelara productos correctamente |
| Resultado obtenido | Dirección de correo electrónico bloqueada |

Tabla Prueba de aceptar y cancelar pedidos

|  |  |
| --- | --- |
| N.º de prueba | 11 |
| Caso de uso a probar | Editar datos |
| Prueba realizada | El usuario tiene que editar datos ya presentes en la aplicación |
| Situación inicial | Hay un empleado con la sesión iniciada |
| Resultado esperado | El usuario editara correctamente |
| Resultado obtenido | Datos editados correctamente |

Tabla Prueba de editar datos como empleado

# Trabajo Futuro

En el futuro se planea añadir a la aplicación nuevas funciones como plataformas de pago externas como pueden ser PayPal o PaySafeCard.

También se planea desplegar la aplicación para que se pueda acceder desde distintos dispositivos

# Conclusiones

Considero que después de haber realizado este proyecto he aprendido bastante sobre java, Sprinboot, JSP y servlets, ThymeLeaf y aumentado mis conocimientos en el resto de las herramientas como pueden ser HTML, Bootstrap, css y JavaScript.

# Bibliografía

iText, P. D. F. (2018, noviembre 14). *IText 7 java*. IText PDF. <https://itextpdf.com/resources/api-documentation/itext-7-java>

*Java platform SE 8*. (s/f). Oracle.com. Recuperado el 28 de mayo de 2023, de https://docs.oracle.com/javase/8/docs/api/

*JavaMail*. (s/f). Github.io. Recuperado el 28 de mayo de 2023, de https://javaee.github.io/javamail/

Largo, E. [@elivarl]. (s/f). *Curso proyecto eCommerce: Proyecto spring framework, spring boot, servicios REST, thymeleaf, spring MVC, spring data JPA, spring security y deployment*. YouTube. Recuperado el 28 de mayo de 2023, de https://www.youtube.com/playlist?list=PL3vxkSlW2FvU9z7Gz\_Nn3E69HjEvv55\_G

*Thymeleaf*. (s/f). Thymeleaf.org. Recuperado el 28 de mayo de 2023, de https://www.thymeleaf.org/documentation.html

Todo, T. I. C. [@todoticpe]. (s/f). *Curso spring boot*. YouTube. Recuperado el 28 de mayo de 2023, de https://www.youtube.com/playlist?list=PLHVMU2rSmSIfOTfoIFUzeTQfm1zOTIWBc

Webb, P., Syer, D., Long, J., Nicoll, S., Winch, R., Wilkinson, A., Overdijk, M., Dupuis, C., Deleuze, S., Simons, M., Pavić, V., Bryant, J., Bhave, M., Meléndez, E., Frederick, S., & Halbritter, M. (s/f). *Spring boot reference documentation*. Spring.Io. Recuperado el 28 de mayo de 2023, de https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/