

DUOC UC - Escuela de informática y telecomunicaciones

Propuesta de Proyecto y Especificación de Requisitos de Software

Proyecto: [EcoMarket]

Revisión: [02]

[31-03-2025]

Contenido

FICHA DEL DOCUMENTO	3
1. INTRODUCCIÓN	4
1.1. PROPÓSITO	4
1.2. ÁMBITO DEL SISTEMA	4
1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	6
1.4. REFERENCIAS	8
1.5. VISIÓN GENERAL DEL DOCUMENTO	8
2. DESCRIPCIÓN GENERAL	9
2.1. PERSPECTIVA DEL PRODUCTO	9
2.2. FUNCIONES DEL PRODUCTO	9
2.3. CARACTERÍSTICAS DE LOS USUARIOS	13
2.4. RESTRICCIONES	14
2.5. SUPOSICIONES Y DEPENDENCIAS	15
2.6. REQUISITOS FUTUROS	15
3. REQUISITOS ESPECÍFICOS	16
3.1 REQUISITOS COMUNES DE LAS INTERFACES	16
3.1.1 <i>Interfaces de usuario</i>	16
3.1.2 <i>Interfaces de hardware</i>	17
3.1.3 <i>Interfaces de software</i>	17
3.1.4 <i>Interfaces de comunicación</i>	17
3.2 REQUISITOS FUNCIONALES	17
3.3 REQUISITOS NO FUNCIONALES	18
3.3.1 <i>Requisitos de rendimiento</i>	18
3.3.2 <i>Seguridad</i>	18
3.3.3 <i>Fiabilidad</i>	18
3.3.4 <i>Disponibilidad</i>	18
3.3.5 <i>Mantenibilidad</i>	18
3.3.6 <i>Portabilidad</i>	19
3.4 OTROS REQUISITOS	19
4. PROPUESTA DE PLANIFICACIÓN	19
4.1 DESCRIPCIÓN GENERAL ACERCA DE LA PLANIFICACIÓN	19
4.1.2 <i>Definición del Equipo de Trabajo</i>	19
4.1.3 <i>Definición de Actividades principales del Proyecto</i>	20

4.1.4 Diagrama EDT	21
4.1.5 Carta Gantt	22
4.1.6 Resumen Costos del Desarrollo del Proyecto	22
5. MODELO 4+1	24
5.1 VISTA LÓGICA	24
5.1.1 Diagrama de Clases	25
5.1.2 Diagrama de Secuencia	26
5.1.3 Diagrama de Comunicación	28
5.2 VISTA FÍSICA	29
5.2.1 Diagrama de Despliegue	29
5.3 VISTA DE PROCESOS	31
5.3.1 Diagrama de Actividad	31
5.4 VISTA DE ESCENARIOS	32
5.4.1 Diagrama de Actividad	32
6. PROPUESTAS DE PATRONES DE DISEÑO	33
6.1 PATRÓN DE DISEÑO	33
6.2 PATRÓN MODELO VISTA CONTROLADOR (MVC)	33
6.3 PATRÓN CLIENTE - SERVIDOR	33
7. ESTÁNDARES DE CALIDAD EN EL DISEÑO DE SOFTWARE	33

Ficha del documento

Fecha	Revisión	Autor	Modificación
31/03	N°1	Alonso Ravanal, Javier Quintana y Diego Muñoz	Comenzamos a hacer la carta Gantt, definiendo etapas y haciendo uso de sprints.
08/04	N°2	Javier Quintana y Diego Muñoz	Comenzamos a describir la planificación

Documento validado por las partes en fecha:

Integrantes:

Nombre Integrante del Equipo	Rol Definido
Javier Quintana	Gestor de proyecto, Analista
Alonso Ravanal	Diseñador
Diego Muñoz	Tester

1. Introducción

El presente documento es una Especificación de Requisitos de Software (ERS) para el proyecto llamado EcoMarket SPA, el cual es una empresa chilena dedicada a la venta de productos ecológicos y sostenibles que se encuentra en problemas de disponibilidad y rendimiento debido a que ha estado experimentando un crecimiento rápido y exponencial de clientes, lo que genera una sobrecarga en el sistema monolítico que utiliza la empresa. En este documento se hará una descripción general del producto y se enumeran los requisitos funcionales, no funcionales y los requisitos específicos.

1.1. Propósito

El objetivo de este documento es permitir al equipo de desarrollo tener conocimiento de manera clara sobre el sistema que el cliente desea, mostrando las necesidades que este debe satisfacer y qué se debe realizar para lograrlo, además de entregar al cliente conocimiento sobre el sistema que le será realizado. Este documento va dirigido al equipo de desarrollo y a los usuarios finales del sistema, quienes revisarán el contenido para señalar modificaciones en caso de que así lo requiera, las cuales se realizarán por medio de versiones sucesivas del documento, hasta lograr la total aprobación de todas las partes involucradas.

1.2. Ámbito del Sistema

Como se explicó anteriormente, la razón detrás de la realización de este sistema es la evidencia de problemas de rendimiento y disponibilidad en el sistema actual de EcoMarket SPA, los cuales han comenzado a evidenciarse tras el rápido crecimiento de clientes a nivel nacional.

El problema presentado, es producto del uso de un sistema de lenta velocidad, cuya arquitectura no permite la escalabilidad del software, ni las modificaciones de este sin tener que modificar el sistema completo. Es por este motivo que el sistema actual de EcoMarket SPA será reemplazado por un sistema de arquitectura de microservicios utilizando un motor de base de datos MySQL, sistema que separa los componentes del software en diferentes servicios y guarda la información en una base de datos, permitiendo la escalabilidad y la rápida actualización de estos sin afectar a los demás, siendo el sistema ideal para una tienda que se encuentra en crecimiento, como lo sería en este caso. El nuevo sistema se llamará "EcoMarket 2.0".

Se implementará la estrategia de microservicios de base de datos por microservicio, estrategia que como su nombre indica, consiste en crear bases de datos separadas por cada microservicio. Esta

estrategia es clave para el desarrollo de EcoMarket 2.0, ya que al separar las bases de datos, cada servicio funciona de manera totalmente independiente, lo cual permite que el resto del sistema continúe funcionando cuando un servicio se cae o presenta fallas, siendo este el motivo por el cual dicha estrategia es considerada más viable que otras existentes.

Para llevar a cabo el proyecto, se hará uso de diversos lenguajes de programación como JAVA, JavaScript y CSS y herramientas como Spring Boot, la cual es una extensión de Spring Framework para facilitar el desarrollo de microservicios y Bootstrap, framework que permitirá que el sistema funcione de manera responsiva. Se implementará además, Amazon Web Service (AWS) para hacer uso de la nube, utilizando Amazon Relational Database Service (RDS) para la eficiente y escalable gestión de la base de datos MySQL.

Se eligieron los frameworks Spring Boot y Bootstrap para el desarrollo del proyecto, por la facilidad al momento de implementar microservicios y su conectividad nativa con la base de datos MySQL que ofrece Spring Boot, y la responsividad que ofrece Bootstrap.

El sistema hará lo siguiente:

- Permitirá al administrador del sistema registrar y gestionar usuarios.
- Permitirá que el cliente pueda registrarse con sus datos personales, iniciar sesión y actualizar información de su perfil.
- Permitirá la gestión del inventario de productos para el usuario Gerente de tienda.
- Almacenará los datos ingresados en una base de datos MySQL.
- Permitirá que el cliente pueda navegar en el catálogo de productos, buscar productos, agregarlos al carrito, realizar el proceso de compra ingresando detalles de pago y opciones de envío y consultar historial de pedidos.
- Permitirá que el gerente de tienda pueda gestionar las tiendas y los pedidos de productos.
- Permitirá al administrador del sistema modificar y asignar permisos de acceso a diferentes módulos y funciones del sistema.
- Permitirá que el empleado de ventas pueda registrar ventas, consultar inventario de productos en tiempo real y generar facturas o boletas.
- Permitirá al empleado de ventas atender devoluciones y reclamos.
- Permitirá al usuario Logística gestionar envíos y proveedores, además de optimizar rutas de entrega y actualizar estado de pedidos.
- Permitirá al usuario Administrador del sistema su monitorización y realización de copias de seguridad de manera periódica para usarlas en caso de fallos.
- Generará reportes de ventas y permitirá configurar detalles específicos de la tienda.
- Permitirá que el cliente solicite soporte y pueda dejar reseñas y calificaciones.
- Permitirá al cliente aplicar cupones y descuentos.
- Permitirá al cliente elegir si desea boleta o factura luego de realizar una compra.

El sistema no hará lo siguiente:

- Contar con diferentes idiomas.
- Contar con modo oscuro.

EcoMarket 2.0 tendrá como beneficio la mejora de la experiencia de usuario y la eficiencia operativa, además del óptimo manejo del crecimiento exponencial de clientes sin tener problemas de rendimiento y la mejora de la disponibilidad. El objetivo de este nuevo sistema es implementar un sistema de microservicios que utilice la base de datos MySQL, de esta forma, reemplazando el actual sistema monolítico. Las metas que se esperan del nuevo sistema son que permita aceptar una cantidad creciente de usuarios sin presentar fallos, que sea una versión mejorada del sistema existente y mejorar la velocidad de las respuestas para mejorar efectivamente la experiencia de usuario.

1.3. Definiciones, Acrónimos y Abreviaturas

ERS: Especificación de requisitos de software.

IEEE 830: Estándar desarrollado por el Institute Electronic Electrical, Engineers que define formato y contenido de un documento ERS. Sugiere que la especificación de requisitos debe ser organizada y estructurada con descripciones sobre la funcionalidad del sistema, su comportamiento con sistemas externos y requisitos funcionales y no funcionales divididos por rendimiento, seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad.

RE: Requisito Específico.

RF: Requisito Funcional.

RNF: Requisito No Funcional.

Arquitectura de Microservicios: Es una arquitectura que permite dividir una aplicación o sistema en partes independientes.

BD: Base de datos.

RDBMS: Sistema de gestión de bases de datos relacionales.

MySQL: RDBMS de código abierto.

AWS: Amazon Web Service.

RDS: Relational Database Service.

SQL: Lenguaje de consulta estructurado.

API: Interfaz de programación de aplicaciones. Permiten la comunicación de dos componentes de software entre sí, mediante un conjunto de definiciones y protocolos.

-

API Rest: Es una API que se ajusta a los principios de diseño del estilo arquitectónico de transferencia de estado representacional. Brindará al proyecto una manera flexible y ligera de conectar componentes.

HTML: Es un lenguaje de marcado de hipertexto utilizado para el desarrollo de cada página web existente. Se utilizará en este sistema para estructurar los contenidos, de modo que el usuario pueda acceder a la información de manera sencilla.

CSS: Es uno de los lenguajes de programación que utilizaremos para definir el diseño de la página web del sistema. Define el estilo y presentación de un documento HTML.

JAVA: Lenguaje de programación usado para codificar aplicaciones web. Es uno de los lenguajes que se usará para el desarrollo del proyecto.

JavaScript: Lenguaje de programación que se utiliza para la realización de páginas web interactivas. Este lenguaje es uno de los que se usará para llevar a cabo este proyecto.

JPA: API de JAVA que indica cómo se debe realizar la persistencia de los objetos y simplifica el acceso a la base de datos de un sistema.

MFA: Medida de seguridad que requiere más de una verificación para iniciar sesión en una cuenta.

OAuth: Estándar que permitirá que el sitio web pueda acceder a recursos alojados por otras aplicaciones web en nombre de un usuario.

Spring Framework: Estructura de código abierto para crear aplicaciones independientes.

Spring Boot: Herramienta que facilita el desarrollo de aplicaciones web y microservicios con Spring Framework.

Bootstrap: Framework de código abierto que permite desarrollar sitios web responsivos con interfaces de usuario consistentes.

Usuario Cliente: El principal consumidor de la plataforma. Es quien compra los productos de EcoMarket SPA y deja reseñas.

Usuario Administrador del Sistema: Es quien se encarga de monitorizar el sistema, crear copias de seguridad, crear cada usuario y manejar sus roles.

Usuario Gerente de Tienda: Es quien se encarga de manejar los datos de los productos disponibles, hacer pedidos de reabastecimiento y los detalles de la tienda, además de generar reportes de rendimiento.

Usuario Empleado de Ventas: Es quien se encarga de registrar las ventas de la tienda en el sistema, además de generar boletas o facturas y atender devoluciones y reclamos.

-

Usuario Logística: Es quien se encarga de los pedidos de productos. Planifica y optimiza rutas de entrega, realiza seguimientos a los envíos y cambia su estado en el sistema.

1.4. Referencias

- [1] “Qué es HTML: el lenguaje de etiquetas”, *Universidad Europea*, 18-oct-2024. [En línea]. Disponible en: <https://universidadeuropea.com/blog/que-es-html/>. [Consultado: 06-abr-2025].
- [2] “¿Qué es Java Spring Boot?”, *Ibm.com*, 17-jul-2023.
- [3] “¿Qué es CSS?”, *Lenguajecss.com*. [En línea]. Disponible en: <https://lenguajecss.com/css/introduccion/que-es-css/>. [Consultado: 06-abr-2025].
- [4] Requirements Engineering, “¿Cuáles son los beneficios y desafíos de usar IEEE 830 para la ingeniería de requisitos?”, *Linkedin.com*, 03-abr-2023. [En línea]. Disponible en: <https://www.linkedin.com/advice/3/what-benefits-challenges-using-ieee-830?lang=es>. [Consultado: 06-abr-2025].
- [5] *Amazon.com*. [En línea]. Disponible en: <https://aws.amazon.com/es/rds/>. [Consultado: 06-abr-2025].
- [6] Atlassian, “Comparación entre la arquitectura monolítica y la arquitectura de microservicios”, *Atlassian*. [En línea]. Disponible en: <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>. [Consultado: 06-mar-2025].

1.5. Visión General del Documento

Este documento ERS cuenta con 3 secciones, esta es la introducción, donde se explica el propósito del documento, el ámbito del sistema planteado y las definiciones, acrónimos y abreviaturas que se utilizarán a lo largo de este. La segunda sección ofrece una descripción de todos los factores que afectan al producto y a sus requerimientos, muestra cuál es la perspectiva del producto, sus funciones, las características de los usuarios, las restricciones, las suposiciones, dependencias y requisitos futuros. En la tercera y última sección, se explica detalladamente cuáles son los requisitos funcionales y no funcionales que deberá satisfacer el sistema.

2. Descripción General

El repentino aumento de clientes en EcoMarket SPA ha causado que su sistema comience a colapsar. El uso de un sistema de arquitectura monolítico que no permite la escalabilidad y la necesidad de recibir nuevos clientes ha sido un desafío en el que se ha visto envuelta la empresa. Es por esto que para mitigar dicho problema, EcoMarket SPA realizará una migración de la arquitectura de su sistema a una arquitectura de microservicios, arquitectura que permite la escalabilidad y divide los módulos del sistema en diferentes servicios, lo cual permitirá recibir nuevos clientes sin presentar fallas, mejorando así la experiencia de usuario y la eficiencia operativa.

El proyecto se considerará como una plataforma web que permitirá además de realizar compras, una gestión más eficiente de usuarios, tiendas, inventarios, ventas, pedidos y logística, formalizando así, el cumplimiento de todos los requisitos que serán mostrados en el resto del documento.

2.1. Perspectiva del Producto

Debido a que se trata de un sistema web, cuyo funcionamiento no depende de sistemas operativos, EcoMarket 2.0 estará disponible en cualquier dispositivo que cuente con un navegador web, sin importar su sistema operativo.

El futuro sistema tendrá interacción con la plataforma Webpay para permitir la realización de pagos en línea.

2.2. Funciones del Producto

El producto debe tener las siguientes funciones para el óptimo manejo del usuario Administrador del sistema de EcoMarket SPA:

- Gestionar usuarios.
- Configurar permisos de otros usuarios.
- Monitorizar el funcionamiento del sistema.
- Respalidar y restaurar datos.

El producto debe tener las siguientes funciones para el óptimo manejo del usuario Gerente de tienda de EcoMarket SPA:

- Gestionar el inventario de productos.
- Generar reportes de ventas, inventario y rendimiento de la tienda.
- Gestionar tiendas.
- Gestionar pedidos.

-

El producto debe tener las siguientes funciones para el óptimo manejo del usuario Empleado de ventas de EcoMarket SPA:

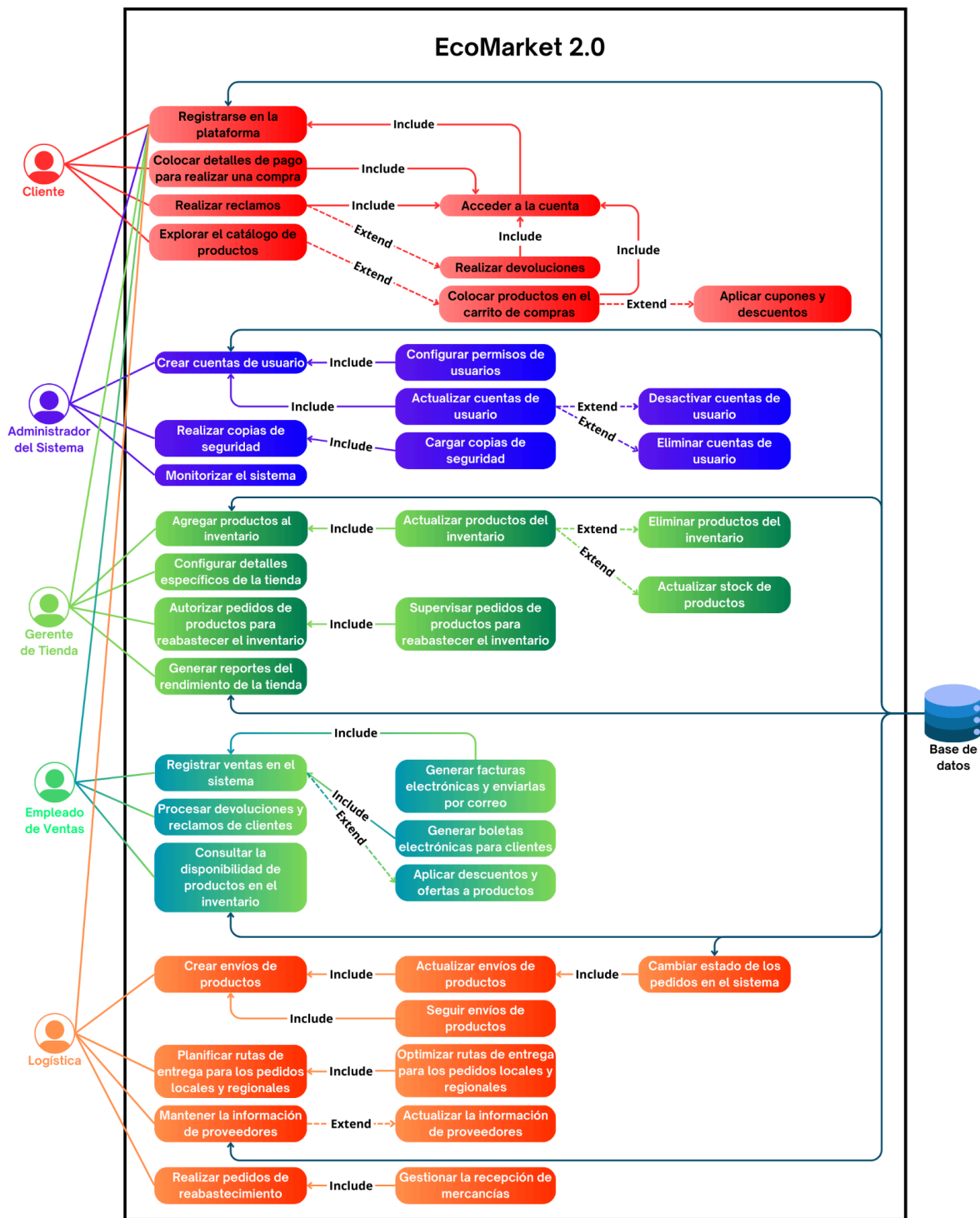
- Registrar las ventas realizadas en el sistema.
- Aplicar descuentos y ofertas a los productos que lo requieran.
- Atender las devoluciones y reclamaciones de los clientes.
- Consultar el inventario de los productos.
- Generar facturas que sean enviadas al correo electrónico de cada cliente.

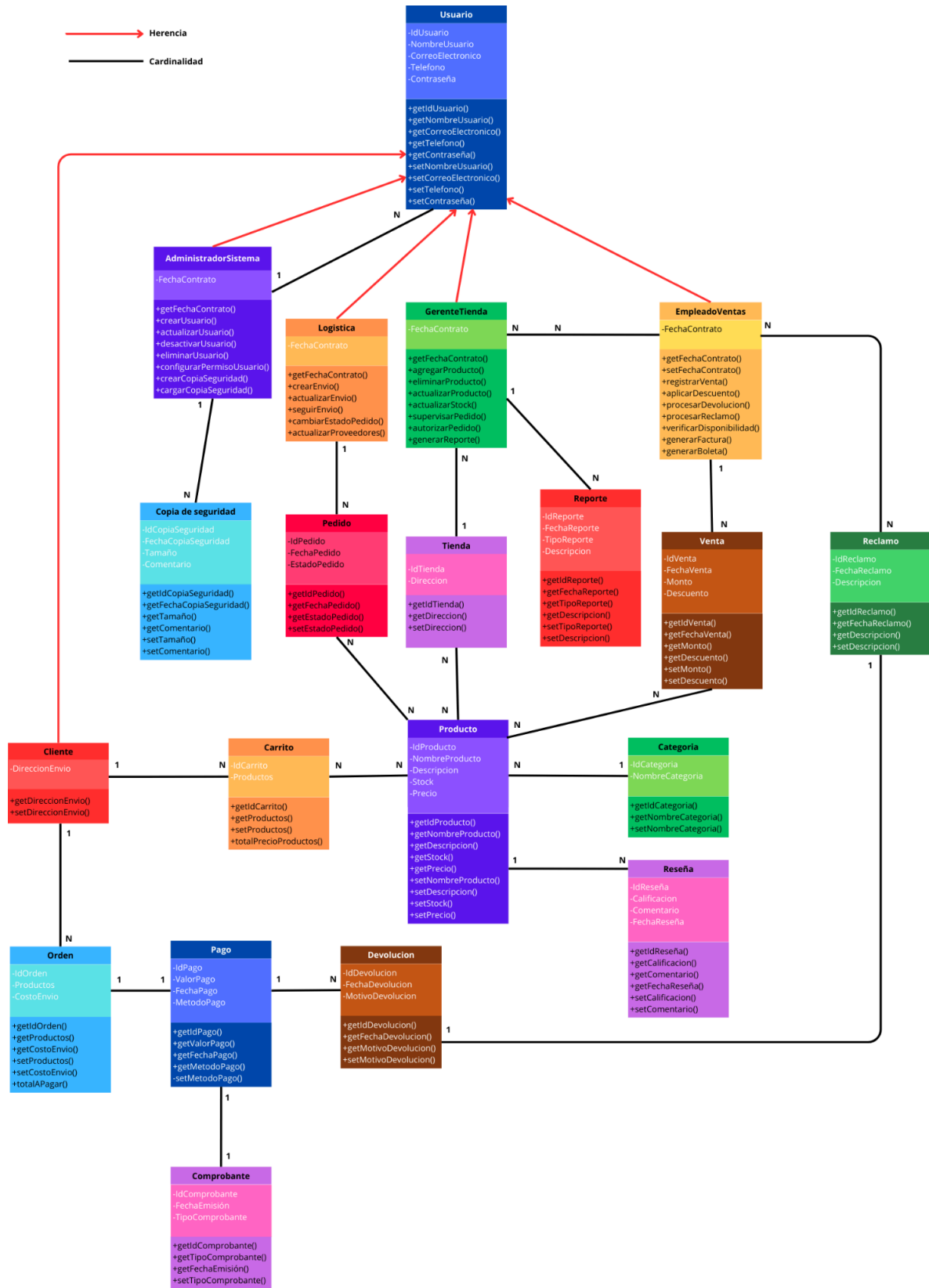
El producto debe tener las siguientes funciones para el óptimo manejo del usuario Logística de EcoMarket SPA:

- Gestionar envíos de productos desde la bodega hasta las tiendas y los clientes.
- Optimizar las rutas de entrega de los pedidos.
- Actualizar el estado de cada pedido en el sistema.
- Gestionar proveedores.

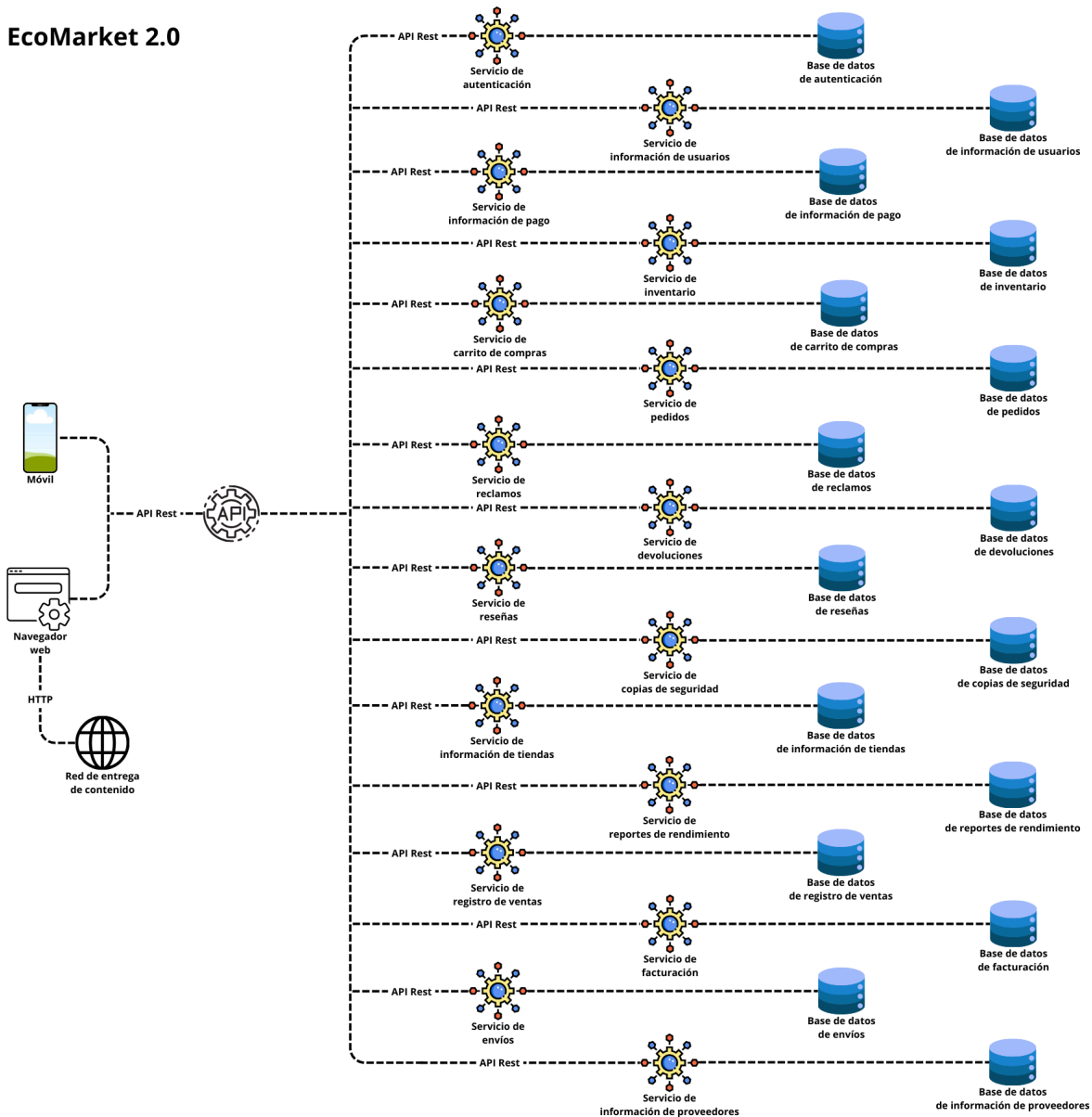
El producto debe tener las siguientes funciones para el óptimo manejo del usuario Cliente de EcoMarket SPA:

- Crear cuenta utilizando datos personales como nombre completo, dirección de correo electrónico, contraseña y dirección de envío.
- Iniciar sesión colocando su correo electrónico y contraseña.
- Navegar por el catálogo de productos y poder buscar uno en específico.
- Agregar los productos que desea comprar al carrito de compras.
- Comprar productos proporcionando detalles de pago y dirección de envío.
- Consultar historial de pedidos.
- Gestionar perfil: Cambiar datos personales como la contraseña, el correo electrónico, el número de teléfono y la dirección de envío.
- Enviar consultas o problemas a través de un formulario de contacto o chat en línea.
- Dejar reseñas y calificaciones.
- Aplicar cupones y descuentos.





EcoMarket 2.0



Este diagrama de arquitectura de microservicios representa la estructura que tendrá el sistema, mostrando todos los microservicios que este tendrá y su interacción con el sistema. El sistema hará uso del protocolo HTTP y tendrá una red de entrega de contenido para acelerar la carga de la página. Cada microservicio tendrá su propia base de datos y se conectarán usando API Rest.

2.3. Características de los Usuarios

Los usuarios administradores que se encargarán de verificar el funcionamiento del sistema y de gestionar al resto de los usuarios y sus permisos, son personas especializadas en el manejo de programas informáticos, las cuales obtuvieron el título de Ingeniería en Informática y tienen pleno conocimiento sobre las acciones que realizan en el sistema.

Los gerentes de tienda, quienes se encargarán de gestionar el inventario de productos, generar reportes y gestionar tiendas y pedidos, son personas con capacidad de liderazgo y habilidades de gestión de equipo, las cuales obtuvieron el título de Administración de empresas.

Los empleados de ventas, quienes se encargarán de registrar ventas, consultar el stock disponible, generar facturas y atender devoluciones y reclamaciones de los clientes, son personas que poseen habilidades de comunicación, las cuales completaron su enseñanza media.

Los usuarios de logística, quienes se encargarán de optimizar las rutas de entrega y gestionar estados de pedidos, envíos y proveedores, son personas que tienen experiencia en transporte, planificación y distribución, las cuales obtuvieron el título de Ingeniería en Gestión Logística.

Los clientes, quienes son parte vital para el funcionamiento de este sistema, son personas mayores de 18 años que pueden tener cualquier nivel educacional, y no necesariamente tener experiencia en el uso de este tipo de sistemas.

2.4. Restricciones

- Políticas de la empresa: El sistema debe adaptarse a las políticas internas de la empresa EcoMarket SPA, garantizando una correcta gestión de las funciones.
- Limitaciones del hardware: Los usuarios deberán contar con conexión a internet y un navegador web actualizado.
- Interfaces con otras aplicaciones: El sistema deberá conectarse con la plataforma Webpay para realizar pagos en línea.
- Operaciones paralelas: El sistema hará uso de Spring Boot y Bootstrap para realizar operaciones paralelas y optimizar la ejecución de tareas en segundo plano.
- Funciones de auditoría: Es de suma importancia que todos los cambios generados al sistema se guarden en logs para que los administradores puedan tener control sobre estos.
- Funciones de control: El sistema debe tener control sobre los permisos y roles de cada usuario y concederle acceso solamente a la información correspondiente a su tipo.

El software debe enviar un correo al usuario cada vez que hayan múltiples intentos fallidos de inicio de sesión para alertar sobre los intentos de inicio de sesión.

- Lenguajes de programación: El sistema será programado con JAVA, JavaScript y CSS.
- Protocolos de comunicación: El sistema tendrá protocolo HTTP para el acceso a internet y red.

Para la comunicación entre los microservicios, el sistema contará con API Rest.

- Requisitos de habilidad: Se deberá realizar una capacitación para el personal de EcoMarket SPA para que aprendan a usar la plataforma.
- Criticalidad de la aplicación: Es necesario realizar pruebas para verificar el funcionamiento del sistema y que se adapte a los requerimientos planteados.
- Consideraciones acerca de la seguridad: Las contraseñas e información de pago de cada usuario deben estar en una base de datos encriptada.

El sistema deberá contar con Autenticación Multifactor (MFA).

El sistema deberá contar con Autorización Abierta (OAUTH).

2.5. Suposiciones y Dependencias

Se asume que el personal de EcoMarket SPA recibirá una capacitación acerca del uso del sistema.

Se asume que el sistema podrá funcionar en cualquier dispositivo que cuente con un navegador web.

Se asume que todos los requisitos mencionados en este documento comenzarán a ser considerados como válidos en el momento en que sean aprobados por los usuarios finales y el equipo de EcoMarket SPA.

EcoMarket SPA cuenta con los siguientes tipos de usuarios: cliente, administrador del sistema, gerente de tienda, empleado de ventas y logística.

El sistema EcoMarket 2.0 depende totalmente de una conexión a internet para su correcto funcionamiento, por lo que de no contar con dicho requerimiento, no será posible ningún tipo de operación y no podría haber comunicación con las bases de datos ni con el usuario.

2.6. Requisitos Futuros

- Implementar un asistente de inteligencia artificial que le sugiera productos al usuario Cliente o le ayude a encontrar un producto en específico.
- Agregar diferentes idiomas.
- Permitir que el usuario pueda personalizar la pantalla del sistema.
- Implementar un historial de productos vistos por el usuario Cliente.
- Implementar lista de deseos para el usuario Cliente.

3. Requisitos Específicos

Mediante una reunión con el equipo de EcoMarket SPA, se llegó al acuerdo de crear un sistema web que permita la compra a distancia de los productos de la empresa y la gestión de inventario, tiendas, ventas, pedidos y logística. Un administrador podrá además gestionar a cada usuario registrado en la plataforma.

El cliente, conocido hoy como una empresa de buena reputación y con una cantidad de clientes que va en ascenso, desea la realización de este sistema como medida para erradicar las fallas que ha provocado su actual sistema, el cual está presentando problemas de disponibilidad y fallas de rendimiento debido al aumento de la demanda y la poca escalabilidad.

Es por este motivo que los requisitos específicos de este proyecto son:

- RE-001: Migración a un sistema de arquitectura de microservicios.
- RE-002: Evitar los problemas de disponibilidad y las caídas.
- RE-003: Mejorar la experiencia de usuario.

3.1 Requisitos comunes de las interfaces

El diseño de la página será responsivo y se adaptará a la pantalla de cada dispositivo en el que se ejecute.

Lo primero que se le debe mostrar a cada usuario es la pantalla de inicio de sesión que consiste en una casilla para ingresar el correo electrónico o el nombre de usuario y otra para ingresar la contraseña, además de los botones correspondientes para iniciar sesión, crear una cuenta en caso de que lo requiera y cambiar contraseña.

3.1.1 Interfaces de usuario

La interfaz del usuario Cliente será intuitiva y amigable con colores verde y blanco. Será de rápido entendimiento, donde verá iconos relacionados a cada opción (buscar, categorías, carrito de compras y configuración de cuenta) y una pantalla con los productos disponibles y su precio en formato grilla.

La interfaz del Administrador del Sistema seguirá con la idea de ser intuitiva y amigable con diferentes íconos relacionados a cada opción (buscar usuario, rendimiento tanto de las sucursales como del sistema mismo, gestionar usuarios, copias de seguridad, verificar funcionamiento del sistema) y una pantalla con gráficos del rendimiento mensual de cada tienda y del sistema.

La interfaz del Gerente de Tienda de igual manera será intuitiva y amigable con diferentes íconos (agregar producto, buscar producto, pedidos de productos, generar reportes y ver información sobre las tiendas) y una pantalla con los productos disponibles y su stock en formato de grilla. Al seleccionar cualquier producto, podrá actualizar su stock, modificar el producto o eliminarlo.

La interfaz del Empleado de Ventas tendrá diferentes íconos (productos, facturas y boletas, devoluciones y reclamos y ventas) y una pantalla similar a la del usuario Gerente de Tienda, mostrando los productos y su stock.

La interfaz del usuario Logística tendrá diferentes íconos (envíos, rutas, pedidos y proveedores) y una pantalla con los envíos pendientes en formato de lista, mostrando el producto, el usuario que realizó la compra y la fecha de compra.

3.1.2 Interfaces de hardware

El sistema deberá mostrar la información a través de la pantalla del dispositivo en que se está usando. Al tratarse de un sistema web, no se hace uso de ningún otro hardware.

3.1.3 Interfaces de software

EcoMarket 2.0 necesitará ser integrado en diversos productos de software.

MySQL: Sistema de gestión de Bases de datos relacionales que se encargará de almacenar datos como el inventario de los productos, los datos de los usuarios registrados, las transacciones realizadas y los pedidos realizados, permitiendo realizar operaciones CRUD en el sistema mediante tablas estructuradas con SQL. MySQL se conectará al sistema haciendo uso de JPA.

Webpay: Plataforma para el proceso de pago con tarjetas de crédito o débito. EcoMarket 2.0 se conectará a esta plataforma cuando el cliente vaya a realizar una compra en el sistema. El contenido de Webpay es la moneda, el monto, el tipo de pago y el ID del usuario en formato JSON.

3.1.4 Interfaces de comunicación

La comunicación entre JAVA y MySQL se realizará mediante JPA y la comunicación entre cada microservicio se realizará mediante API Rest con protocolo HTTP.

3.2 Requisitos funcionales

3.2.1 Cliente

- RF-001: El cliente debe poder registrarse en la plataforma web proporcionando la información necesaria. Para llevar a cabo el proceso de registro, se necesitan los siguientes datos: RUT, nombre, apellidos, dirección de correo electrónico, contraseña y dirección de envío.
- RF-002: El cliente debe poder explorar el catálogo de productos utilizando filtros y la barra de búsqueda para encontrar un producto específico.
- RF-003: El cliente debe poder realizar una compra usando detalles de pago.

3.2.2 Administrador del Sistema

- RF-004: El administrador del sistema deberá poder crear, actualizar, desactivar y eliminar cuentas de usuario.
- RF-005: El administrador del sistema deberá poder configurar los permisos de los usuarios.

- RF-006: El administrador del sistema deberá poder monitorizar el sistema para saber si está funcionando correctamente.

3.2.3 Gerente de Tienda

- RF-007: El gerente de tienda deberá poder agregar, actualizar y eliminar productos del inventario, además de actualizar el stock disponible.

3.2.4 Empleado de ventas

- RF-008: El empleado de ventas deberá poder registrar las ventas, aplicando descuentos y ofertas cuando se necesite.
- RF-009: El empleado de ventas deberá poder verificar la disponibilidad de los productos.

3.2.5 Logística

- RF-010: El usuario de logística deberá poder crear, actualizar y seguir envíos de productos desde la bodega, hasta el destino.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- RNF-001: El catálogo de productos cargará en menos de 1 segundo en el 95% de ocasiones.
- RNF-002: El cliente deberá poder realizar una compra rápidamente en un 92% de casos.

3.3.2 Seguridad

- RNF-003: El registro deberá ser de forma segura, con Autenticación Multifactor (MFA) y encriptación de datos.
- RNF-004: El módulo que guarda los datos de los usuarios deberá ser separado y tener permisos restringidos.

3.3.3 Fiabilidad

- RNF-005: El sistema deberá soportar al menos 500 clientes entre las 09:00 y las 12:00 horas sin presentar problemas, con una duración aproximada de 5 minutos por cada proceso de compra realizado.

3.3.4 Disponibilidad

- RNF-006: El sistema deberá estar en funcionamiento las 24 horas del día, los 7 días de la semana en un 99% de casos.

3.3.5 Mantenibilidad

- RNF-007: El sistema deberá tener como tipo de mantenimiento necesario la monitorización de este.

3.3.6 Portabilidad

- RNF-008: El sistema deberá funcionar al menos en Windows, MacOS, Linux, Android e IOS, esto debido a que son los sistemas operativos más usados.

3.4 Otros Requisitos

- RNF-009: El personal de EcoMarket SPA debe reubicar a un empleado en caso de que se vea afectado por la automatización de sus tareas.
- RNF-010: El personal de EcoMarket SPA debe realizar una capacitación a todos los empleados para que aprendan a usar el sistema.

4. Propuesta de Planificación

4.1 Descripción general acerca de la Planificación

El desarrollo del sistema se abordará durante un periodo de 119 días estimados, usando la metodología Scrum con sprints de entre 2 y 4 semanas de duración. El proyecto contará con un gestor de proyecto, un analista, un diseñador y un tester que se encargará de evaluar la calidad y el funcionamiento del sistema antes de su lanzamiento. Las buenas prácticas que se considerarán serán la integración continua y despliegue continuo para entregar software de alta calidad tras la finalización de cada sprint, y la realización de pruebas de penetración y encriptación de extremo a extremo para verificar la seguridad del sistema.

4.1.2 Definición del Equipo de Trabajo

El equipo de trabajo definido para EcoMarket 2.0 está formado por un Gestor de proyecto, un Analista, un Diseñador y un Tester. El trabajo será organizado en sprints con reuniones diarias para hacer seguimiento al proyecto y evaluar mejoras.

Rol	Función
Gestor de proyecto	Es quien planifica, dirige y controla el proyecto en todas las etapas.
Analista	Es quien se comunica con el cliente para clarificar requisitos y documentarlos.
Diseñador	Es quien se encarga del diseño y desarrollo del sistema. Él diseñará las interfaces, creará las bases de datos e implementará la infraestructura AWS.
Tester	Es quien evalúa la calidad y el funcionamiento del sistema antes de su lanzamiento.

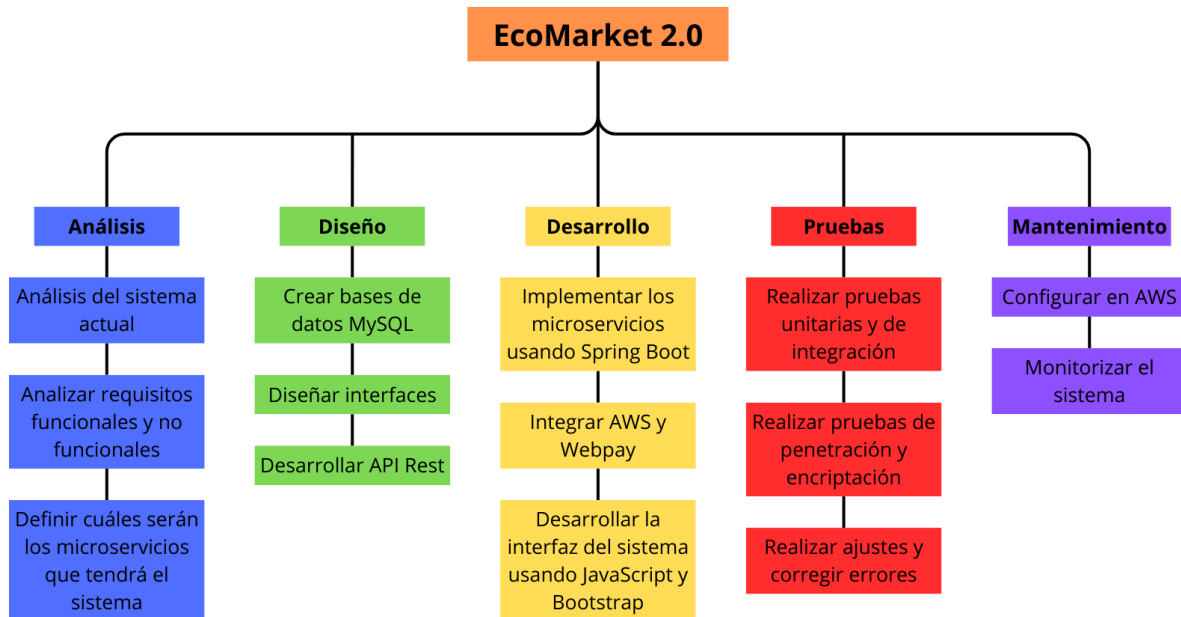
4.1.3 Definición de Actividades principales del Proyecto

Nuestra programación de planificación consiste en 5 etapas del ciclo de vida de software (Análisis, diseño, desarrollo, pruebas y mantenimiento) las cuales se basan en las buenas prácticas de la Gestión de la planificación PMI e Ingeniería de Software. Cada etapa del proyecto se encuentra estructurada bajo la metodología Scrum mediante sprints para garantizar la agilidad.

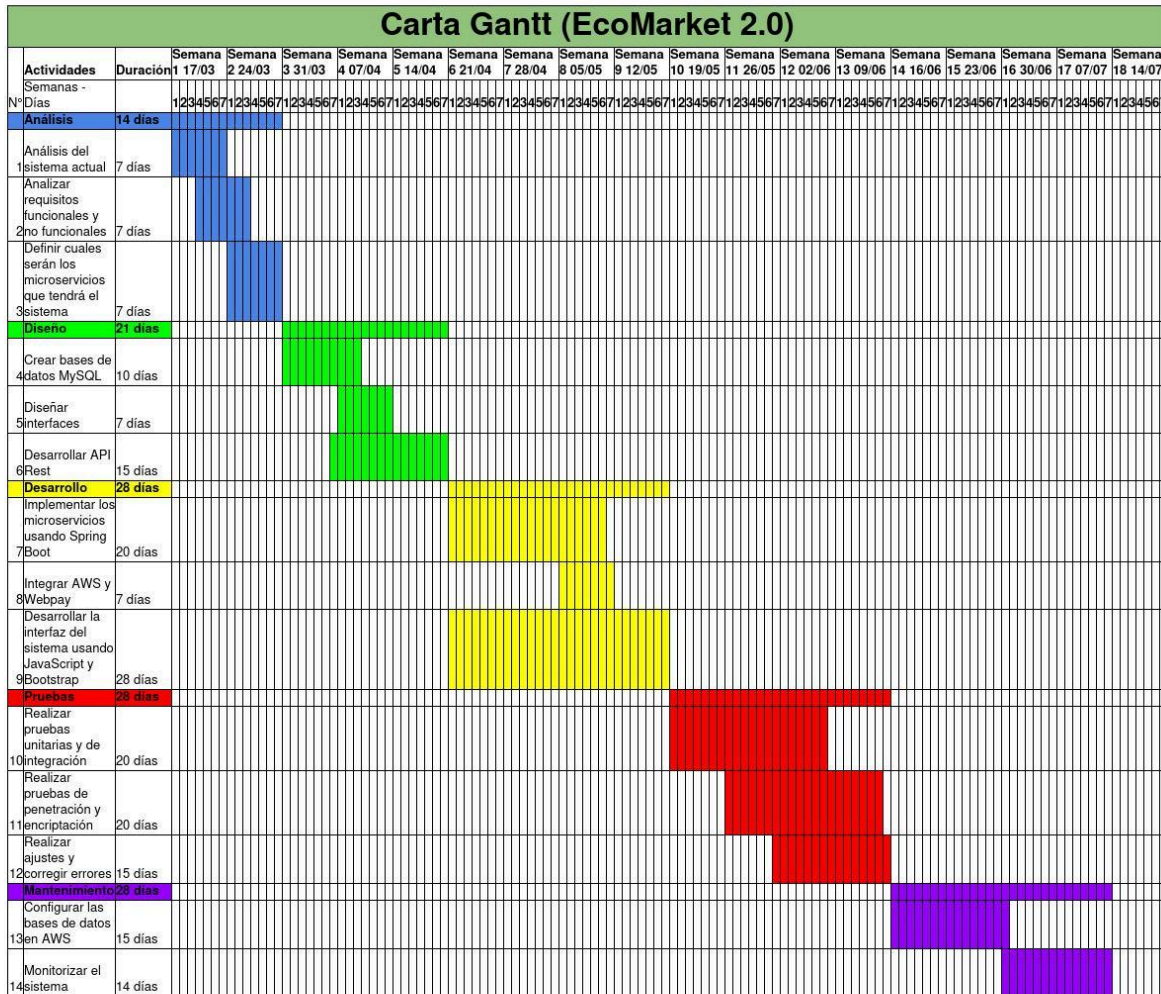
Fase	Costo
Análisis	Entre \$500.000 y \$1.500.000
Diseño	Entre \$25.000 y \$45.000
Desarrollo	Entre \$500.000 y \$2.000.000
Pruebas	Entre \$1.000.000 y \$4.000.000
Mantenimiento	Entre \$2.000.000 y \$4.500.000

Rol	Costo
Gestor de proyecto	Entre \$2.000.000 y \$3.500.000
Analista	Entre \$1.000.000 y \$1.500.000
Diseñador	Entre \$500.000 y \$1.500.000
Tester	Entre \$1.000.000 y \$2.000.000

4.1.4 Diagrama EDT



4.1.5 Carta Gantt



La presente carta Gantt corresponde a un periodo de 119 días (17 semanas) para la realización del proyecto de EcoMarket 2.0. Para reducir el total de días lineales, se usó la lógica de desarrollo paralelo, donde cosas como la realización de frontend y backend se realizan al mismo tiempo. Además, se utilizó sprints para cada etapa porque de esta forma, habría más eficiencia, evitando retrasos en el proyecto al impedir la presencia de tiempos muertos.

4.1.6 Resumen Costos del Desarrollo del Proyecto

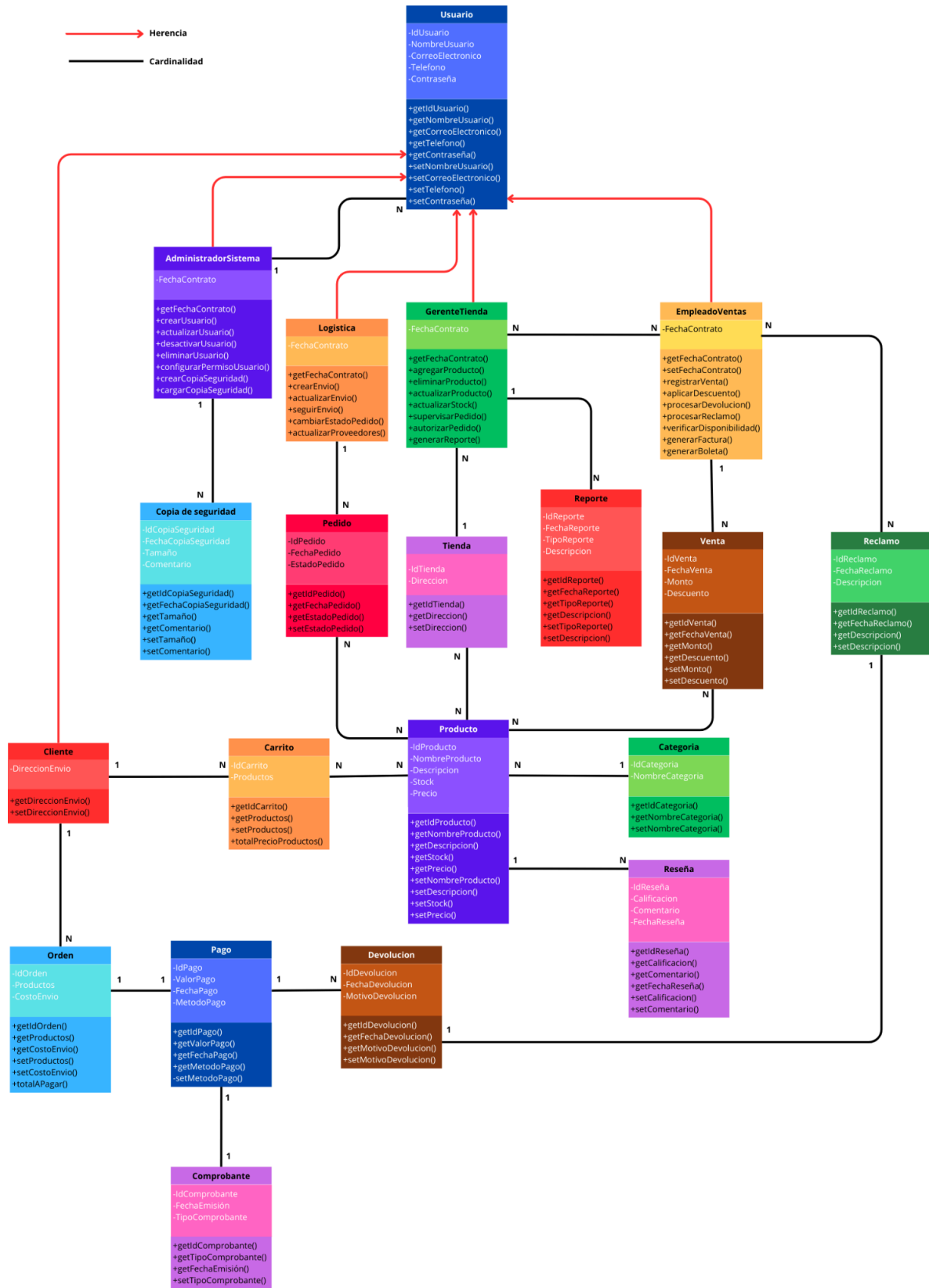
Rol	Costo total base esfuerzo hora hombre	Nombre fase	Costo por fase
Gestor de proyecto	\$4.200.000	Análisis	\$1.000.000

	\$6.300.000	Diseño	\$2.000.000
	\$11.200.000	Desarrollo	\$3.000.000
	\$8.400.000	Pruebas	\$1.000.000
	\$8.400.000	Mantenimiento	\$1.000.000
Analista	\$4.900.000	Análisis	\$1.500.000
	\$7.350.000	Diseño	\$3.000.000
	\$9.800.000	Desarrollo	\$5.000.000
	\$4.900.000	Pruebas	\$1.500.000
	\$9.800.000	Mantenimiento	\$1.500.000
Diseñador	\$3.500.000	Análisis	\$1.000.000
	\$5.250.000	Diseño	\$2.500.000
	\$7.000.000	Desarrollo	\$5.000.000
	\$5.600.000	Pruebas	\$3.000.000
	\$4.200.000	Mantenimiento	\$1.500.000
Tester	\$4.200.000	Análisis	\$1.500.000
	\$6.300.000	Diseño	\$2.000.000
	\$8.400.000	Desarrollo	\$6.000.000
	\$7.000.000	Pruebas	\$3.500.000
	\$7.000.000	Mantenimiento	\$3.000.000

5. Modelo 4+1

5.1 Vista lógica

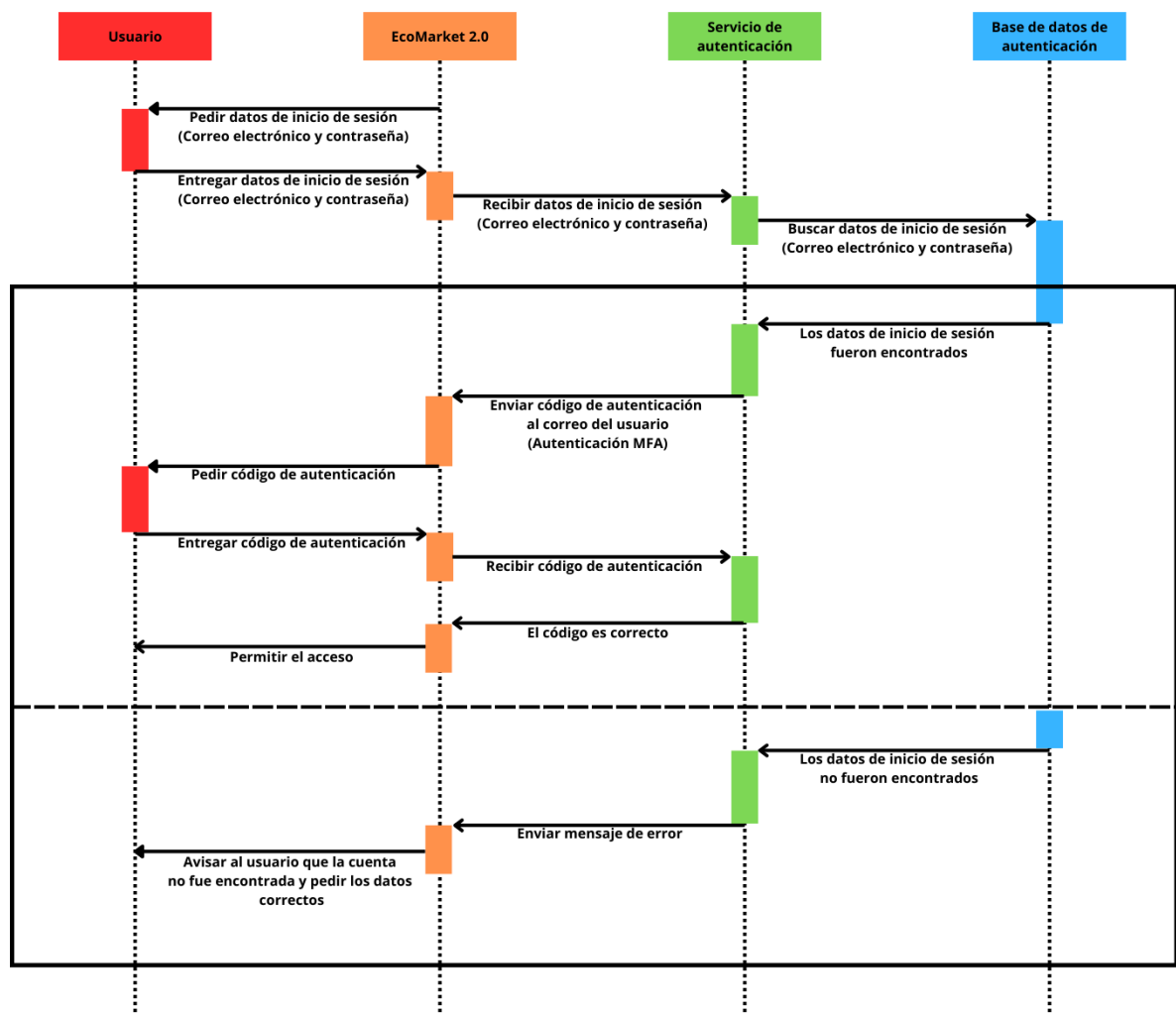
5.1.1 Diagrama de clases



Este diagrama de clases representa la estructura que tendrá el sistema en un enfoque orientado a objetos, mostrando las clases, los atributos y su relación. Los tipos de usuarios que estarán presentes son mostrados como diferentes entidades que heredan de una clase llamada “Usuario” y poseen su respectiva cardinalidad.

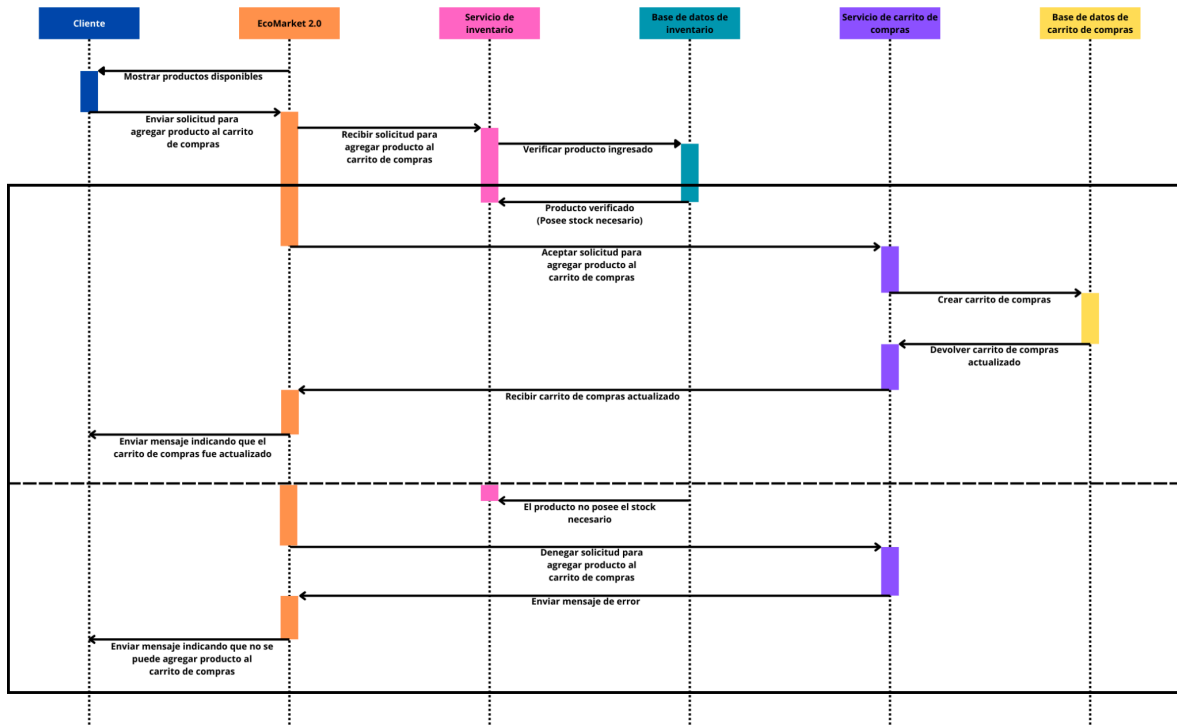
5.1.2 Diagrama de secuencia

EJEMPLO 1: Inicio de sesión



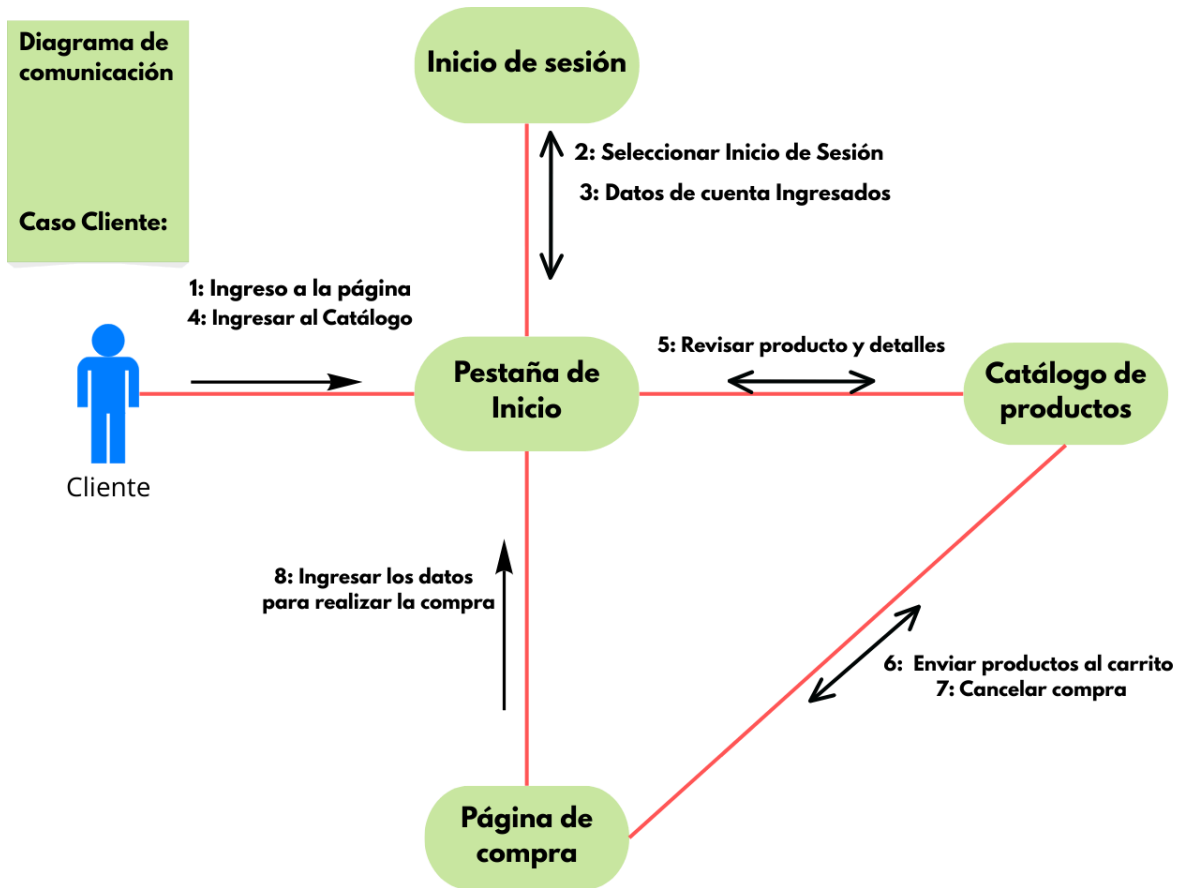
Este ejemplo de diagrama de secuencia muestra los objetos que van a interactuar al momento de iniciar sesión en la plataforma, mostrando también lo que pasaría si no se encuentran los datos ingresados por el usuario. Esto es relevante porque se podrá saber el comportamiento que debería tener el sistema en ambas situaciones.

EJEMPLO 2: Carrito de compras

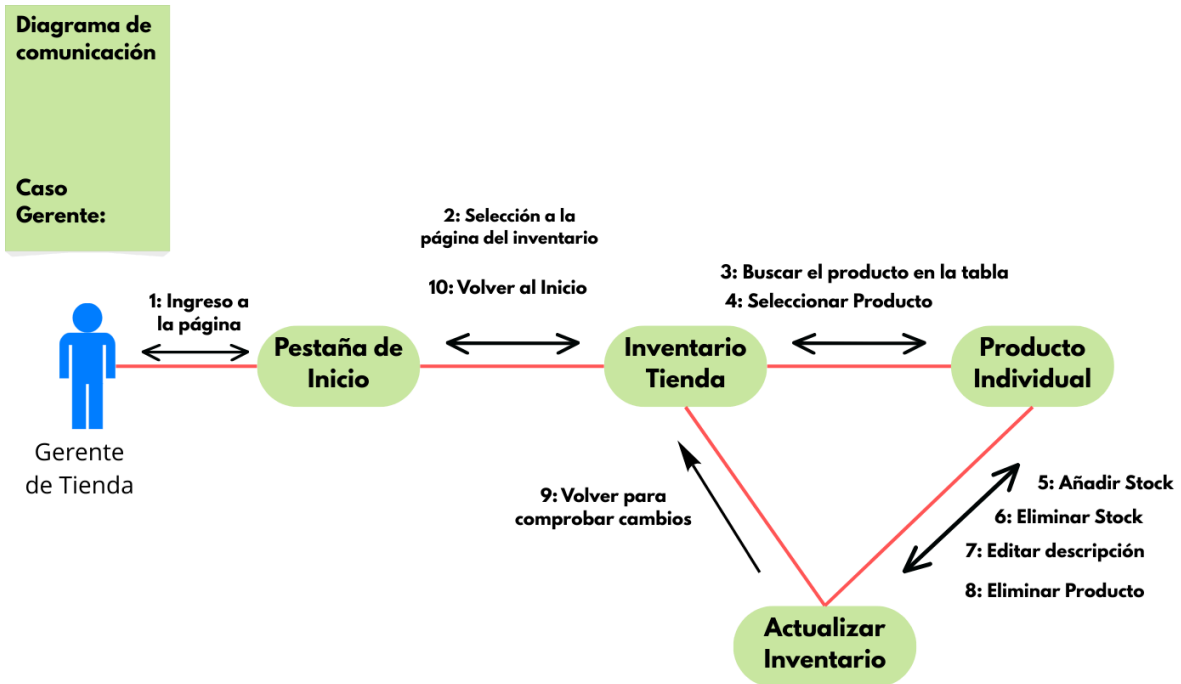


El segundo diagrama de secuencia muestra cómo será la interacción de los objetos al momento de que un usuario Cliente agregue un producto al carrito de compras, demostrando además que para que este pueda agregar el producto, debe haber stock disponible del mismo y cómo debe proceder el sistema en ese caso.

5.1.3 Diagrama de comunicación



Este ejemplo de diagrama de comunicación muestra los mensajes que serán enviados entre el sistema y el usuario Cliente al momento de realizar una compra desde el momento en que accede a la página. Mediante opciones, el usuario Cliente envía mensajes al sistema para pedirle que le muestre la pestaña de inicio, los detalles de cada producto, su catálogo, el acceso a la página de compra y el ingreso de datos.

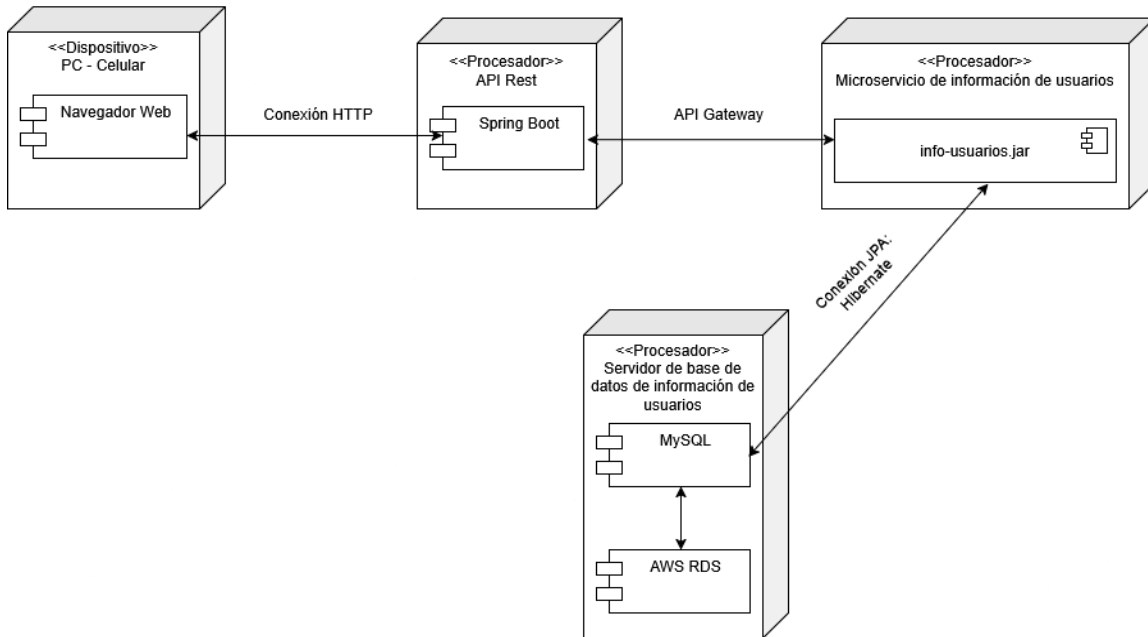


Este segundo ejemplo de diagrama de comunicación muestra los mensajes que enviará el Gerente de Tienda al sistema para que le permita editar los datos de un producto en el inventario y cómo el sistema le responderá de vuelta, permitiendo que pueda realizar dicha acción.

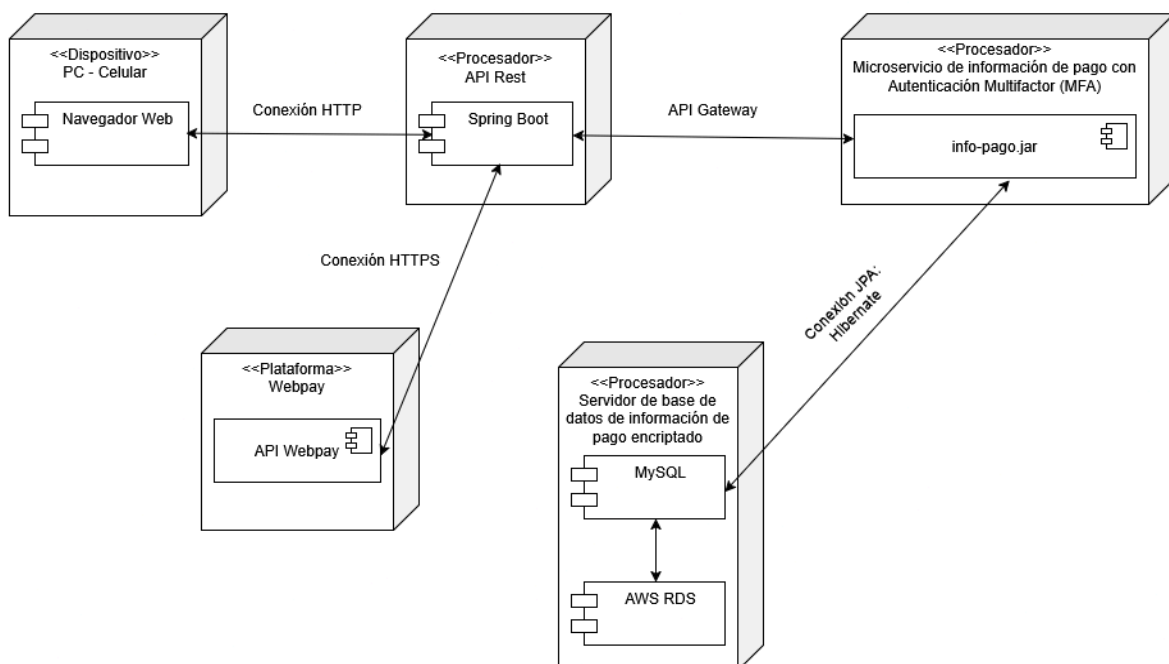
5.2 Vista física

5.2.1 Diagrama de despliegue

EJEMPLO 1: Información de usuarios



EJEMPLO 2: Pago

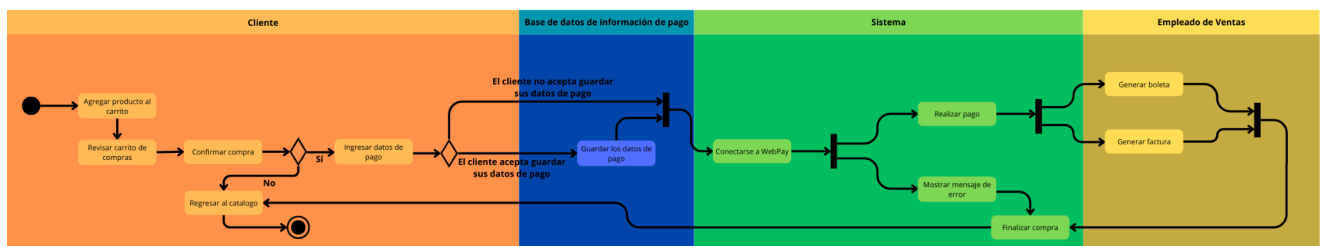


Estos ejemplos de diagramas de despliegue representan como será la comunicación que tendrá el sistema y los microservicios mostrando sus componentes y artefactos. El primer ejemplo muestra que la comunicación con el usuario con el servicio de información de usuarios será con conexión HTTP, la comunicación de la API Rest al microservicio será con API Gateway y la comunicación del microservicio a su base de datos correspondiente será con JPA, haciendo uso del proveedor Hibernate. Por otro lado, el segundo ejemplo muestra la comunicación con el sistema a la plataforma Webpay haciendo uso de la API Rest y una conexión HTTPS.

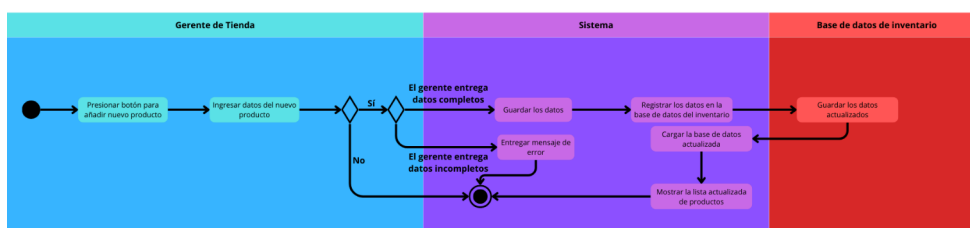
5.3 Vista de procesos

5.3.1 Diagrama de actividad

EJEMPLO 1: Actividad de pago



EJEMPLO 2: Actividad de agregar producto al inventario



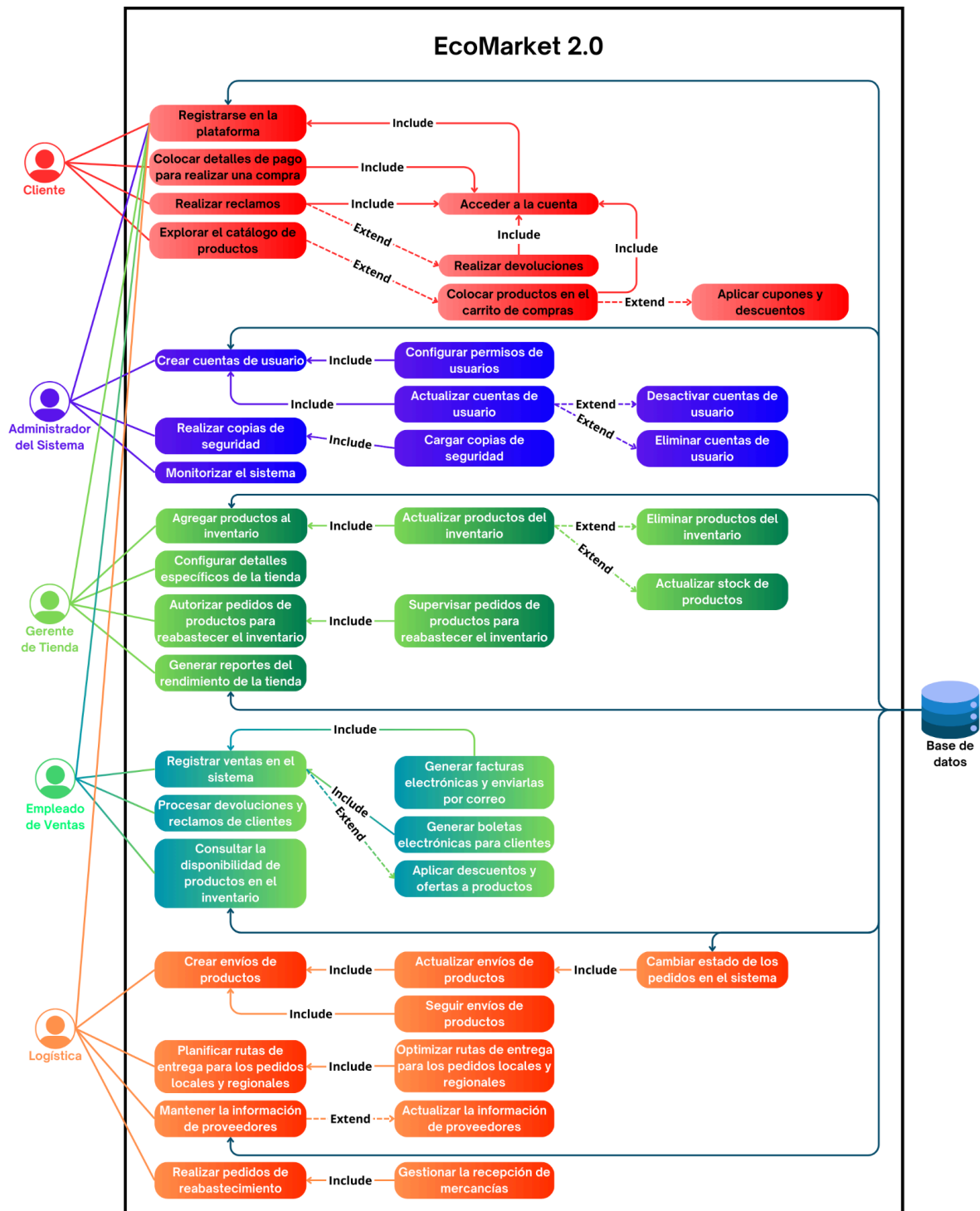
El primer ejemplo de diagrama de actividad representa cómo será el flujo del sistema cuando el cliente realice una compra desde el momento en que agrega productos al carrito de compras hasta el momento de finalizar la compra.

El segundo ejemplo representa el flujo del sistema cuando el usuario Gerente de Tienda agregue productos nuevos al inventario, considerando lo que pasaría si el usuario coloca datos incompletos u opta por cancelar el proceso.

Ambos diagramas son relevantes porque permitirán entender la relación de cada evento del caso de uso.

5.4 Vista de escenarios

5.4.1 Diagrama de casos de uso



-

El diagrama de casos de uso representa la interacción que tendrá cada usuario y las bases de datos en la plataforma, mostrando lo que podrá hacer cada uno y la información que se guardará en las bases de datos.

6. Propuestas de patrones de diseño

6.1 Patrón de dominio

El patrón de dominio es un patrón adecuado para EcoMarket 2.0 porque permite la organización de las entidades del sistema, facilitando el mantenimiento y las actualizaciones. El sistema deberá permitir la navegación web haciendo uso de dominios para acceder a los datos.

6.2 Patrón Modelo Vista Controlador (MVC)

Es adecuado para el sistema de EcoMarket 2.0 agregar un patrón de diseño de Modelo Vista Controlador (MVC), ya que al separar los componentes, se genera una mejor organización. Las entidades del negocio, como por ejemplo, Producto, Usuario, Inventario, etc, estarían ubicadas en Modelo, las interfaces en Vista y los controllers en Controlador. Aplicar este patrón facilita la escalabilidad del sistema y la abstracción del código.

6.2 Patrón Cliente - Servidor

Es relevante usar este patrón en el sistema por el hecho de que al ser una plataforma web, los usuarios deberán conectarse haciendo uso de HTTP para interactuar con un servidor realizado con Spring Boot, el cuál tendrá API Rest, cuya función es responder las solicitudes de los usuarios. El uso de este patrón es fundamental para lograr que el sistema logre ejecutarse en diversos dispositivos.

7. Estándares de calidad en el diseño del software

En este proyecto, se entiende como estándares de calidad una serie de técnicas que permitirán que el sistema sea de calidad, cumpla con su objetivo y sea claro en cuanto a diseño se refiere. Es por eso que en este proyecto se optó por el estándar de calidad ISO/IEC/IEEE 42010, el cual define lineamientos para el diseño de software en el contexto de su entorno. El uso de este estándar permite determinar el comportamiento de EcoMarket 2.0 usando diagramas UML y el modelo 4+1, el cual explica de manera visual el funcionamiento desde diferentes perspectivas y ayuda a organizar los diferentes diagramas realizados. Esto genera una comprensión contundente del sistema en que se está trabajando y su funcionamiento, tanto a nivel de conexión como a nivel de toma de decisiones, además de permitir identificar problemas antes de que ocurran.

8. Plan de pruebas

8.1 Introducción plan de pruebas

Este plan de pruebas se diseñó para definir cuáles pruebas serán realizadas en el proyecto de EcoMarket 2.0 y cómo se harán, además de los riesgos y suposiciones.

8.2 Recursos

Tester	Porcentaje de participación
Diego Muñoz	100%

8.3 Alcance

Las pruebas incluyen las pruebas unitarias de los componentes Service del proyecto de EcoMarket 2.0, las pruebas de integración de los componentes RestController del proyecto y las pruebas de interfaz de usuario simuladas.

8.4 Fuera del alcance

Por el momento, solo se tomarán en cuenta las pruebas unitarias, de integración y de interfaz de usuario simuladas, por lo que otras pruebas no serán parte de este plan de pruebas.

8.5 Pruebas unitarias

Las pruebas unitarias serán realizadas usando los frameworks JUnit y Mockito. Se harán simulaciones del comportamiento de los métodos.

8.6 Pruebas de integración

Las pruebas de integración para los componentes RestController serán realizadas usando los frameworks JUnit y Mockito, además de MockMVC. Se harán simulaciones a las peticiones HTTP.

8.7 Pruebas de interfaz de usuario simuladas

Las pruebas de interfaz de usuario simuladas se realizarán en Figma, la cual es la herramienta que usamos para crear el mockup, y en la cual se evaluará que se muestren pantallas en distintos casos.

8.8 Infraestructura

El dispositivo que será usado para realizar las pruebas debe contar con todos los programas y frameworks necesarios para realizar las pruebas.

8.9 Riesgos

N°	Riesgos	Probabilidad (1 - 5)	Impacto (1 - 5)	Severidad (Probabilidad * Impacto)	Plan de mitigación
1	Los microservicios no están completamente implementados.	3	5	15	Reunirse con el equipo de desarrollo y evaluar lo que falta.
2	Las pruebas tienen errores.	2	5	10	Hacer una revisión del código para encontrar el error y corregirlo.
3	Errores en las anotaciones de Mockito o JUnit	4	5	20	Revisar meticulosamente que las anotaciones incluidas en las clases sean las correctas.

9. Planilla de casos de pruebas

N°	Nombre identificador	Descripción	Pre condiciones	Entradas	Pasos	Resultados esperados	Pos condición	Estado	Prioridad
1	CP_Reg1	Registrarse a la plataforma colocando todos los datos.	Entrar a la plataforma y hacer clic en "Registrarse"	RUT, nombre, apellidos, número de teléfono, correo electrónico, contraseña y dirección de envío.	1- Entrar a la plataforma. 2- Ir a "Registrarse" 3- Llenar todos los campos. 4- Hacer clic en "Registrarse".	Mostrará una pantalla para iniciar sesión con los datos de la cuenta y el texto "¡Se ha registrado correctamente!".	Se registra el usuario en la plataforma.	Abierto	ALTA

2	CP_Ini1	Iniciar sesión colocando todos los datos.	Estar registrado en la plataforma.	Correo electrónico y clave.	1- Entrar a la plataforma. 2- Ir a "Iniciar sesión". 3- Colocar todos los datos. 4- Hacer clic en "Iniciar sesión".	Se mostrará la pantalla principal.	El usuario accede a la cuenta.	Abierto	ALTA
3	CP_Car1	Acceder al carrito de compras sin tener productos agregados.	No agregar ningún producto.	Ninguna	1- Entrar a la plataforma. 2- Iniciar sesión. 3- Ir a "Carrito".	Se mostrará una pantalla con el texto "¡No tienes productos en el carrito de compras!" y un botón para regresar a la pantalla principal.	Se mantiene todo igual, no hay cambios.	Abierto	ALTA
4	CP_Car2	Acceder al carrito de compras con un producto agregado.	Agregar un producto al carrito de compras.	El usuario hace clic en "Agregar al carrito" al ver los detalles del producto.	1- Entrar a la plataforma. 2- Iniciar sesión. 3- Buscar un producto. 4- Seleccionar producto. 5- Hacer clic en "Agregar al carrito" 6- Regresar a la pantalla principal. 7- Hacer clic en "Carrito".	Se muestra el nombre del producto y su precio, además del precio del envío y el total en la pantalla del carrito de compras, junto con un botón para realizar la compra de todos los productos y otro botón para eliminar un producto.	Se agrega el producto al carrito.	Abierto	ALTA

5	CP_Car3	Eliminar producto del carrito de compras.	Acceder al carrito de compras.	El usuario hace clic en "Eliminar un producto del carrito" y luego hace clic en uno de los símbolos para eliminar ubicados a un costado de la información de un producto.	1- Entrar a la plataforma. 2- Iniciar sesión. 3- Agregar un producto en el carrito. 4- Ir a "Carrito". 5- Hacer clic en "Eliminar un producto". 6- Hacer clic en uno de los símbolos para eliminar ubicados a un costado de la información de un producto.	Se muestra una pantalla que dice "Producto eliminado correctamente" junto con un botón para deshacer la acción y otro botón para volver a la página principal.	Se elimina el producto del carrito.	Abierto	ALTA
---	---------	---	--------------------------------	---	---	--	-------------------------------------	---------	------

10. Registro de control de cambios

Versión	Cambios
1.0.0	<ul style="list-style-type: none"> - Se dió comienzo a la creación del mockup con las páginas de registro, inicio de sesión y cambio de contraseña. - Se realizó el flujo de compra para el cliente.
1.1.0	<ul style="list-style-type: none"> - Se realizó el flujo para añadir y eliminar productos del carrito de compras.
1.2.0	<ul style="list-style-type: none"> - Se agregaron las pantallas de inicio, usuarios, rendimiento y perfil para el usuario Administrador del Sistema. - Se añadieron las funcionalidades para

	eliminar y desactivar usuarios para el usuario Administrador del Sistema.
1.3.0	<ul style="list-style-type: none"> - Se agregaron las pantallas de inicio, ventas y perfil para el usuario Empleado de Ventas. - Se añadieron las funcionalidades para registrar ventas para el usuario Empleado de Ventas.
1.3.1	<ul style="list-style-type: none"> - Corrección de errores en las pantallas de Empleado de Ventas. En el menú inferior, en el botón de ventas se mostraba el ícono de usuarios del usuario Administrador del Sistema. Se cambió por un carro.
1.4.0	<ul style="list-style-type: none"> - Se agregaron las pantallas de inicio, envíos y perfil para el usuario Logística. - Se añadieron las funcionalidades para crear y actualizar envíos para el usuario Logística.