

e-commerce casa europea de repuestos

Yonner Hazziel Duvan Suárez Huertas

Asesor

Ing. Javier Hernán Jiménez Beltrán

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias básicas, tecnología e Ingeniería

Tecnología en Desarrollo de Software

2025

Resumen

El proyecto de E-commerce Casa Europea de Repuestos aborda las limitaciones en el modelo de ventas tradicional de la empresa, incluyendo la ausencia de un canal digital y las dificultades en la gestión de inventarios y procesos manuales, lo cual impacta negativamente su competitividad. El Objetivo General del trabajo fue desarrollar un sistema de gestión de ventas en línea enfocado en la optimización de la experiencia del cliente.

Para ello, se utilizó una metodología de investigación aplicada con enfoque mixto, complementada con el marco ágil Scrum para estructurar el ciclo de desarrollo. El sistema fue diseñado con una arquitectura basada en microservicios, lo que garantiza escalabilidad, modularidad y facilidad de mantenimiento. Las tecnologías principales empleadas son React para el frontend, y .NET y MySQL para el backend.

La plataforma integra módulos esenciales como catálogo de productos, plataforma de pagos, carrito de compras, gestión de pedidos y administración de inventarios, todos ellos orientados a una interfaz intuitiva (UX). El proceso incluyó la identificación y documentación de requerimientos funcionales y no funcionales. Se ejecutaron pruebas funcionales, de seguridad y de usabilidad para validar la estabilidad y confiabilidad del software. Aunque el sistema aún está pendiente de aprobación empresarial, fue desplegado y validado en un entorno de pruebas (QA). Esta propuesta se consolida como una herramienta estratégica que fortalece la competitividad de Casa Europea de Repuestos, proyectándola hacia un modelo de negocio digital sostenible.

Palabras clave: Comercio Electrónico, Microservicios, Scrum, React, .NET, Inventario.

Abstract

The Casa Europea de Repuestos E-commerce project addresses limitations in the company's traditional sales model, including the absence of a digital channel and difficulties in manual inventory management and processes, which negatively affect its competitiveness. The General Objective of the work was to develop an online sales management system focused on optimizing the customer experience.

The project utilized an applied research methodology with a mixed approach, complemented by the agile framework Scrum to structure the development lifecycle. The system was designed with a microservices architecture, which ensures scalability, modularity, and ease of maintenance. The primary technologies employed include React for the frontend, and .NET, and MySQL for the backend and data management.

The platform integrates essential modules such as product catalog, payment processing, shopping cart, order management, and inventory administration, all geared towards an intuitive User Experience (UX). The process involved identifying and documenting both functional and non-functional requirements. Functional, security, and usability tests were executed to validate the software's stability and reliability. Although the system is still pending corporate approval, it was successfully deployed and validated in a Quality Assurance (QA) testing environment. This proposal establishes itself as a strategic tool that strengthens the competitiveness of Casa Europea de Repuestos, moving it toward a sustainable digital business model.

Keywords: E-commerce, Microservices, Scrum, React, .NET, Inventory.

Tabla de Contenido

Introducción	11
Presentación Del Proyecto	13
Problema de Investigación	13
Objetivos.....	14
Objetivo General	14
Objetivos Específicos.....	14
Marco Conceptual y Teórico.....	15
Comercio Electrónico	15
<i>Conceptos y Características del Comercio Electrónico</i>	<i>15</i>
<i>Arquitectura Técnica del Comercio Electrónico</i>	<i>17</i>
<i>Procesos de Compra en el Comercio Electrónico.....</i>	<i>19</i>
<i>Seguridad y Confianza Dentro del Contexto de Comercio Electrónico</i>	<i>20</i>
<i>Marketing Digital</i>	<i>21</i>
<i>Innovación y Tendencias en el Comercio Electrónico</i>	<i>23</i>
Marco Legal y Normativo	24
<i>Legislación Nacional sobre Comercio Electrónico.....</i>	<i>24</i>
<i>Normas Internacionales Aplicables</i>	<i>26</i>
<i>Aspectos Fiscales y Tributarios</i>	<i>27</i>
<i>Seguridad en Pagos y Protección de Datos.....</i>	<i>28</i>
Casos de Estudio en Comercio Electrónico	30
<i>Ejemplos Nacionales.....</i>	<i>30</i>
<i>Análisis Comparativo</i>	<i>32</i>
Lecciones Aprendidas para el Proyecto.....	33
Fundamentos Teóricos de la Gestión en comercio Electrónico.....	34
<i>Teoría de la Cadena de Suministro</i>	<i>34</i>
<i>Teoría del Comportamiento del Consumidor en Entornos Digitales</i>	<i>36</i>
<i>Principios de Gestión de Inventarios</i>	<i>38</i>
Conceptos Clave para el Éxito del Comercio Electrónico	40
<i>Confianza y Seguridad en el Comercio Electrónico</i>	<i>40</i>

<i>Experiencia de Usuario (UX) como Factor Diferencial</i>	42
Lenguajes, Técnicas, Metodologías y Patrones de Diseño	43
<i>Lenguajes de Programación</i>	43
<i>Técnicas de Desarrollo</i>	45
Metodologías de Desarrollo	47
Patrones de Diseño	49
Diseño Metodológico	52
Tipo de Investigación	52
Características de la Investigación Mixta Aplicada al Proyecto	52
Variables	53
Técnicas e Instrumentos de Recolección	54
Metodología Scrum	54
<i>El Equipo de Desarrollo</i>	55
<i>Scrum Master</i>	56
<i>El Product Backlog</i>	56
<i>Estimaciones Afinadas</i>	56
<i>Los Sprints</i>	57
<i>Reunión Diaria (Stand-Up)</i>	57
<i>La Revision del Sprint (Sprint Review)</i>	57
<i>La Retrospectiva (Sprint Retrospective)</i>	58
Cronograma de Actividades	58
Requerimientos del Sistema	60
Requerimientos funcionales	62
REQ-F01	62
REQ-F02	63
REQ-F03	64
REQ-F04	64
REQ-F05	65
REQ-F06	66
REQ-F07	67
REQ-F08	68

Requerimiento No Funcionales	69
<i>REQ-NF01</i>	69
<i>REQ-NF02</i>	70
<i>REQ-NF03</i>	71
<i>REQ-NF04</i>	72
Arquitectura del Sistema	75
<i>API Gateway (.NET)</i>	77
<i>Microservicios (.NET)</i>	77
<i>Base de datos (MySQL)</i>	78
Diseño del Sistema	79
Actores del Sistema	79
Casos de Uso	80
Casos de Uso General	81
Casos de Uso Individual	83
Diagrama de Actividades	88
Modelo Entidad Relación	89
MockUp	91
<i>Página Principal “Casa Europea de Repuestos”</i>	92
<i>Módulo Administrador “Casa Europea de Repuestos”</i>	95
<i>Catálogo Productos “Casa Europea de Repuestos”</i>	101
<i>Módulo Compra “Casa Europea de Repuestos”</i>	104
Código Fuente	109
<i>FrontEnd</i>	109
<i>Gateway (Ocelot)</i>	112
<i>Microservicio Usuarios</i>	113
<i>Microservicio Catalogo Productos</i>	116
<i>MicroServicio Inventario</i>	117
<i>Microservicio Pedidos</i>	119
<i>Microservicio Pagos</i>	121
Pruebas del Sistema	123
Pruebas Requerimientos Funcionales	124

<i>Caso de prueba 1: Registro de usuarios (REQ-F01)</i>	124
<i>Caso de prueba 2: Prueba de Autenticación de Administradores (REQ-F02)</i>	125
<i>Caso de prueba 3: Prueba de Gestión de Productos (REQ-F03)</i>	127
<i>Caso de prueba 4: Prueba de Carrito de Compras (REQ-F04)</i>	130
<i>Caso de prueba 5: Prueba de Procesamiento de Pagos (REQ-F05)</i>	132
Pruebas Requerimientos No Funcionales	134
<i>Caso de prueba 7: Prueba de Seguridad de la Información (REQ-NF02)</i>	134
<i>Caso de prueba 8: Prueba de Compatibilidad (REQ-NF04)</i>	136
Despliegue del Sistema	138
Requisitos de hardware	138
Requisitos de Software	141
Configuración de Entorno	145
Recomendaciones para despliegue	148
Manual de usuario	150
Trabajos Futuros	155
Accesibilidad	155
Internacionalización	156
Módulo de Analítica de Datos	156
Modulo Reportador	157
Conclusiones	159
Referencias Bibliografías	161

Lista de Figura

Figura 1 <i>Arquitectura del Sistema</i>	78
Figura 2 <i>Caso de Uso General</i>	82
Figura 3 <i>CU01 – Registro de Datos Usuarios</i>	83

Figura 4 <i>CU02 – Gestionar Catálogo</i>	85
Figura 5 <i>CU03 – Realizar Compras</i>	86
Figura 6 <i>CU04 – Gestión Pago con Bancos</i>	87
Figura 7 <i>CU05 – Gestionar Pedidos</i>	88
Figura 8 <i>Diagrama de Actividades</i>	89
Figura 9 <i>Modelo Entidad Relación</i>	90
Figura 10 <i>Página de Inicio Casa Europea de Repuestos</i>	93
Figura 11 <i>Modales Inicio de Sesión</i>	95
Figura 12 <i>Módulo Catálogo de Producto</i>	96
Figura 13 <i>Modal Creación de Producto</i>	97
Figura 14 <i>Módulo Usuarios</i>	98
Figura 15 <i>Creación de Usuario</i>	99
Figura 16 <i>Módulo Reporte de Pedidos</i>	100
Figura 17 <i>Módulo Catálogo de Productos</i>	102
Figura 18 <i>Detalles del Producto</i>	103
Figura 19 <i>Módulo de carrito del Sistema (Vacio)</i>	105
Figura 20 <i>Módulo de carrito del Sistema (con Productos)</i>	107
Figura 21 <i>Módulo de Pago del Sistema</i>	109
Figura 22 <i>Registro Usuarios</i>	125
Figura 23 <i>Ingreso Modulo Administrativo</i>	127
Figura 24 <i>Resultado de la prueba para el requerimiento identificado con REQ-F03</i>	129
Figura 25 <i>Resultado de la prueba para el requerimiento identificado con REQ-F04</i>	131
Figura 26 <i>Resultado de la prueba para el requerimiento identificado con REQ-F05</i>	133
Figura 27 <i>Resultado de la prueba para el requerimiento identificado con REQ-NF02</i> ...	135

Figura 28 <i>Resultado de la prueba para el requerimiento identificado con REQ-NF04...</i>	136
Figura 29 <i>Despliegue Front en Vercel.....</i>	152
Figura 30 <i>Despliegue de Micriservicios en Render.....</i>	153
Figura 31 <i>Despliegue de Base de Datos en Cloud Clever</i>	154

Lista de Tablas

Tabla 1 <i>Cronograma de Actividades del Proyecto</i>	58
Tabla 2 <i>Requerimiento Funcional 1</i>	62
Tabla 3 <i>Requerimiento Funcional 2</i>	63
Tabla 4 <i>Requerimiento Funcional 3</i>	64
Tabla 5 <i>Requerimiento Funcional 4</i>	65
Tabla 6 <i>Requerimiento Funcional 5</i>	66
Tabla 7 <i>Requerimiento Funcional 6</i>	67
Tabla 8 <i>Requerimiento Funcional 7</i>	68
Tabla 9 <i>Requerimiento Funcional 8</i>	69
Tabla 10 <i>Requerimiento no Funcional 1</i>	70
Tabla 11 <i>Requerimiento no Funcional 2</i>	71
Tabla 12 <i>Requerimiento no Funcional 3</i>	72
Tabla 13 <i>Requerimiento no Funcional 4</i>	73
Tabla 14 <i>Requerimiento no Funcional 5</i>	74
Tabla 15 <i>Actores de Sistema</i>	79
Tabla 16 <i>Arquitectura de la Plataforma en la Nube</i>	140
Tabla 17 <i>Inventario de Tecnologías y Herramientas Utilizadas en el Proyecto</i>	143

Introducción

En un contexto de transformación digital acelerada, las empresas se enfrentan al desafío de adaptar sus modelos de negocio a las nuevas dinámicas del mercado y a los hábitos de consumo contemporáneos. Casa Europea de Repuestos, dedicada a la comercialización de repuestos automotrices, presenta limitaciones en su proceso de ventas tradicionales, las cuales afectan su competitividad, la eficiencia en la gestión de inventarios y su capacidad para alcanzar nuevos segmentos de clientes. La ausencia de un canal digital, sumada a procesos manuales que incrementan la posibilidad de errores y reducen la capacidad de respuesta, evidencia la necesidad de implementar un sistema de comercio electrónico que optimice las operaciones, modernice la interacción con los usuarios y potencie el crecimiento del negocio.

La presente tesis desarrolla un sistema e-commerce integral que aborda tanto los requerimientos funcionales como los no funcionales, incorporando aspectos legales, técnicos y comerciales esenciales para garantizar su operación en un entorno confiable y seguro. Para ello, se analizan conceptos fundamentales del comercio electrónico, su arquitectura técnica, los procesos de compra, el marco legal colombiano aplicable, así como estudios de caso nacionales e internacionales que permiten identificar buenas prácticas y lecciones relevantes para el proyecto. Asimismo, se abordan teorías y modelos relacionados con la gestión en entornos digitales, como la cadena de suministro, el comportamiento del consumidor y los principios de gestión de inventarios, fundamentales para comprender la dinámica del comercio electrónico moderno.

El proyecto se desarrolla bajo una metodología de investigación aplicada con enfoque mixto, complementada con el uso del marco ágil Scrum, lo que permitió organizar y priorizar las actividades del ciclo de desarrollo. Se diseñó una arquitectura basada en microservicios con

tecnologías como React, .NET y MySQL, garantizando escalabilidad, modularidad y facilidad de mantenimiento. El sistema integra módulos esenciales como catálogo de productos, plataforma de pagos, carrito de compras, gestión de pedidos, administración e inventarios, acompañados de una interfaz intuitiva y orientada a la experiencia del usuario (UX).

Para validar la calidad y confiabilidad del software, se realizaron pruebas funcionales, de seguridad y de usabilidad que aseguraron el cumplimiento de los requerimientos definidos. Finalmente, se efectuó el despliegue en producción y la capacitación del personal del almacén, permitiendo la adopción efectiva del sistema y asegurando su correcta operación. Esta propuesta no solo representa una solución tecnológica, sino también una herramienta estratégica que fortalece la competitividad de Casa Europea de Repuestos y la proyecta hacia un modelo de negocio digital sostenible.

Presentación Del Proyecto

Problema de Investigación

En un entorno comercial cada vez más digitalizado, las empresas enfrentan la necesidad de adaptarse a nuevas formas de consumo para mantener su competitividad. Casa Europea de Repuestos, dedicada a la venta de repuestos automotrices, atraviesa dificultades para ampliar su presencia en el mercado a través de medios digitales. La empresa presenta limitaciones en su modelo de ventas tradicional, lo que impacta en su alcance comercial, eficiencia operativa y capacidad de respuesta ante la demanda actual.

Entre los principales problemas se identifican la falta de un canal digital para la comercialización de productos, dificultades en la gestión de inventario en tiempo real y procesos manuales. Estos factores afectan el crecimiento del negocio, reducen su competitividad frente a empresas que ya han adoptado estrategias digitales, y generan una brecha entre la empresa y los hábitos actuales de consumo. La situación exige una transformación en su modelo operativo para alinearse con las nuevas dinámicas del mercado.

Objetivos

Objetivo General

Desarrollar un sistema de gestión de ventas en línea (e-commerce), para la empresa Casa Europea de Repuestos enfocado en la experiencia del cliente.

Objetivos Específicos

Identificar y documentar los requerimientos funcionales y no funcionales del sistema e-commerce, incluyendo aspectos legales, técnicos y comerciales, alineados con las necesidades del cliente, el almacén y el usuario final.

Diseñar la arquitectura del sistema, la interfaz gráfica (UI) y experiencia de usuario (UX) contemplando módulos para catálogo de productos, carrito de compras, pagos en línea, gestión de pedidos, atención al cliente y administración de inventario.

Desarrollar los módulos funcionales del sistema e-commerce, incluyendo catálogo de productos, gestión administrativa, inventarios y plataforma de pagos segura.

Ejecutar pruebas funcionales, de seguridad y usabilidad para validar la integridad, estabilidad y confiabilidad del sistema en todos sus componentes.

Desplegar el sistema en un entorno de producción y capacitar al personal del almacén para asegurar su correcta operación y gestión.

Marco Conceptual y Teórico

Comercio Electrónico

Conceptos y Características del Comercio Electrónico

El comercio electrónico es el proceso de compra, venta o intercambio de bienes, servicios e información a través de Internet y otras redes informáticas, lo que ha transformado la manera en que se realizan los negocios a nivel global (Torres & Guerra, 2012). Este modelo elimina las barreras de tiempo y espacio, permitiendo que un consumidor adquiera un producto o servicio desde cualquier lugar del mundo con tan solo una conexión a la red. Asimismo, el e-commerce integra actividades diversas como la transmisión electrónica de datos, pagos en línea, suministro de contenidos digitales, subastas y transacciones entre empresas y consumidores (Schneider, 2004). Entre sus modalidades más relevantes se encuentran el B2B (Business to Business), que engloba las transacciones entre empresas; el B2C (Business to Consumer), dirigido al consumidor final; y el C2C (Consumer to Consumer), que facilita los intercambios entre particulares mediante plataformas digitales (Alonso, 2004).

De acuerdo con Malca (2001), el comercio electrónico usa las tecnologías de la informática y las telecomunicaciones para soportar transacciones de productos o servicios entre empresas, particulares o con el Estado. Esto implica que el e-commerce forma parte de un ciclo que incluye atraer al cliente mediante publicidad, interactuar con él respondiendo a sus necesidades, coordinar la ejecución de la venta y finalmente brindar servicios posventa que aseguren su lealtad.

Una característica fundamental del comercio electrónico es que se sostiene sobre una infraestructura tecnológica que involucra personas, políticas, estándares técnicos y protocolos. En este marco, intervienen no solo vendedores y compradores, sino también intermediarios, servicios de pago, marcos normativos y organismos reguladores que garantizan seguridad en las transacciones (Malca, 2001). La red de Internet, junto con tecnologías asociadas como intranets y extranets, constituye la base para la interacción entre empresas y consumidores, facilitando la integración de procesos de negocio y el acceso a mercados globales.

El comercio electrónico también se caracteriza por su impacto económico y organizacional. Ha transformado las relaciones entre empresas y clientes al permitir mayor eficiencia, reducción de costos y alcance internacional. Al mismo tiempo, exige una adaptación constante de las organizaciones a través de nuevas estrategias de marketing, gestión de datos de clientes y modelos de negocio digitales. El contacto directo, la disponibilidad permanente y la posibilidad de personalización son aspectos que incrementan la competitividad de las empresas y mejoran la experiencia de los consumidores (Seoane, 2005; Malca, 2001).

En síntesis, el comercio electrónico representa una evolución en la forma de comprar y vender y constituye un nuevo paradigma de intercambio económico y social, en el que confluyen la globalización, la digitalización de procesos y la innovación tecnológica. Sus características — interactividad, accesibilidad global, reducción de costos, rapidez en la transacción, disponibilidad 24/7 y personalización del servicio— lo convierten en una herramienta esencial para el desarrollo de los negocios en la economía digital contemporánea.

Arquitectura Técnica del Comercio Electrónico

La arquitectura técnica del comercio electrónico constituye el soporte tecnológico que hace posible la operación de los sistemas digitales de compraventa, garantizando que las transacciones se realicen de manera eficiente, segura y escalable. En su concepción más general, incluye los elementos de hardware, software, redes de comunicación, protocolos de seguridad y sistemas de gestión que permiten la interacción entre consumidores, empresas y entidades reguladoras. Como señalan Torres y Guerra (2012), esta arquitectura ha pasado de ser una infraestructura basada en páginas estáticas con catálogos limitados, a sistemas dinámicos e integrados que soportan miles de transacciones en tiempo real, con altos niveles de disponibilidad y personalización.

Desde una perspectiva organizacional, Malca (2001) explica que la arquitectura técnica del comercio electrónico no solo involucra los componentes informáticos, sino también a las personas, las políticas, los estándares técnicos y los marcos regulatorios que definen cómo se lleva a cabo el intercambio digital. Esto implica integrar sistemas de gestión de pedidos, pasarelas de pago electrónico, bases de datos para administrar información de clientes y proveedores, y protocolos de comunicación como HTTPS, XML o SOAP, que garantizan la interoperabilidad entre plataformas. La correcta coordinación de todos estos elementos es fundamental para asegurar la confianza en las transacciones.

En los últimos años, las arquitecturas tradicionales de tipo monolítico han sido reemplazadas progresivamente por modelos más flexibles y escalables, como la arquitectura de microservicios. Según Espinoza (2021), este enfoque consiste en dividir una aplicación en múltiples servicios pequeños, independientes y desplegados de manera autónoma, cada uno

encargado de una funcionalidad específica (como pagos, gestión de usuarios, inventarios o envíos). Estos microservicios se comunican entre sí a través de API REST, GraphQL o sistemas de mensajería asíncrona, lo que incrementa la capacidad de respuesta y reduce la dependencia de un único sistema centralizado.

El uso de microservicios en comercio electrónico ofrece varias ventajas. En primer lugar, favorece la escalabilidad horizontal, permitiendo que un servicio específico pueda aumentar su capacidad de procesamiento sin necesidad de escalar toda la aplicación. En segundo lugar, incrementa la resiliencia, ya que si un microservicio falla, el resto de la aplicación puede seguir operando sin interrupciones significativas. Finalmente, facilita la integración y la entrega continuas (CI/CD), permitiendo actualizaciones frecuentes sin afectar la operatividad de la plataforma (Espinoza, 2021).

A esta arquitectura se suman tecnologías complementarias como los contenedores (Docker), que facilitan la portabilidad de aplicaciones, y los sistemas de orquestación de contenedores (Kubernetes), que permiten gestionar el despliegue automático, balanceo de carga, escalado dinámico y monitoreo de recursos. Estas herramientas son esenciales en el comercio electrónico moderno, donde los picos de demanda —como el “Black Friday” o el “Cyber Monday”— requieren una infraestructura capaz de soportar millones de usuarios simultáneos.

Además, la arquitectura técnica del comercio electrónico debe contemplar componentes orientados a la seguridad y confianza. Entre ellos destacan la encriptación de datos mediante protocolos SSL/TLS, el uso de certificados digitales, la implementación de firewalls y sistemas de detección de intrusos, así como políticas de respaldo y recuperación ante desastres. Estos

elementos son necesarios para prevenir fraudes, proteger la información sensible de los clientes y garantizar la continuidad de los servicios.

En síntesis, la arquitectura técnica del comercio electrónico ha evolucionado desde infraestructuras básicas de intercambio de información hasta complejas plataformas digitales que integran microservicios, contenedores, protocolos de seguridad avanzados y servicios en la nube. Este conjunto no solo responde a las demandas de escalabilidad y disponibilidad propias de los entornos digitales, sino que también se adapta a los cambios tecnológicos y a las expectativas crecientes de los consumidores, consolidándose como un pilar fundamental en la economía digital contemporánea.

Procesos de Compra en el Comercio Electrónico

Los procesos de compra en el comercio electrónico abarcan más que la simple transacción. Según Malca (2001), este ciclo comprende la atracción del cliente, la interacción durante la negociación, la ejecución de la compra y la atención posventa, cada uno de los cuales garantiza la satisfacción del consumidor y la fidelización a largo plazo.

En la práctica, Torres y Guerra (2012) explican que el proceso inicia con la búsqueda y selección de productos en catálogos digitales, seguido por el registro de pedidos y la confirmación de pago en línea. Posteriormente, se ejecuta la entrega física o digital del producto y, finalmente, se ofrecen servicios de soporte y seguimiento al cliente. Estos procesos se distinguen por su rapidez, accesibilidad global y disponibilidad permanente, lo que constituye una ventaja frente al comercio tradicional.

De acuerdo con el documento *Innovación en E-commerce* (2022), los procesos de compra modernos incorporan automatización de pedidos, chatbots con inteligencia artificial para atención al cliente, sistemas de recomendación personalizados y seguimiento de envíos en tiempo real. A su vez, la integración de billeteras electrónicas y plataformas de pago digitales ha reducido la fricción en las transacciones, aumentando la confianza de los usuarios y mejorando la tasa de conversión. En este sentido, los procesos de compra ya no se entienden únicamente como una secuencia lineal, sino como un viaje del cliente (customer journey) que se extiende más allá de la compra inicial y busca generar fidelidad y recompra.

Seguridad y Confianza Dentro del Contexto de Comercio Electrónico

La seguridad es uno de los pilares más críticos en el comercio electrónico, ya que constituye la base de la confianza del consumidor. Torres y Guerra (2012) advierten que los principales riesgos incluyen fraudes electrónicos, suplantación de identidad (phishing), robo de datos y ataques informáticos. Estos factores generan desconfianza en los usuarios y pueden frenar el crecimiento del sector si no se abordan adecuadamente.

Para Malca (2001), la confianza se construye a partir de marcos normativos claros, estándares de seguridad tecnológica (como certificados digitales y encriptación), y políticas de protección de datos. Además, la transparencia en la información, las garantías de devolución y la reputación digital de las empresas son elementos clave para fortalecer la credibilidad en las transacciones electrónicas. En este sentido, la seguridad no solo es un componente técnico, sino también organizacional y legal.

De La Cruz Rodríguez, Méndez Fernández y Méndez Fernández (2023) sostienen que la seguridad de la información en el comercio electrónico debe alinearse con normas internacionales como la ISO 27001, que establece políticas, controles y procesos para garantizar la confidencialidad, integridad y disponibilidad de la información. Estos autores destacan que la implementación de sistemas de gestión de seguridad basados en ISO permite reducir riesgos, prevenir fraudes y aumentar la confianza de los clientes en entornos digitales.

Asimismo, los autores resaltan que la seguridad no solo depende de las medidas técnicas, sino también de la concienciación de los usuarios y la capacitación del personal, ya que el factor humano es uno de los principales vectores de vulnerabilidad. De esta manera, prácticas como la autenticación multifactor, la biometría, los certificados digitales y el monitoreo constante de incidentes se convierten en elementos fundamentales para garantizar la protección de datos sensibles en las transacciones electrónicas.

En síntesis, la seguridad en el comercio electrónico debe entenderse como un sistema integral que combina medidas técnicas (encriptación, firewalls, autenticación multifactor), legales (marcos regulatorios de protección de datos) y organizacionales (protocolos de gestión de riesgos y formación en seguridad). La confianza, por tanto, se convierte en un activo estratégico que no solo asegura la permanencia de los clientes, sino que también impulsa la competitividad de las empresas en el entorno digital.

Marketing Digital

El marketing digital constituye una de las principales herramientas estratégicas del comercio electrónico. A través de este, las empresas logran atraer y retener clientes mediante el

uso de publicidad segmentada, motores de búsqueda, redes sociales y correo electrónico. Según Torres y Guerra (2012), el marketing en la red surgió como una evolución natural de la publicidad tradicional y se consolidó con la aparición de bases de datos y aplicaciones dinámicas, que permiten personalizar ofertas en función del perfil del consumidor.

Por su parte, Malca (2001) enfatiza que el marketing digital no solo cumple una función promocional, sino que también forma parte del ciclo integral del comercio electrónico, al intervenir en la atracción de clientes, la gestión de relaciones (CRM) y la creación de comunidades virtuales alrededor de marcas y productos. De este modo, se convierte en una estrategia indispensable para generar valor agregado y construir lealtad en entornos digitales.

De acuerdo con el documento complementario sobre marketing y e-commerce, el marketing digital en el comercio electrónico se basa en estrategias omnicanales, donde los diferentes puntos de contacto (tiendas online, redes sociales, aplicaciones móviles y marketplaces) se articulan de manera coherente para mejorar la experiencia del cliente. Asimismo, la automatización del marketing y el uso de herramientas de análisis de datos (big data y machine learning) permiten conocer mejor los hábitos de consumo y anticipar las necesidades de los usuarios, diseñando campañas más efectivas y personalizadas.

Otro aspecto destacado es el papel de la interactividad y la inmediatez, ya que las plataformas digitales permiten una comunicación bidireccional entre la empresa y el consumidor, fomentando relaciones más cercanas y de confianza. Además, el marketing digital actual se apoya en el contenido de valor (content marketing) y en la influencia de líderes de opinión en redes sociales (influencer marketing), lo que amplía el alcance de las marcas y potencia su credibilidad en comunidades digitales específicas.

En síntesis, el marketing digital en el comercio electrónico no solo es un medio para publicitar productos, sino un ecosistema integral de estrategias y herramientas tecnológicas que permiten atraer, convertir y fidelizar clientes. Su importancia radica en que se adapta continuamente a los cambios del entorno digital, respondiendo a la globalización de los mercados y a la creciente demanda de experiencias personalizadas por parte de los consumidores.

Innovación y Tendencias en el Comercio Electrónico

El comercio electrónico se caracteriza por estar en constante evolución. Torres y Guerra (2012) identifican cuatro generaciones en su desarrollo: desde los catálogos estáticos, pasando por los centros comerciales virtuales, hasta llegar a plataformas dinámicas con sistemas de pago seguro y experiencias interactivas. Actualmente, se observa una tendencia hacia la personalización avanzada, la movilidad y la omnicanalidad, que busca integrar diferentes dispositivos y plataformas para ofrecer una experiencia unificada al cliente.

Malca (2001) añade que la innovación en el comercio electrónico también implica la incorporación de tecnologías emergentes, como la inteligencia artificial aplicada al análisis de datos de consumo, la realidad aumentada para mejorar la experiencia de compra y la blockchain como soporte de pagos más seguros y transparentes. Estas tendencias confirman que el e-commerce no solo es un canal de ventas, sino un ecosistema en permanente transformación, alineado con la globalización y la digitalización de la economía.

El documento *Innovación en E-commerce* (2022) destaca que las tendencias actuales giran en torno a la experiencia del usuario (UX), integrando herramientas como la realidad

virtual, asistentes de voz para compras conversacionales y el comercio social a través de redes sociales. Además, plantea que la sostenibilidad se ha convertido en un eje estratégico, impulsando iniciativas como empaques ecológicos, optimización de la logística inversa y reducción de la huella de carbono en la cadena de suministro. Finalmente, menciona la consolidación del live shopping, es decir, las compras en vivo mediante transmisiones en streaming, como una de las innovaciones más relevantes para captar a consumidores jóvenes y digitales.

Marco Legal y Normativo

Legislación Nacional sobre Comercio Electrónico

En Colombia, el comercio electrónico se encuentra regulado por un marco normativo que busca garantizar seguridad jurídica en las transacciones digitales y proteger los derechos de los consumidores.

La Ley 527 de 1999 constituye la piedra angular de esta regulación, al establecer las bases para el uso de mensajes de datos, la firma digital y las funciones de las entidades de certificación. Con esta ley, Colombia se alineó con la Ley Modelo de la CNUDMI sobre Comercio Electrónico de 1996, favoreciendo la confianza en los negocios electrónicos al equiparar jurídicamente los documentos electrónicos con los documentos en soporte físico (Gómez, 2004; Guio Molano, 2019).

Posteriormente, la Ley 1480 de 2011 o Estatuto del Consumidor reforzó la protección al consumidor en el entorno digital, estableciendo derechos como el de retracto, la reversión del pago y el acceso a información clara y veraz. De igual forma, otorgó competencias a la

Superintendencia de Industria y Comercio (SIC) para ejercer funciones jurisdiccionales y resolver controversias derivadas de relaciones de consumo en línea (Guio Molano, 2019).

El Decreto 1074 de 2015 complementa este marco al exigir que los proveedores que ofrezcan bienes y servicios mediante mecanismos de venta a distancia o comercio electrónico brinden información suficiente y precisa sobre identidad, características del producto, precios, gastos adicionales y tiempos de entrega (Ministerio de Industria y Comercio, 2015).

Desde una perspectiva más amplia, estudios académicos han resaltado que, a pesar de la existencia de este marco legal, persisten desafíos en la aplicación práctica de los derechos del consumidor en entornos digitales. Uno de los principales problemas identificados es la asimetría de información entre oferentes y consumidores, lo que genera riesgos de publicidad engañosa, cláusulas abusivas y limitaciones en el ejercicio efectivo del derecho de retracto (Gutiérrez, 2017). En este sentido, la normativa nacional se articula con estándares internacionales de protección al consumidor en comercio electrónico, como las Directrices de Naciones Unidas para la Protección del Consumidor y las recomendaciones de la Unión Europea en materia de contratación digital.

En conjunto, la normativa colombiana busca equilibrar la relación contractual entre las partes y generar confianza en el comercio electrónico, aunque la doctrina advierte la necesidad de fortalecer la capacidad de control y vigilancia, así como de promover una mayor educación digital en los consumidores para el ejercicio pleno de sus derechos.

Normas Internacionales Aplicables

El análisis del marco legal y normativo del comercio electrónico evidencia que Colombia ha avanzado de manera significativa en la construcción de un entorno regulatorio que brinda seguridad jurídica a las transacciones digitales. La Ley 527 de 1999 marcó un punto de partida al reconocer la validez de los mensajes de datos y firmas digitales, en consonancia con la Ley Modelo de la CNUDMI (1996), lo que permitió al país alinearse con estándares internacionales (Guerrero, 2016). Posteriormente, la Ley 1480 de 2011 fortaleció los derechos del consumidor, estableciendo mecanismos como la reversión del pago y el derecho de retracto, esenciales para el comercio electrónico (Gutiérrez, 2017).

El marco internacional, compuesto por instrumentos como las Directrices de las Naciones Unidas para la Protección del Consumidor, la Convención de Viena sobre Compraventa Internacional de Mercaderías y las directivas de la Unión Europea, ha servido como referencia para el desarrollo normativo nacional. Estos lineamientos han influido en la manera en que Colombia concibe la protección de los usuarios digitales, aunque persisten retos en torno a la armonización normativa y la jurisdicción en contratos electrónicos internacionales (Quintero, 2012).

Se concluye que, aunque Colombia cuenta con una base sólida de legislación en materia de comercio electrónico, todavía existen desafíos asociados a la asimetría de información, la prevención de cláusulas abusivas y la protección del consumidor en transacciones transfronterizas. Para afrontarlos, resulta necesario fortalecer los mecanismos de cooperación internacional, la educación digital de los consumidores y la capacidad de vigilancia de las autoridades nacionales. De esta forma, el marco legal podrá responder a las dinámicas

cambiantes del comercio electrónico y consolidar un entorno confiable y competitivo a nivel global.

Aspectos Fiscales y Tributarios

El crecimiento del comercio electrónico en Colombia no solo ha planteado retos en materia de regulación contractual y de protección al consumidor, sino también en lo relativo a los aspectos fiscales y tributarios. En un entorno digital caracterizado por la desmaterialización de las transacciones y la participación de actores nacionales e internacionales, el Estado ha debido diseñar mecanismos que permitan garantizar el recaudo fiscal y la equidad tributaria frente a las operaciones en línea.

En el ámbito nacional, la legislación reconoce que las transacciones electrónicas generan las mismas obligaciones tributarias que las realizadas de forma presencial. De acuerdo con la Ley 527 de 1999, los mensajes de datos y documentos electrónicos tienen plena validez jurídica, lo que implica que las facturas electrónicas y los contratos celebrados por medios digitales se encuentran sujetos a los tributos previstos en el ordenamiento, como el impuesto sobre las ventas (IVA) y el impuesto sobre la renta (Guio Molano, 2019).

El Estatuto del Consumidor (Ley 1480 de 2011) también se relaciona indirectamente con la dimensión fiscal, en la medida en que exige a los proveedores informar con claridad el precio total de los bienes o servicios, incluyendo impuestos, cargos adicionales y costos de envío. Este principio de transparencia no solo protege al consumidor, sino que asegura el cumplimiento de las obligaciones tributarias derivadas de la operación (Gutiérrez, 2017).

Por su parte, el Decreto 1074 de 2015 refuerza la obligación de los oferentes en ventas a distancia de suministrar información clara sobre la identidad del vendedor, lo cual contribuye a la trazabilidad fiscal y a la identificación de los responsables tributarios. Esto resulta fundamental en un contexto donde la evasión y la informalidad representan desafíos recurrentes para el control estatal (Ministerio de Industria y Comercio, 2015).

A nivel internacional, organismos como la Organización para la Cooperación y el Desarrollo Económicos (OCDE) han resaltado la necesidad de diseñar políticas fiscales adaptadas al comercio digital, particularmente en lo relacionado con la tributación de servicios prestados desde el extranjero. Aunque Colombia ha avanzado en la adopción de medidas para gravar plataformas digitales y servicios transfronterizos, persiste el reto de armonizar estas disposiciones con los marcos internacionales, a fin de evitar la doble tributación o, en su defecto, vacíos normativos que favorezcan la evasión (Guerrero, 2016).

En conclusión, los aspectos fiscales y tributarios del comercio electrónico en Colombia se enmarcan en la equiparación de las transacciones digitales con las tradicionales, asegurando el cumplimiento de impuestos como el IVA y la renta. Sin embargo, los desafíos actuales exigen una actualización permanente de la normativa, así como una cooperación internacional más estrecha para garantizar que el comercio electrónico se desarrolle en un escenario de equidad tributaria y sostenibilidad fiscal.

Seguridad en Pagos y Protección de Datos

Uno de los principales retos del comercio electrónico es garantizar la seguridad de los pagos y la protección de los datos personales de los usuarios. Estos aspectos son determinantes

para generar confianza en los consumidores, quienes frecuentemente expresan preocupación frente al uso indebido de su información financiera y a los riesgos de fraude en línea.

En el contexto colombiano, la Ley 527 de 1999 reconoce la validez de las firmas digitales y los mensajes de datos, lo cual constituye un fundamento jurídico para la implementación de sistemas de pago electrónicos confiables. No obstante, como advierte la doctrina, la sola existencia de un marco legal no es suficiente, ya que la confianza del consumidor depende en gran medida de la implementación de protocolos técnicos y de seguridad internacionales (Guio Molano, 2019).

De acuerdo con Villatoro Alas (2015), la seguridad en las transacciones en línea se apoya en mecanismos como la encriptación de datos, la autenticación mediante múltiples factores y el uso de certificados digitales que permiten verificar la identidad de los participantes. Asimismo, el autor resalta la importancia del cumplimiento de estándares internacionales como el PCI DSS (Payment Card Industry Data Security Standard), que establece requisitos de seguridad para el manejo de información financiera en sistemas de pago con tarjeta.

En lo que respecta a la protección de datos personales, la Ley 1581 de 2012 establece principios como la legalidad, la finalidad, la libertad y la seguridad en el tratamiento de la información. En el comercio electrónico, esto se traduce en la obligación de los proveedores de garantizar la confidencialidad de los datos de los consumidores, así como en el deber de obtener un consentimiento informado para su recolección y tratamiento.

La seguridad no debe entenderse únicamente desde la dimensión técnica, sino como un proceso integral que incluye educación digital al consumidor, transparencia en las políticas de

privacidad y la existencia de mecanismos de resolución de conflictos en línea. En este sentido, la Superintendencia de Industria y Comercio cumple un papel central como entidad de vigilancia y sanción frente a incumplimientos en materia de habeas data y protección al consumidor digital (Villatoro Alas, 2015).

Casos de Estudio en Comercio Electrónico

Ejemplos Nacionales

El comercio electrónico en Colombia ha mostrado un crecimiento constante durante las últimas dos décadas, consolidándose como un mercado pionero en la región. Según Zuluaga (2018), este desarrollo ha estado marcado por la emergencia de plataformas nacionales que, en sus primeras etapas, lograron captar un número importante de consumidores locales antes de la entrada masiva de gigantes internacionales del e-commerce.

Uno de los casos emblemáticos es Mercado Libre Colombia, que, aunque hace parte de un grupo multinacional, se convirtió en una de las primeras plataformas de confianza para los usuarios colombianos. Su éxito se debió principalmente a la implementación de sistemas de pago seguro como Wompi, que permitieron disminuir la desconfianza hacia las transacciones digitales, además de una amplia oferta de productos (Zuluaga, 2018).

Otro ejemplo significativo lo constituye Éxito.com, la tienda en línea del Grupo Éxito, que aprovechó su infraestructura física y logística para adaptarse al comercio digital. Según Zuluaga (2018), este modelo híbrido entre canales presenciales y digitales permitió integrar la omnicanalidad en Colombia, ofreciendo opciones como la compra en línea con recogida en

tienda, lo cual generó confianza en los consumidores y facilitó la adopción del comercio electrónico.

Asimismo, el caso de Linio Colombia ilustra cómo la competencia de actores internacionales influyó en el desarrollo del e-commerce local. Aunque inicialmente logró posicionarse con una estrategia agresiva en precios y promociones, la falta de una logística sólida y la creciente presencia de Amazon y AliExpress limitaron su sostenibilidad en el mercado (Zuluaga, 2018).

Por otro lado, en el ámbito regional se encuentran experiencias que reflejan el impacto del comercio electrónico en ciudades intermedias. Tobar (2015) analizó el caso de grandes y medianas empresas en Sogamoso (Boyacá), encontrando que el comercio electrónico no solo mejoró la visibilidad y el alcance de las organizaciones, sino que también representó una herramienta para reducir costos de operación y fortalecer la competitividad. Sin embargo, el autor señala que las barreras culturales, la resistencia al cambio y la desconfianza en los sistemas de pago siguen siendo obstáculos para una adopción más amplia en este tipo de contextos empresariales.

Estos ejemplos demuestran que, si bien Colombia ha sido un mercado pionero en la región, el comercio electrónico nacional enfrenta desafíos significativos frente a la consolidación de grandes plataformas extranjeras y las limitaciones locales. No obstante, también evidencian la capacidad de adaptación de actores locales, desde grandes superficies hasta empresas regionales, para integrar modelos digitales y responder a las necesidades cambiantes de los consumidores.

Análisis Comparativo

El análisis de los casos de comercio electrónico en Colombia permite observar tanto fortalezas como limitaciones frente a la dinámica internacional. Según Zuluaga (2018), Colombia fue uno de los primeros países en la región en implementar plataformas de comercio electrónico, lo que le otorgó una posición pionera en América Latina. Sin embargo, este liderazgo inicial se ha visto amenazado por la llegada de gigantes del e-commerce como Amazon y AliExpress, que han impuesto estándares de servicio y logística difíciles de igualar para los actores locales.

En contraste, países como Brasil y México han logrado consolidar ecosistemas digitales más robustos, donde empresas nacionales e internacionales conviven de manera competitiva. En Brasil, por ejemplo, Magazine Luiza y B2W Digital compiten de forma directa con Amazon gracias a una infraestructura logística altamente desarrollada, mientras que en México, plataformas como Liverpool han sabido integrar exitosamente la omnicanalidad en sus operaciones. En Colombia, aunque casos como Éxito.com han seguido esta estrategia, aún se perciben vacíos en la cobertura logística y en la velocidad de entregas, lo cual constituye una brecha competitiva frente a estos mercados (Zuluaga, 2018).

Otro punto de comparación es el nivel de confianza de los consumidores. Tobar (2015) destaca que en ciudades intermedias como Sogamoso, la desconfianza en los medios de pago electrónicos y la resistencia cultural al cambio constituyen barreras importantes. Esta situación contrasta con mercados más maduros como el europeo, donde la cultura digital está más consolidada y la adopción de pagos en línea es generalizada. En países como España o Alemania, por ejemplo, los consumidores realizan transacciones en línea con altos niveles de

seguridad percibida gracias a la autenticación de dos factores y al cumplimiento de protocolos PCI DSS.

En el plano regional, Colombia comparte con otros países latinoamericanos retos asociados a la informalidad y a la limitada inclusión financiera. Mientras que en Chile el crecimiento del e-commerce se ha visto favorecido por altos índices de bancarización y por políticas públicas que promueven la digitalización, en Colombia el acceso desigual a medios de pago electrónicos continúa siendo un obstáculo. No obstante, la penetración creciente de billeteras digitales como Nequi y Daviplata ha contribuido a reducir esta brecha en los últimos años (Zuluaga, 2018; Tobar, 2015).

Lecciones Aprendidas para el Proyecto

Del análisis de los casos nacionales y su contraste con experiencias internacionales se pueden extraer lecciones relevantes que orientan la formulación de un proyecto de comercio electrónico en Colombia.

La primera lección está relacionada con la confianza y la seguridad. Tanto Zuluaga (2018) como Tobar (2015) coinciden en que la desconfianza hacia los pagos electrónicos es una de las principales limitaciones del mercado colombiano. La experiencia de Mercado Libre, a través de Wompi, demuestra cómo un sistema de pago propio puede contribuir a fortalecer la confianza del consumidor y dinamizar las ventas.

Otra enseñanza se deriva de la omnicanalidad. La integración entre canales digitales y físicos, como en el caso de Éxito.com, evidencia la importancia de ofrecer al consumidor

múltiples alternativas de compra y entrega. Esta estrategia, adoptada también en países como México y Chile, refuerza la experiencia del usuario y amplía el alcance de las plataformas.

La logística se presenta como un factor decisivo. La trayectoria de Linio en Colombia muestra que la falta de una red de distribución eficiente limita la permanencia en el mercado frente a competidores internacionales. Por el contrario, la experiencia de empresas brasileñas como Magazine Luiza demuestra que una infraestructura logística sólida constituye una ventaja competitiva esencial.

Asimismo, resulta fundamental la adaptación a contextos locales. Tobar (2015) destaca que en ciudades intermedias es necesario acompañar la digitalización con procesos de educación digital para empresarios y consumidores, con el fin de superar la resistencia cultural al comercio electrónico.

Finalmente, la presencia de actores globales enseña que la competencia internacional puede ser un estímulo para la innovación. La entrada de Amazon y AliExpress en Colombia ha obligado a las empresas locales a mejorar su servicio, incorporar estándares de seguridad más altos y explorar nuevos modelos de negocio, lo que impulsa el fortalecimiento del ecosistema digital nacional.

Fundamentos Teóricos de la Gestión en comercio Electrónico

Teoría de la Cadena de Suministro

La gestión en comercio electrónico no solo se limita a la implementación tecnológica, sino que también se fundamenta en teorías administrativas y modelos de negociación que orientan la toma de decisiones en entornos digitales. En este contexto, los procesos de gestión

deben articular aspectos como la coordinación de la cadena de suministros, la fijación de precios, la relación con los proveedores y la satisfacción del cliente final.

Ayaviri-Panozo y Ramírez-Correa (2019) destacan que una de las bases conceptuales más relevantes para la gestión del comercio electrónico proviene de las teorías utilizadas en la negociación de precios colaborativos dentro de las cadenas de suministro. Estas teorías explican cómo las empresas pueden cooperar, compartir información y diseñar estrategias conjuntas para establecer precios más competitivos y equitativos. En el comercio electrónico, este enfoque resulta fundamental para coordinar a múltiples actores, desde proveedores hasta distribuidores digitales, que interactúan en un mismo entorno virtual.

Entre las teorías señaladas por los autores se encuentran la teoría de juegos, aplicada para analizar el comportamiento estratégico de los participantes en la negociación; la teoría de la agencia, que explica las relaciones entre las empresas y los intermediarios en un contexto de asimetría de información; y la teoría de la cooperación, que resalta la importancia de la colaboración mutua como mecanismo para lograr beneficios compartidos (Ayaviri-Panozo & Ramírez-Correa, 2019).

Desde una perspectiva complementaria, Jingyi (2021) enfatiza que la aplicación de la cadena de suministro en el comercio electrónico requiere de un rediseño estructural, en el que la tecnología digital y la integración logística desempeñan un papel clave. El autor sostiene que la eficiencia en el e-commerce depende de la capacidad de sincronizar procesos de aprovisionamiento, almacenamiento, distribución y atención al cliente, con el fin de garantizar tiempos de entrega reducidos y experiencias de compra satisfactorias.

La gestión de la cadena de suministro aplicada al comercio electrónico permite a las empresas responder con mayor rapidez a la demanda, reducir costos operativos y optimizar la relación con los consumidores. Además, Jingyi (2021) señala que el uso de tecnologías emergentes como el big data y la inteligencia artificial contribuye a predecir patrones de consumo y a gestionar inventarios de manera más eficiente, lo que incrementa la competitividad de las organizaciones en mercados digitales.

En el marco del comercio electrónico, estas perspectivas teóricas permiten comprender que la gestión debe orientarse no solo a la eficiencia interna de la empresa, sino también a la colaboración estratégica con socios de la cadena de suministro. Esto implica un cambio de paradigma hacia modelos de gestión flexibles, interconectados y centrados en el consumidor, capaces de responder a la dinámica cambiante de los mercados digitales y de generar ventajas competitivas sostenibles.

Teoría del Comportamiento del Consumidor en Entornos Digitales

El estudio del comportamiento del consumidor en entornos digitales constituye un eje central para comprender el funcionamiento del comercio electrónico, dado que las decisiones de compra en línea se ven influenciadas por factores psicológicos, sociales y tecnológicos que difieren de los contextos tradicionales.

Díaz Doria (2023) sostiene que la conducta del consumidor digital está estrechamente vinculada a la percepción de valor y a las estrategias de precios implementadas por las empresas de comercio electrónico. En este escenario, aspectos como la facilidad de navegación, la confianza en los medios de pago, la reputación de la plataforma y la rapidez en la entrega inciden

directamente en la disposición de compra. A diferencia del consumidor tradicional, el usuario digital cuenta con mayor acceso a la información y a la comparación de precios, lo que modifica su sensibilidad hacia las estrategias de fijación dinámica o personalizada de precios.

El autor también resalta que la teoría del comportamiento del consumidor en entornos digitales debe analizarse a partir de dos dimensiones fundamentales: la racional y la emocional. En la primera, los consumidores buscan maximizar beneficios y minimizar costos mediante comparaciones objetivas entre múltiples alternativas disponibles en línea. En la segunda, las emociones y percepciones asociadas a la experiencia digital, como la usabilidad de la página, la interacción con el servicio al cliente y la influencia de reseñas o comentarios, se convierten en factores determinantes en la toma de decisiones.

En esta misma línea, Vasilica-Maria (2022) plantea que la intención de compra en sectores como la moda está condicionada por factores específicos como la confianza en la marca, la calidad percibida de los productos, el diseño de la plataforma y la influencia social. La autora propone un modelo teórico para América Latina en el que variables como la percepción de riesgo, el nivel de satisfacción previo y la influencia de pares en redes sociales son decisivas para que los consumidores concreten transacciones digitales. Asimismo, destaca que la experiencia de usuario (UX) y el grado de personalización ofrecido por la plataforma incrementan la probabilidad de conversión, reforzando la importancia de la analítica de datos y las recomendaciones inteligentes en las decisiones de compra.

Por su parte, Colmenero Seneira (2025) amplía esta perspectiva al analizar el comportamiento del consumidor en entornos de comercio online desde una visión más integral. El autor sostiene que la experiencia digital no solo depende de factores funcionales o

emocionales, sino también de elementos contextuales como la accesibilidad tecnológica, la omnicanalidad y la sostenibilidad percibida de las marcas. En este sentido, la elección de un producto o servicio en línea está mediada por la alineación entre los valores de la empresa y las expectativas sociales de los consumidores, lo que refuerza la tendencia hacia un consumo digital más consciente y responsable.

La articulación de los planteamientos de Díaz Doria (2023), Vasilica-Maria (2022) y Colmenero Seneira (2025) permite comprender que el comportamiento del consumidor digital no puede analizarse únicamente desde la perspectiva económica de los precios, sino también desde factores sociales, emocionales y contextuales que influyen en la decisión de compra. La confianza, la reputación de la marca, la experiencia de usuario, la influencia social y la coherencia con valores sostenibles se configuran como variables clave para explicar y predecir la conducta de los usuarios en entornos digitales.

Principios de Gestión de Inventarios

La gestión de inventarios constituye un eje estratégico dentro del comercio electrónico, ya que de su eficiencia depende no solo la disponibilidad de productos para los clientes, sino también la sostenibilidad operativa y financiera de las empresas. En los entornos digitales, los inventarios se convierten en un recurso dinámico que debe adaptarse a la demanda fluctuante, la inmediatez de los pedidos y la integración con plataformas logísticas de alcance global.

Candela Mantilla (2025) señala que en el contexto del comercio electrónico, la gestión de inventarios debe orientarse hacia la optimización de recursos mediante sistemas digitales que garanticen un control preciso de las existencias, evitando tanto el exceso de stock como el

desabastecimiento. La autora destaca que uno de los principios fundamentales consiste en la integración de tecnologías de información que permitan sincronizar las bases de datos de ventas, compras y almacén, lo que asegura una mayor visibilidad y eficiencia en la toma de decisiones.

Entre los principios aplicados se encuentra la utilización de sistemas de inventarios en tiempo real, los cuales facilitan la actualización constante de la información y permiten responder de manera inmediata a la variación en los niveles de demanda. Asimismo, Candela Mantilla (2025) enfatiza la importancia de aplicar modelos predictivos basados en el análisis de datos históricos y en la estacionalidad del consumo, lo que posibilita anticipar necesidades y planificar con mayor exactitud las compras y la reposición de productos.

Otro principio clave es la vinculación de la gestión de inventarios con la estrategia logística. En el comercio electrónico, la eficiencia en el manejo de los inventarios no solo garantiza la disponibilidad de productos, sino que también impacta directamente en la experiencia del cliente al reducir los tiempos de entrega y mejorar la fiabilidad del servicio. Esto requiere que las empresas coordinen sus procesos de almacenamiento, transporte y distribución en una plataforma integrada.

En esta misma línea, Malca (2001) sostiene que el comercio electrónico demanda una gestión eficiente de inventarios para asegurar la continuidad en el servicio, ya que el consumidor digital exige inmediatez, disponibilidad y transparencia en el proceso de compra. La digitalización de los inventarios permite a las empresas alcanzar mayor competitividad, al integrar la información en tiempo real con la demanda proyectada y con los procesos de atención al cliente.

De manera complementaria, Castañeda y Zavala (2012) señalan que los inventarios en el comercio electrónico no solo cumplen una función de almacenamiento, sino que representan un elemento estratégico en la generación de valor. Los autores destacan que el control adecuado de inventarios favorece la reducción de costos operativos, mejora los tiempos de entrega y contribuye a la fidelización del cliente al garantizar un servicio confiable y continuo.

Finalmente, la gestión de inventarios en e-commerce también debe atender criterios de costo y rentabilidad. El equilibrio entre el capital invertido en stock y la rotación de productos es esencial para mantener márgenes de utilidad sostenibles. En este sentido, los principios de la gestión de inventarios en entornos digitales deben combinar eficiencia tecnológica, control financiero y orientación al cliente, consolidando así una ventaja competitiva en mercados cada vez más dinámicos (Candela Mantilla, 2025; Malca, 2001).

Conceptos Clave para el Éxito del Comercio Electrónico

Confianza y Seguridad en el Comercio Electrónico

La confianza y la seguridad constituyen pilares fundamentales para el éxito de cualquier proyecto de comercio electrónico. En un entorno donde las interacciones entre clientes y empresas se realizan de manera virtual, la percepción de riesgo se convierte en un factor decisivo que puede determinar el éxito o fracaso de una transacción. Jones (2010) señala que uno de los principales motivos de fracaso en los proyectos de comercio electrónico ha sido la ausencia de estrategias claras para generar confianza en los usuarios y garantizar la protección de sus datos personales y financieros.

La confianza en el comercio electrónico está asociada a la transparencia en la información ofrecida por las empresas, al cumplimiento de las condiciones de venta y a la reputación digital que estas construyen con el tiempo. La falta de credibilidad o de coherencia entre la oferta y el servicio recibido genera desconfianza y limita la disposición de los consumidores a realizar compras en línea (Jones, 2010).

En relación con la seguridad, Jones (2010) destaca que los consumidores perciben un alto riesgo en lo referente a la protección de datos sensibles, como la información financiera y los métodos de pago. Por ello, las empresas deben implementar mecanismos de seguridad robustos que incluyan encriptación de datos, sistemas de autenticación y políticas claras de privacidad. Estas medidas, además de proteger las transacciones, envían un mensaje de confiabilidad al cliente, fortaleciendo la relación de largo plazo.

Malca (2001) complementa este enfoque señalando que la seguridad en el comercio electrónico no solo es técnica, sino también organizacional, ya que requiere políticas claras de gestión de la información y cumplimiento normativo. En este sentido, la confianza se genera cuando las empresas demuestran coherencia entre la protección de los datos y la transparencia de sus procesos. Por su parte, Castañeda y Zavala (2012) destacan que la construcción de confianza también depende de la percepción del consumidor frente a la capacidad de respuesta de la empresa, lo cual incluye la atención posventa, las devoluciones y el cumplimiento en los tiempos de entrega.

De esta manera, la confianza y la seguridad se configuran como condiciones imprescindibles para el desarrollo sostenible del comercio electrónico, ya que determinan tanto

la decisión de compra como la continuidad de la relación comercial en un entorno digital altamente competitivo (Jones, 2010; Malca, 2001).

Experiencia de Usuario (UX) como Factor Diferencial

En el comercio electrónico, la experiencia de usuario (UX) se ha consolidado como un elemento diferenciador que influye directamente en la percepción de valor y en la decisión de compra. La UX no solo se relaciona con el diseño estético de una plataforma digital, sino también con su facilidad de uso, rapidez, accesibilidad y capacidad de generar confianza en el consumidor. Según Jones (2010), muchos de los fracasos en proyectos de e-commerce han estado vinculados a la falta de atención en la usabilidad de los sistemas, lo cual provoca abandono de carritos de compra, pérdida de clientes y baja fidelización.

La simplicidad en la navegación, la claridad de la información y la rapidez en los procesos de búsqueda y pago son factores fundamentales para garantizar una experiencia satisfactoria. Jones (2010) destaca que los usuarios valoran especialmente los entornos digitales que minimizan los pasos innecesarios y que les permiten completar sus transacciones de manera ágil y segura.

Otro aspecto clave de la UX es la personalización. En la medida en que las plataformas de comercio electrónico integran herramientas que reconocen los intereses y comportamientos del usuario, se logra un mayor nivel de cercanía y se incrementan las posibilidades de conversión. De acuerdo con Jones (2010), la adaptación de los contenidos y ofertas en función de las preferencias del cliente es un factor que contribuye a diferenciar a las empresas en un mercado digital altamente competitivo.

Malca (2001) aporta a este enfoque destacando que la experiencia digital debe garantizar no solo facilidad de uso, sino también confianza en el proceso, ya que el consumidor en línea requiere rapidez, seguridad y coherencia en la interacción con la plataforma. De manera complementaria, Castañeda y Zavala (2012) sostienen que la UX también está vinculada al servicio integral que recibe el cliente, lo cual incluye la atención en línea, la claridad en las políticas de compra y devolución, y la accesibilidad de la plataforma en distintos dispositivos. Estos factores fortalecen la fidelización y generan una ventaja competitiva sostenible.

En este sentido, la UX constituye un factor diferencial en el comercio electrónico, ya que no solo influye en la decisión de compra inmediata, sino también en la fidelización del cliente. Aquellas empresas que logran integrar la usabilidad, la personalización y la coherencia en la experiencia digital obtienen una ventaja competitiva en un entorno caracterizado por la innovación constante y la alta exigencia de los consumidores (Jones, 2010; Malca, 2001).

Lenguajes, Técnicas, Metodologías y Patrones de Diseño

Lenguajes de Programación

En el desarrollo de sistemas de comercio electrónico, la elección de los lenguajes de programación es un aspecto determinante para garantizar la funcionalidad, escalabilidad y experiencia del usuario. El frontend, encargado de la interacción directa con el cliente, requiere herramientas que permitan construir interfaces dinámicas, accesibles y seguras, adaptadas a los entornos digitales actuales.

De acuerdo con Gaona García (2024), los lenguajes de programación más utilizados en proyectos de e-commerce modernos tienden a priorizar aquellos que ofrecen flexibilidad y

compatibilidad con múltiples plataformas. En este contexto, JavaScript se consolida como el lenguaje fundamental para el desarrollo del frontend, gracias a su capacidad de ejecutar procesos en el navegador y de ofrecer interactividad en tiempo real. La versatilidad de JavaScript permite implementar desde funciones básicas de validación hasta integraciones complejas con APIs y servicios externos, aspectos indispensables en la automatización de ventas y envíos en plataformas de comercio electrónico.

Asimismo, el uso de React como biblioteca de JavaScript representa un avance significativo en la construcción de interfaces de usuario. React facilita el desarrollo de componentes reutilizables, lo que optimiza el mantenimiento y la escalabilidad de las aplicaciones. Según Gaona García (2024), la adopción de esta herramienta en proyectos de comercio electrónico responde a la necesidad de construir experiencias de usuario dinámicas y fluidas, que reduzcan los tiempos de carga y permitan gestionar grandes volúmenes de datos de manera eficiente.

Otro aspecto relevante del uso de React es su ecosistema, que incluye herramientas como React Router para la navegación, Redux para la gestión del estado y la integración con librerías de diseño que enriquecen la presentación visual. Estas características hacen que React sea ampliamente utilizado en el desarrollo de sistemas de e-commerce que buscan ofrecer una experiencia personalizada y optimizada para distintos dispositivos.

En el caso del backend, el lenguaje empleado determina la capacidad del sistema para gestionar procesos críticos como la autenticación de usuarios, la validación de transacciones y la comunicación con la base de datos. De acuerdo con Romero Sánchez (2015), el uso de ASP.NET en el desarrollo de portales de comercio electrónico ofrece ventajas como la seguridad integrada,

la escalabilidad y la capacidad de integrar componentes empresariales bajo la plataforma .NET. Esto permite crear aplicaciones robustas y confiables, capaces de soportar grandes volúmenes de operaciones en entornos de e-commerce.

En cuanto a la gestión de datos, MySQL se posiciona como uno de los motores de base de datos más utilizados en aplicaciones de comercio electrónico por su rendimiento, escalabilidad y compatibilidad con múltiples lenguajes y frameworks. Romero Sánchez (2015) señala que la combinación de ASP.NET con MySQL facilita la creación de portales de comercio electrónico al integrar un backend seguro con un motor de base de datos eficiente, optimizando el manejo de inventarios, clientes, transacciones y registros de pedidos.

En síntesis, la combinación de JavaScript y React en el frontend, junto con ASP.NET y MySQL en el backend, constituye una arquitectura tecnológica sólida para el desarrollo de sistemas de comercio electrónico. Esta integración no solo garantiza experiencias de usuario dinámicas y seguras, sino que también proporciona una infraestructura confiable para la gestión de datos y la escalabilidad del negocio digital (Gaona García, 2024; Romero Sánchez, 2015).

Técnicas de Desarrollo

El desarrollo de sistemas de comercio electrónico requiere la adopción de técnicas que permitan manejar la complejidad, la variabilidad y la escalabilidad de los productos software. Una de las aproximaciones más destacadas en este ámbito es el uso de líneas de productos de software, entendidas como un conjunto de sistemas que comparten características comunes, pero que pueden adaptarse mediante variaciones para responder a diferentes necesidades de negocio.

Según Laguna (2009), las líneas de productos representan una estrategia eficaz para el desarrollo de aplicaciones de comercio electrónico, ya que posibilitan la reutilización sistemática de componentes y modelos. La técnica central de este enfoque es el modelado de características, que identifica tanto los elementos obligatorios, como los opcionales y alternativos de un sistema, por ejemplo, la inclusión de métodos de pago, búsqueda avanzada o diferentes tipos de productos. Este modelado permite derivar múltiples configuraciones de portales de e-commerce a partir de una misma base.

Otro aspecto fundamental es la trazabilidad entre modelos y código, garantizada mediante el uso del mecanismo de combinación de paquetes en UML 2 (package merge). Este recurso facilita que las variaciones definidas en el modelo de características se reflejen directamente en la arquitectura y, posteriormente, en la implementación del sistema. Para la fase de codificación, se utilizan clases parciales en C# sobre la plataforma .NET, lo que permite consolidar de manera modular las distintas variantes del sistema de comercio electrónico sin comprometer la mantenibilidad.

Asimismo, la técnica se apoya en herramientas convencionales como editores UML, entornos de desarrollo integrados (IDEs) y plantillas en ASP.NET, lo que favorece la adopción de estas prácticas incluso por equipos con poca experiencia en líneas de productos. Esta aproximación ha demostrado ser útil en escenarios académicos e industriales, donde la generación de prototipos de portales de ventas puede realizarse en pocos minutos, integrando funcionalidades como catálogos multinivel, carritos de compras o pasarelas de pago (Laguna, 2009).

En síntesis, las técnicas de desarrollo basadas en líneas de productos ofrecen un marco metodológico sólido para enfrentar los retos del comercio electrónico. Su valor reside en la capacidad de gestionar la variabilidad y optimizar la reutilización de componentes, lo cual permite crear soluciones escalables, flexibles y alineadas con las demandas de un mercado altamente competitivo.

Metodologías de Desarrollo

El desarrollo de sistemas de software, y en particular de aplicaciones orientadas al comercio electrónico, requiere la adopción de metodologías que orienten de manera sistemática el ciclo de vida del proyecto. Estas metodologías permiten establecer procesos organizados para la planificación, diseño, construcción, pruebas e implementación de soluciones informáticas, asegurando la calidad y la adaptabilidad del producto final.

De acuerdo con Maida, Pacienza y Casiello (2015), las metodologías de desarrollo de software pueden agruparse en dos grandes enfoques: metodologías tradicionales y metodologías ágiles. Las metodologías tradicionales, como el modelo en cascada, se caracterizan por su naturaleza secuencial y rigurosa, en la cual cada fase debe completarse antes de iniciar la siguiente. Este tipo de enfoque resulta útil en proyectos de gran escala con requisitos bien definidos y estables, aunque puede presentar limitaciones cuando se requieren cambios frecuentes en el transcurso del desarrollo.

Por otra parte, las metodologías ágiles surgen como respuesta a la rigidez de los enfoques tradicionales, promoviendo la flexibilidad, la iteración y la colaboración continua con el cliente. Entre las metodologías ágiles más utilizadas en el desarrollo de software se encuentran Scrum,

Kanban y Extreme Programming (XP). Estas metodologías permiten adaptarse a contextos dinámicos, propios del comercio electrónico, donde la evolución del mercado y las demandas de los usuarios requieren de actualizaciones rápidas y constantes en las plataformas digitales.

Ojeda y Fuentes (2012) aportan una visión complementaria al establecer una taxonomía de los modelos y metodologías más empleados en el desarrollo de software. Según los autores, esta clasificación permite entender las fortalezas y debilidades de cada metodología en función de los objetivos del proyecto, la naturaleza de los requisitos y el entorno de implementación. Dentro de esta taxonomía, se reconocen metodologías predictivas, orientadas a la planificación y control estricto del proceso, y metodologías adaptativas, centradas en la flexibilidad y la capacidad de respuesta ante cambios.

La contribución de Ojeda y Fuentes (2012) resulta especialmente relevante en el comercio electrónico, donde la rápida evolución tecnológica y la competitividad del mercado obligan a seleccionar metodologías que garanticen no solo la calidad del software, sino también la velocidad de entrega y la capacidad de adaptación. Así, las metodologías ágiles se alinean con estas demandas, mientras que las metodologías tradicionales encuentran su espacio en proyectos con mayor estabilidad y menor variabilidad de los requisitos.

En este sentido, la elección de la metodología de desarrollo depende de factores como la magnitud del proyecto, el nivel de incertidumbre en los requisitos, la disponibilidad de recursos y la necesidad de rapidez en la entrega de funcionalidades. Para proyectos de comercio electrónico, donde la experiencia del usuario, la seguridad de las transacciones y la capacidad de adaptación son elementos esenciales, las metodologías ágiles suelen representar una opción más adecuada,

sin descartar la aplicación de enfoques híbridos que integren la estructura de los métodos tradicionales con la flexibilidad de los ágiles.

En síntesis, las metodologías de desarrollo de software constituyen un marco de referencia que orienta la gestión de proyectos tecnológicos, garantizando la calidad del producto y su alineación con los objetivos organizacionales. En el contexto del comercio electrónico, el uso de metodologías ágiles ha demostrado favorecer la innovación, la respuesta rápida a los cambios del mercado y la entrega continua de valor a los usuarios (Maida, Pacienza y Casiello, 2015; Ojeda y Fuentes, 2012).

Patrones de Diseño

En el desarrollo de software, los patrones de diseño constituyen soluciones probadas y reutilizables a problemas comunes que surgen durante la construcción de sistemas. Su valor radica en ofrecer un lenguaje compartido entre desarrolladores, facilitando la comunicación, la estandarización de prácticas y la mejora en la calidad del código.

De acuerdo con Gamma, Helm, Johnson y Vlissides (2000), los patrones de diseño se clasifican en tres grandes grupos: patrones de creación, patrones estructurales y patrones de comportamiento. Los patrones de creación, como Factory Method o Abstract Factory, proporcionan mecanismos para instanciar objetos de manera flexible y desacoplada, lo cual resulta esencial en aplicaciones de comercio electrónico que requieren gestionar múltiples tipos de productos o métodos de pago. Los patrones estructurales, como Adapter, Composite o Decorator, permiten definir relaciones eficientes entre objetos, favoreciendo la integración de módulos como catálogos, carritos de compras y sistemas de inventario. Por su parte, los patrones

de comportamiento, entre los que destacan Observer, Strategy o Command, se orientan a la interacción y responsabilidad de los objetos, facilitando funcionalidades dinámicas como notificaciones de pedidos, selección de promociones o procesamiento de transacciones.

Blanco (2008) amplía esta clasificación al describir los elementos que componen un patrón: nombre, problema, solución y consecuencias. Cada patrón funciona como un esqueleto adaptable a las necesidades de la aplicación, aportando claridad en el diseño y asegurando la consistencia del sistema. Además, resalta que la correcta aplicación de patrones debe acompañarse de la identificación de *antipatrones*, es decir, soluciones ineficaces o inadecuadas, como el “objeto todopoderoso” o el abuso del patrón Singleton. Estos antipatrones pueden generar diseños rígidos, difíciles de mantener o poco escalables, por lo que su detección temprana es clave para garantizar la calidad del software.

La aplicación de patrones de diseño en sistemas de comercio electrónico ofrece múltiples beneficios. En primer lugar, contribuye a la escalabilidad de la plataforma, ya que la reutilización de soluciones permite incorporar nuevas funcionalidades sin comprometer la arquitectura existente. En segundo lugar, favorece la mantenibilidad al organizar el código en estructuras claras y coherentes. Finalmente, incrementa la robustez del sistema, asegurando que procesos críticos, como las pasarelas de pago o la gestión de usuarios, se desarrollen bajo prácticas consolidadas y ampliamente validadas en la ingeniería de software.

En síntesis, los patrones de diseño representan un recurso fundamental para el desarrollo de aplicaciones de comercio electrónico, ya que facilitan la construcción de sistemas modulares, reutilizables y alineados con estándares de calidad. La integración de buenas prácticas y la

prevención de antipatrones fortalecen la eficiencia y confiabilidad del software en un entorno digital altamente competitivo (Gamma, Helm, Johnson y Vlissides, 2000; Blanco, 2008).

Diseño Metodológico

Tipo de Investigación

El presente trabajo se enmarca en una investigación aplicada con enfoque mixto, dado que integra métodos cuantitativos y cualitativos orientados a dar solución a una problemática real en un almacén de repuestos automotrices relacionada con la falta de un canal de ventas en línea. La investigación aplicada se caracteriza por generar conocimiento útil y práctico que permite optimizar procesos de negocio. Este enfoque se justifica plenamente en la naturaleza del proyecto, que busca implementar una plataforma de comercio electrónico para resolver una necesidad específica. Como señala Hernández, Fernández y Baptista (2014), este tipo de investigación se orienta a generar soluciones prácticas y aplicables. El enfoque mixto, por su parte, fortalece la validez de los resultados al combinar la recolección de datos numéricos con el análisis descriptivo, lo que permite una comprensión más profunda de la situación problemática.

Características de la Investigación Mixta Aplicada al Proyecto

La investigación desarrollada incorpora elementos cuantitativos y cualitativos. El componente cuantitativo se refleja en la recopilación y análisis de información relacionada con los procesos de inventario, el registro de insumos y la frecuencia de abastecimiento, datos que serán sistematizados a partir de la implementación del sistema propuesto. Paralelamente, el componente cualitativo se aplica a través de técnicas como la observación de los procesos de venta manuales y las entrevistas realizadas al personal, lo que permite obtener una comprensión más rica de los requisitos del usuario y de la dinámica organizacional.

Entre las características distintivas del estudio se encuentran su orientación hacia la solución de un problema específico, como la limitada visibilidad y el alcance de las ventas, y el

uso de datos empíricos obtenidos directamente de la organización y la plataforma en línea. Asimismo, el proyecto se caracteriza por la aplicación de instrumentos tecnológicos para la sistematización de la información y la combinación de métodos de análisis. Esta aproximación enriquece la comprensión del problema y la propuesta de solución, como lo argumentan Creswell y Plano Clark (2011), quienes destacan la capacidad del enfoque mixto para brindar una perspectiva más completa y detallada. El proyecto resalta que la toma de requerimientos, a través de estos métodos, es crucial para que la plataforma satisfaga las necesidades del cliente y sus usuarios.

Variables

En el desarrollo de la investigación se identificaron las siguientes variables clave. La variable independiente corresponde a la plataforma de comercio electrónico (e-commerce) propuesta. Esta variable es la causa o el factor que se manipula para observar su efecto. La variable dependiente. Esta es la variable de resultado, cuyo cambio se espera medir una vez que la variable independiente sea implementada.

Como variables de control se consideraron factores que podrían influir en los resultados, tales como los procesos de venta manuales, el tiempo de respuesta a los clientes y los errores en la gestión del inventario físico. Estas variables son fundamentales para medir de manera objetiva la incidencia que tiene la implementación del e-commerce en la eficiencia de los procesos de venta y el alcance del negocio. El análisis de estas variables permitirá determinar el éxito de la solución tecnológica y su impacto en la reducción de ineficiencias y la expansión del mercado.

Técnicas e Instrumentos de Recolección

Para la recolección de información se emplearán diversas técnicas que permitirán comprender la situación actual del negocio y definir los requerimientos de la plataforma de e-commerce. Se utilizará la observación directa de los procesos de venta que se llevan a cabo en el almacén, con el fin de identificar las deficiencias y necesidades operativas. Asimismo, se realizarán entrevistas estructuradas dirigidas al personal encargado de la administración y a los clientes potenciales, lo que permitirá obtener una perspectiva detallada de las necesidades y expectativas de los usuarios. También se aplicarán encuestas para conocer el grado de aceptación de la propuesta tecnológica. Finalmente, se procederá a la revisión documental de los registros de ventas e inventario existentes para cuantificar la situación actual.

En cuanto a los instrumentos, se diseñarán guías de observación para estructurar la información recolectada, cuestionarios de entrevistas para asegurar la consistencia en las conversaciones, y formularios de encuestas. Estos instrumentos garantizarán la obtención de información confiable y válida, la cual será indispensable para el análisis de la situación actual y el posterior desarrollo de la plataforma de e-commerce. La información recopilada servirá como la base fundamental para el desarrollo de la tienda en línea.

Metodología Scrum

El presente proyecto adopta la metodología ágil Scrum como marco de trabajo para la organización y ejecución de las actividades. Scrum es un enfoque iterativo e incremental que permite gestionar proyectos con alta flexibilidad, atendiendo a la naturaleza cambiante de los requerimientos y facilitando la entrega de valor en intervalos cortos de tiempo. Su implementación se basa en roles definidos, artefactos concretos y eventos que estructuran el flujo

del trabajo, promoviendo la autoorganización, la transparencia y la mejora continua (Sachdeva, 2016).

En el contexto del desarrollo de esta tesis, se decidió emplear Scrum de manera adaptada, teniendo en cuenta que el proyecto es llevado a cabo por una sola persona. Esto implicó que el autor asumiera los roles principales de Product Owner, Equipo de Desarrollo y Scrum Master, aplicando las prácticas y principios del marco de manera disciplinada, pero ajustada a las condiciones particulares del estudio.

Responsable del Producto (Product Owner)

El Product Owner es el rol central dentro de Scrum, responsable de definir la visión del producto, priorizar los requerimientos y asegurar que el proyecto aporte valor a los usuarios finales. En este trabajo, dicho rol es desempeñado por el autor de la tesis, quien establece los objetivos del sistema de inventario, construye y gestiona el Product Backlog, y toma decisiones estratégicas sobre qué funcionalidades se desarrollan primero y cuáles se ajustan en el transcurso del proyecto. Además, ejerce como mediador entre los intereses académicos, los requerimientos técnicos y la aplicabilidad práctica en la empresa, lo que garantiza la coherencia entre la solución planteada y las necesidades reales.

El Equipo de Desarrollo

En un proyecto Scrum convencional, el equipo de desarrollo está compuesto por profesionales de distintas áreas, como programadores, diseñadores, testers e ingenieros de sistemas. Sin embargo, en esta tesis el equipo de desarrollo recae en una sola persona: el autor. Esto implica que todas las funciones, desde la recolección de requisitos hasta la codificación, validación y documentación, son asumidas por el mismo responsable. A pesar de esta limitación,

la organización del trabajo se mantuvo fiel a los principios de Scrum, procurando entregar incrementos funcionales en cada sprint.

Scrum Master

El Scrum Master es la figura encargada de asegurar que se cumplan los valores y reglas de Scrum, facilitando la eliminación de obstáculos y promoviendo la mejora del proceso. En este caso, el autor asumió también dicho rol, velando por la correcta aplicación de la metodología, organizando los eventos Scrum y asegurando que las actividades se desarrollaran bajo un esquema de disciplina, autoevaluación y mejora continua.

El Product Backlog

El Product Backlog constituye el listado priorizado de requerimientos del sistema de inventario. En este proyecto fue elaborado en forma de historias de usuario, las cuales describen de manera clara las funcionalidades necesarias, como el registro de insumos, la gestión de entradas y salidas, la generación de reportes y el control de productos por fechas de caducidad. El Product Owner fue responsable de revisar y ajustar continuamente el backlog en función del progreso alcanzado y de los hallazgos durante el desarrollo.

Estimaciones Afinadas

Las estimaciones son fundamentales en Scrum para planificar de manera realista el trabajo de cada sprint. En el presente proyecto se realizaron de forma aproximada, tomando como referencia la complejidad percibida de cada historia de usuario y la experiencia previa del autor en el manejo de tareas similares. Aunque no se cuenta con un equipo para realizar estimaciones colectivas, la práctica de dividir el backlog en tareas pequeñas y alcanzables permitió afinar la carga de trabajo y avanzar de manera progresiva hacia los objetivos finales.

Los Sprints

El desarrollo del sistema se organizó en sprints de corta duración, cada uno con metas específicas orientadas a completar un conjunto funcional del producto. Esta planificación permitió verificar avances de manera continua y ajustar aspectos técnicos en función de los resultados obtenidos. Al finalizar cada sprint, se evaluaron los entregables para garantizar que respondieran a lo establecido en el backlog.

Reunión Diaria (Stand-Up)

En proyectos convencionales, la reunión diaria se utiliza para que los miembros del equipo compartan avances, dificultades y próximos pasos. En el caso de este trabajo, esta práctica fue adaptada como un ejercicio de autoevaluación diaria en el que el autor registró de manera sistemática lo realizado, lo pendiente y los obstáculos encontrados. Este hábito permite mantener la organización del trabajo y asegurar un avance constante en el cronograma establecido.

La Revisión del Sprint (Sprint Review)

Al final de cada sprint se realizó una revisión en la que el autor contrastó el producto desarrollado con los objetivos del backlog. Esta revisión funciona como un espacio de evaluación crítica de los resultados, en el cual se determinaron los ajustes necesarios y se identifican posibles mejoras. Aunque en un entorno empresarial este evento incluye a múltiples interesados, en este caso se aplican como un ejercicio reflexivo personal orientado a mantener la calidad y el cumplimiento de los objetivos del proyecto.

La Retrospectiva (Sprint Retrospective)

Finalmente, la retrospectiva se utiliza como un espacio para analizar lo que funcionó bien durante cada sprint, las dificultades encontradas y las oportunidades de mejora en los siguientes ciclos. Este proceso fue clave para optimizar la organización personal, mejorar la gestión del tiempo y ajustar estrategias que permitieran avanzar de manera más eficiente en el desarrollo del sistema

Cronograma de Actividades

Tabla 1

Cronograma de Actividades del Proyecto

Objetivo General: Desarrollar un sistema de gestión de ventas en línea (E-Commerce), para la empresa Casa Europea de Repuestos enfocado en la experiencia del cliente.			
Objetivo Especifico	Fase de la Metodología Scrum	Actividades	Producto
Identificar y documentar los requerimientos funcionales y no funcionales del sistema e-commerce	Sprint Planning / Product Backlog	Actividad 1: Recolección y análisis de información. Actividad 2: Identificación de criterios legales, técnicos y comerciales. Actividad 3: Priorización de requerimientos.	Documento de requerimientos validado
Diseñar la arquitectura del sistema, la interfaz gráfica (UI) y la experiencia de usuario (UX)	Sprint 1 – Diseño inicial	Actividad 1: Diseño de la arquitectura del sistema. Actividad 2: Prototipado de la interfaz gráfica. Actividad 3: Definición de experiencia de usuario (UX).	Diagramas de arquitectura y prototipo UI/UX
Desarrollar los módulos funcionales del sistema e-commerce	Sprint 2 – Desarrollo	Actividad 1: Programación del catálogo de productos. Actividad 2: Desarrollo del módulo administrativo e inventarios. Actividad 3: Integración de plataforma de pagos.	Módulos funcionales implementados

Ejecutar pruebas funcionales, de seguridad y usabilidad	Sprint 3 – Validación	Actividad 1: Pruebas unitarias. Actividad 2: Pruebas de seguridad. Actividad 3: Pruebas de usabilidad.	Informe de pruebas y validación del sistema
---------------------------------------------------------	-----------------------	--------------------------------------------------------------------------------------------------------------	---------------------------------------------

Nota. Esta tabla muestra el objetivo del proyecto, la fase y la actividad. *Fuente.* Autor

Requerimientos del Sistema

El desarrollo de sistemas de comercio electrónico exige una adecuada gestión de los requisitos, ya que de esta depende no solo la correcta funcionalidad de la plataforma, sino también su seguridad, fiabilidad y la experiencia de usuario. La ingeniería de requisitos se constituye, por tanto, en un proceso esencial que debe apoyarse en normas y estándares internacionales que permitan garantizar calidad desde la definición inicial de los requerimientos hasta la evaluación del producto final.

En este marco, el estándar ISO/IEC/IEEE 29148:2018 establece un conjunto de prácticas y lineamientos que abarcan todo el ciclo de vida de los requisitos, incluyendo su elicitación, análisis, documentación y gestión. Su aplicación en proyectos de comercio electrónico asegura que los requerimientos sean claros, verificables y trazables, lo cual reduce riesgos y mejora la alineación entre las expectativas de los clientes y las funcionalidades entregadas (Parra-Quero, Palma-Urdaneta, Torres-Samuel & Durán-Garrido, 2020).

Por su parte, la norma ISO/IEC 25010:2011 define un modelo de calidad de software estructurado en ocho características principales: funcionalidad, fiabilidad, usabilidad, eficiencia, compatibilidad, seguridad, mantenibilidad y portabilidad. Estas dimensiones permiten evaluar los productos desde diferentes perspectivas, garantizando que cumplan con los niveles de calidad esperados. En un sistema de comercio electrónico, la seguridad y la usabilidad resultan especialmente críticas, ya que influyen directamente en la confianza del cliente y en la facilidad de interacción con el sistema (Taica Sánchez, 2023).

De forma complementaria, Gordieiev, Kharchenko y Gordieieva (2022) proponen el modelo de calidad del perfil de requisitos (*Software Requirements Profile Quality Model*), que aporta un enfoque integral para evaluar no solo la calidad de cada requisito individual, sino también la consistencia del conjunto. Este modelo permite reducir ambigüedades y asegura que los requisitos puedan ser medidos y validados en función de criterios estandarizados, lo que incrementa la fiabilidad de la plataforma en contextos de alta demanda como el comercio electrónico.

Por otro lado, Calabrese, Muñoz, Pasini, Esponda, Boracchia y Pesado (2017) destacan la utilidad del enfoque GQM (*Goal, Question, Metric*) en la evaluación de los productos de software de acuerdo con la norma ISO/IEC 25010. Este enfoque consiste en definir objetivos de calidad, formular preguntas clave y asociar métricas concretas que permitan medir el grado de cumplimiento de dichos objetivos. Aplicado al comercio electrónico, posibilita una evaluación sistemática de subcaracterísticas como la confidencialidad, integridad y autenticidad, que son esenciales para la protección de datos de usuarios y transacciones.

Finalmente, los aportes de Taica Sánchez (2023) refuerzan la aplicabilidad de estas normas en contextos reales, al comparar la ISO/IEC 9126 con la ISO/IEC 25010 en una aplicación web del sector turismo. Sus resultados muestran que la norma ISO/IEC 25010 resulta más completa y adecuada para garantizar la calidad interna, externa y en uso de las aplicaciones, lo que evidencia su pertinencia para sistemas de comercio electrónico que requieren niveles superiores de seguridad, confiabilidad y satisfacción del usuario.

La integración de los estándares ISO/IEC/IEEE 29148:2018 e ISO/IEC 25010:2011 con modelos de calidad y enfoques de evaluación como los propuestos por Gordieiev et al. (2022),

Calabrese et al. (2017) y Taica Sánchez (2023), proporciona un marco sólido para gestionar y evaluar los requerimientos de los sistemas de comercio electrónico. Esto asegura que los productos desarrollados sean consistentes, escalables y centrados en las necesidades de los usuarios, contribuyendo a la sostenibilidad y éxito de las plataformas digitales.

Requerimientos funcionales

REQ-F01

Tabla 2

Requerimiento Funcional 1

# ID requerimiento	REQ-F01	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir el registro de clientes mediante formulario en línea, solicitando información básica como nombre, correo electrónico, contraseña y dirección de envío, para facturación.		
Justificación	El registro de usuarios es fundamental para personalizar la experiencia de compra y gestionar pedidos.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		

Verificación El usuario debe poder registrarse sus datos al sistema con los datos de facturación.

Nota. Esta tabla muestra el requerimiento funcional No 1. *Fuente.* Autor

REQ-F02

Tabla 3

Requerimiento Funcional 2

# ID requerimiento	REQ-F02	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir la autenticación de administradores registrados.		
Justificación	La autenticación protege el acceso a la información personal y permite el uso de funciones exclusivas como historial de pedidos.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Se valida el ingreso con credenciales correctas.		

Nota. Esta tabla muestra el requerimiento funcional No 2. *Fuente.* Autor

REQ-F03**Tabla 4***Requerimiento Funcional 3*

# ID requerimiento	REQ-F03	Tipo de requerimiento	Funcional
Descripción	El sistema debe gestionar un catálogo de productos organizado por categorías, permitiendo búsquedas y filtrado por nombre, precio o disponibilidad.		
Justificación	Facilita la navegación y localización de productos por parte de los usuarios.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Se prueban consultas y los resultados coinciden con los filtros aplicados.		

Nota. Esta tabla muestra el requerimiento funcional No 3. *Fuente.* Autor

REQ-F04

Tabla 5*Requerimiento Funcional 4*

# ID requerimiento	REQ-F04	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir agregar productos al carrito de compras, modificar cantidades y eliminar artículos antes de proceder al pago.		
Justificación	El carrito es esencial en un entorno de comercio electrónico para gestionar compras múltiples.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Se prueba la adición, modificación y eliminación de productos en el carrito.		

Nota. Esta tabla muestra el requerimiento funcional No 4. *Fuente.* Autor

REQ-F05

Tabla 6*Requerimiento Funcional 5*

# ID requerimiento	REQ-F05	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir a los clientes procesar sus compras mediante pasarela de pago, utilizando tarjetas de crédito/débito con uso de mercadopago.		
Justificación	Garantiza transacciones confiables y seguras, elemento central en un e-commerce.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Se confirma que el pago genera comprobante y actualiza el estado del pedido.		

Nota. Esta tabla muestra el requerimiento funcional No 5. *Fuente.* Autor

REQ-F06

Tabla 7*Requerimiento Funcional 6*

# ID requerimiento	REQ-F06	Tipo de requerimiento	Funcional
Descripción	El sistema debe enviar notificaciones por correo electrónico al cliente para confirmar el registro, la compra y el estado de envío del pedido.		
Justificación	Incrementa la confianza del usuario y mejora la comunicación con el cliente.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Media		
Verificación	Tras la compra, se verifica el envío automático del correo de confirmación.		

Nota. Esta tabla muestra el requerimiento funcional No 5. *Fuente.* Autor

REQ-F07

Tabla 8*Requerimiento funcional 7*

# ID requerimiento	REQ-F07	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir al administrador gestionar el inventario de productos, incluyendo altas, bajas y actualizaciones.		
Justificación	Permite mantener actualizado el catálogo y garantizar disponibilidad de productos.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Media		
Verificación	Se comprueba que los cambios del administrador se reflejen en el catálogo.		

Nota. Esta tabla muestra el requerimiento funcional No 7. *Fuente.* Autor

REQ-F08

Tabla 9*Requerimiento Funcional 8*

# ID requerimiento	REQ-F08	Tipo de requerimiento	Funcional
Descripción	El sistema debe permitir al administrador gestionar el inventario de productos, incluyendo altas, bajas y actualizaciones.		
Justificación	Permite mantener actualizado el catálogo y garantizar disponibilidad de productos.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Media		
Verificación	Se comprueba que los cambios del administrador se reflejen en el catálogo.		

Nota. Esta tabla muestra el requerimiento funcional No 8. *Fuente.* Autor

Requerimiento No Funcionales***REQ-NF01***

Tabla 10*Requerimiento no Funcional 1*

# ID requerimiento	REQ-NF01	Tipo de requerimiento	No Funcional
Descripción	El sistema debe garantizar la seguridad de la información mediante cifrado de datos sensibles.		
Justificación	La seguridad es prioritaria en un entorno de comercio electrónico para proteger datos financieros y personales.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Pruebas de seguridad y revisión de certificados activos.		

Nota. Esta tabla muestra el requerimiento no funcional No 1. *Fuente.* Autor

REQ-NF02

Tabla 11*Requerimiento no funcional 2*

# ID requerimiento	REQ-NF02	Tipo de requerimiento	No Funcional
Descripción	El sistema debe cumplir con el nivel A de accesibilidad de las Web Content Accessibility Guidelines (WCAG 2.1), reconocidas por el estándar ISO/IEC 40500:2012. Esto implica incluir texto alternativo en imágenes, y posibilidad de navegar únicamente mediante teclado.		
Justificación	Garantiza que la aplicación web cumpla con requisitos mínimos de accesibilidad reconocidos internacionalmente.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Pruebas en diferentes navegadores y dispositivos móviles.		

Nota. Esta tabla muestra el requerimiento no funcional No 2. *Fuente.* Autor

REQ-NF03

Tabla 12*Requerimiento no Funcional 3*

# ID requerimiento	REQ-NF03	Tipo de requerimiento	No Funcional
Descripción	El tiempo de respuesta del sistema no debe superar los 3 segundos en operaciones críticas como búsquedas inicios de sesión.		
Justificación	Asegura un rendimiento óptimo y mejora la satisfacción del usuario.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Pruebas de carga con diferentes volúmenes de usuarios.		

Nota. Esta tabla muestra el requerimiento no funcional No 3. *Fuente.* Autor

REQ-NF04

Tabla 13*Requerimiento no Funcional 4*

# ID requerimiento	REQ-NF04	Tipo de requerimiento	No Funcional
Descripción	El sistema debe cumplir con los principios de usabilidad establecidos en la norma ISO/IEC 25010, garantizando que al menos el 80 % de los usuarios seleccionados en pruebas de usabilidad puedan completar tareas críticas (registro, búsqueda de productos y compra) en un tiempo máximo definido, sin errores graves de navegación.		
Justificación	La usabilidad se evalúa mediante métricas objetivas que permitan medir la eficiencia, efectividad y satisfacción del usuario en la interacción con la aplicación web.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Pruebas de usabilidad con usuarios finales.		

Nota. Esta tabla muestra el requerimiento no funcional No 4. *Fuente.* Autor

REQ-NF05**Tabla 14***Requerimiento no Funcional 5*

# ID requerimiento	REQ-NF05	Tipo de requerimiento	No Funcional
Descripción	El sistema debe ser compatible y estar optimizado para su uso en al menos dos navegadores web estándar, específicamente Google Chrome y Mozilla Firefox, en sus versiones más recientes en el momento del despliegue del sistema.		
Justificación	Asegura la correcta visualización y funcionamiento del sistema en los navegadores más utilizados por los usuarios, garantizando compatibilidad multiplataforma.		
Autor		Versión	0.1
Fecha Creación	05/06/2025	Fecha Modificación	
Porcentaje implementación			
Origen	Documento de tesis		
Prioridad	Alta		
Verificación	Pruebas de funcionamiento en las versiones más recientes de Chrome y Firefox, verificando la correcta ejecución de las funcionalidades principales (registro, búsqueda, carrito y pago).		

Nota. Esta tabla muestra el requerimiento no funcional No 5. *Fuente.* Autor

Arquitectura del Sistema

La arquitectura del sistema de comercio electrónico para la empresa Casa Europea de Repuestos se fundamenta en el paradigma de microservicios, debido a las ventajas que este enfoque ofrece en términos de escalabilidad, resiliencia y facilidad de mantenimiento. A diferencia de las arquitecturas monolíticas tradicionales, los microservicios dividen la aplicación en componentes pequeños e independientes, cada uno encargado de una funcionalidad específica como la gestión de usuarios, el catálogo de productos, el carrito de compras, el procesamiento de pagos y la administración de inventarios. Esta separación permite que cada servicio sea desarrollado y desplegado de manera autónoma, lo que reduce la complejidad de la aplicación y aumenta la capacidad de respuesta ante cambios en la demanda (Ortiz, 2022).

En el caso del comercio electrónico, esta arquitectura resulta especialmente pertinente debido a la naturaleza dinámica de los procesos de compra y a la necesidad de garantizar disponibilidad continua en picos de demanda. Como destacan Quevedo-Ávila, Zhindón-Mora y Quevedo-Sacoto (2020), los microservicios en plataformas de compras en línea posibilitan una mayor flexibilidad al aislar funcionalidades críticas como el procesamiento de pagos y el manejo de pedidos, evitando que una falla en un módulo afecte la operación completa del sistema. Este enfoque no solo fortalece la resiliencia, sino que también facilita la integración de nuevas tecnologías y herramientas externas, como pasarelas de pago digitales o sistemas de gestión logística.

La propuesta arquitectónica se basa en un esquema de despliegue tradicional, en el cual el frontend desarrollado en React se conecta con un API Gateway implementado en .NET. Este Gateway centraliza la comunicación y enruta las solicitudes hacia los microservicios, que

también estarán desarrollados en .NET. Cada microservicio atiende un dominio funcional específico, tales como usuarios, catálogo de productos, pedidos, pagos e inventario. A su vez, los microservicios gestionan sus propios modelos de datos en una base de datos relacional MySQL, garantizando la consistencia y seguridad de la información almacenada.

En cuanto a la seguridad, cada microservicio incorporará mecanismos de autenticación y autorización mediante tokens JWT, además del uso de protocolos de comunicación segura como HTTPS. Estos elementos, en conjunto con la independencia de los servicios, fortalecen la confiabilidad del sistema y garantizan la protección de la información sensible de los usuarios.

En síntesis, la arquitectura de microservicios seleccionada para la aplicación web de Casa Europea de Repuestos se ajusta a las necesidades de flexibilidad, escalabilidad y seguridad propias del comercio electrónico. Su adopción garantiza la capacidad de evolución continua de la plataforma y la integración de nuevas funcionalidades sin comprometer la estabilidad del sistema, alineándose con las mejores prácticas y experiencias documentadas en investigaciones recientes (Ortiz, 2022; Quevedo-Ávila et al., 2020; Manobanda Natif, 2025).

Componentes de la Arquitectura

Frontend (React)

Es la capa de presentación que interactúa con el usuario final. Proporciona acceso al catálogo de productos, carrito de compras, pedidos y gestión de cuentas. Se comunica con el API Gateway a través de solicitudes REST.

API Gateway (.NET)

Actúa como intermediario entre el frontend y los microservicios. Su función es recibir las solicitudes del cliente, autenticarlas, aplicar reglas de enrutamiento y redirigirlas al microservicio correspondiente. También centraliza la seguridad mediante autenticación con tokens JWT y garantiza que solo los usuarios autorizados accedan a los recursos.

Microservicios (.NET)

Inventario. Administra la disponibilidad de los repuestos en el sistema. Se encarga de actualizar las existencias cuando se realiza una compra o se registra una entrada de productos en bodega.

Usuarios. Gestiona el registro, autenticación y administración de perfiles de clientes y administradores. Controla accesos mediante credenciales y emite tokens JWT para sesiones seguras.

Catálogo de Productos. Contiene la información de los productos disponibles (nombre, descripción, precio, categoría). Permite búsquedas y filtrados para optimizar la experiencia de compra.

Pedidos. Controla la creación y seguimiento de órdenes. Registra el detalle de cada compra, estado del pedido y actualiza el inventario en coordinación con otros microservicios.

Pagos. Procesa las transacciones económicas de manera segura. Se comunica con pasarelas de pago externas y registra la confirmación de las operaciones en la base de datos.

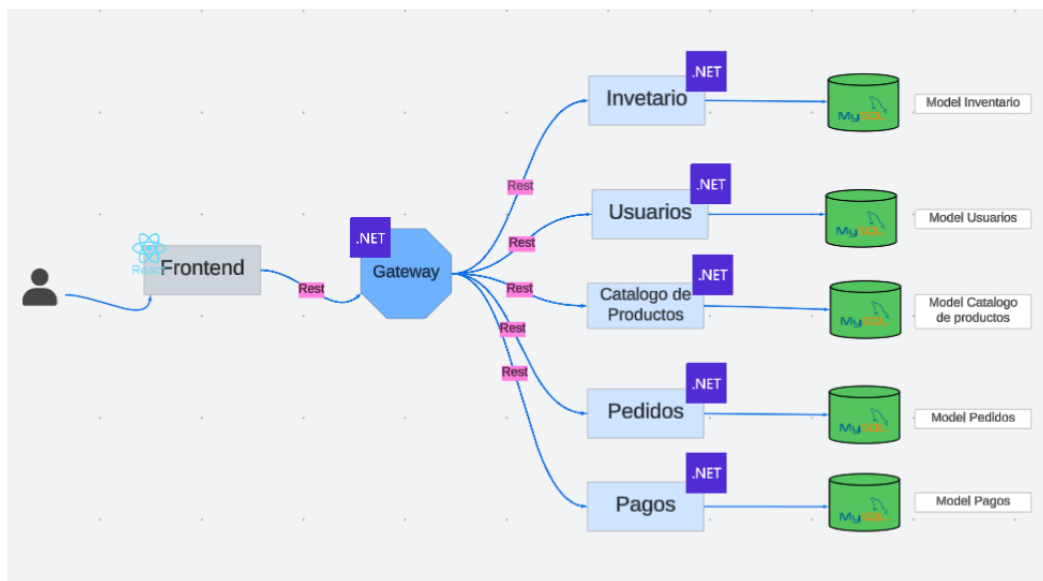
Base de datos (MySQL)

Cada microservicio tiene su propio modelo de datos dentro del sistema de gestión MySQL, lo que garantiza independencia lógica. Por ejemplo, el microservicio de usuarios gestiona la tabla de clientes, el de pedidos administra las órdenes y el de inventario controla las existencias. Esta segmentación de datos permite una mayor seguridad y reduce la dependencia entre servicios.

En contraste con lo anterior, la ilustración 1, muestra un sistema basado en una arquitectura de microservicios, en el cual el frontend (React) consume los servicios expuestos a través del Gateway (.NET). Cada microservicio opera de manera independiente y gestiona directamente su interacción con el modelo de datos correspondiente.

Figura 1

Arquitectura de Sistema



Nota. Diagrama de arquitectura del sistema de comercio electrónico E-commerce Casa Europea de Repuestos. Elaborado por el autor (2025).

Diseño del Sistema

Actores del Sistema

Tabla 15

Actores de sistema

Actor	Descripción	Rol en el sistema
Cliente	Usuario final de la aplicación web que busca, consulta y adquiere repuestos en línea. Puede registrarse, iniciar sesión, gestionar su carrito de compras y realizar pagos.	Consumidor principal, encargado de interactuar con el catálogo, los pedidos y el proceso de pago.
Administrador	Persona responsable de la gestión interna del sistema. Supervisa usuarios, productos, inventarios y pedidos, asegurando la correcta operación del e-commerce.	Gestor del sistema, con privilegios para modificar y controlar datos críticos.
Pasarela de pago	Sistema de servicio externo integrado que procesa transacciones económicas con medios digitales como tarjetas de crédito, débito o billeteras virtuales.	Medio de validación y autorización de las transacciones financieras.
Personal de logística	Persona encargada de recibir los reportes de los pedidos, iniciar la gestión de entrega y	Usuario de logística.

actualizar su estado, el cual puede ser: en alistamiento, enviado o entregado. Además, reporta el número de guía asignado por el transportador.

Nota. Esta tabla muestra los actores del sistema de comercio electrónico de Casar europea de Repuestos. *Obtenido de Autor*

Casos de Uso

Los casos de uso constituyen una herramienta fundamental para describir el comportamiento de un sistema desde la perspectiva de los usuarios. A través de ellos, se definen las interacciones entre los actores y el sistema, representando escenarios que buscan alcanzar una meta específica del usuario. Según Canchala et al. (2004), un caso de uso es una narración estructurada que describe cómo los actores externos interactúan con el sistema para lograr un objetivo determinado.

El modelado de casos de uso permite establecer los límites del sistema, identificar a los actores principales y detallar las funcionalidades clave que este debe ofrecer. Asimismo, proporciona una visión clara del alcance del software y facilita la comunicación entre los desarrolladores y los usuarios finales, lo cual resulta esencial en proyectos de comercio electrónico, donde convergen distintos roles como clientes, administradores, sistemas de pago y servicios de logística.

En el caso del sistema de comercio electrónico para la empresa Casa Europea de Repuestos, los casos de uso describen las principales operaciones que deben realizar tanto los

clientes como los administradores, considerando también la interacción con servicios externos. Estas operaciones incluyen el registro y autenticación de usuarios, la consulta y gestión del catálogo de productos, la administración del carrito de compras, el procesamiento de pedidos y la realización de pagos en línea.

El desarrollo de los casos de uso no se limita a los diagramas gráficos, sino que también implica la construcción de descripciones textuales donde se especifican los actores involucrados, las precondiciones, el flujo principal de eventos, los flujos alternativos y las postcondiciones. Este nivel de detalle garantiza que cada caso de uso represente un requerimiento funcional concreto, en coherencia con los lineamientos metodológicos de la ingeniería de software (Canchala et al., 2004).

Casos de Uso General

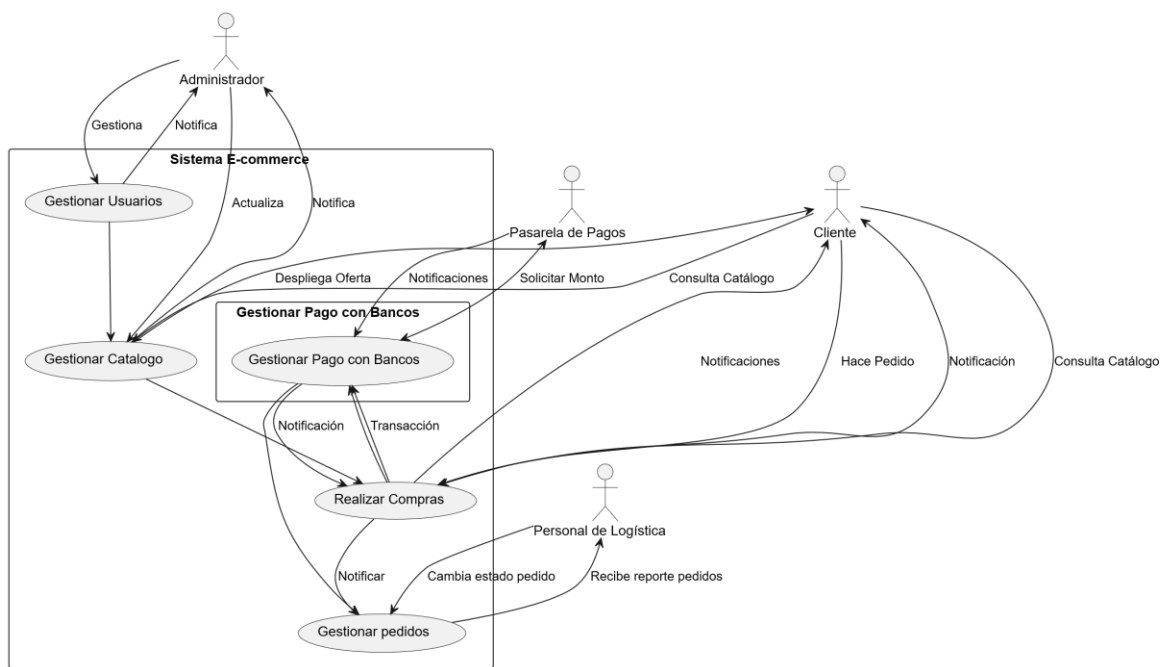
El diagrama de caso de uso general integra todos estos elementos en una vista unificada, proporcionando un esquema gráfico del funcionamiento global del sistema y constituyendo la base para la descomposición en casos de uso individuales como se muestra en la ilustración 2. La descripción de caso de uso general es el siguiente:

El sistema de comercio electrónico contempla diversos casos de uso fundamentales. En primer lugar, la gestión de usuarios, a cargo del administrador, permite crear, actualizar, consultar o eliminar registros, mostrando notificaciones de confirmación o error según la acción ejecutada. La gestión del catálogo involucra tanto al administrador, quien puede dar de alta o baja productos, modificar precios, descripciones o imágenes, como al cliente, que consulta el catálogo para visualizar la oferta disponible. En cuanto a la realización de compras, el cliente selecciona los productos y genera un pedido, mientras el sistema se conecta con la pasarela de pagos para

procesar la transacción y comunicarle el resultado. La gestión de pagos con bancos, liderada por la pasarela de pagos, valida la operación, solicita el monto, notifica éxito o fallo y confirma la transacción al sistema de compras, en interacción con el cliente. Finalmente, la gestión de pedidos, en la que interviene el personal de logística, permite recibir los reportes de pedidos, actualizar su estado (alistamiento, envío, entrega) y generar el número de guía asignado por el transportador, manteniendo informado al cliente en cada etapa del proceso.

Figura 2

Caso de Uso General



Nota. Diagrama de caso de uso general para el sistema de comercio electrónico E-commerce Casa Europea de Repuestos. Elaborado por el autor (2025).

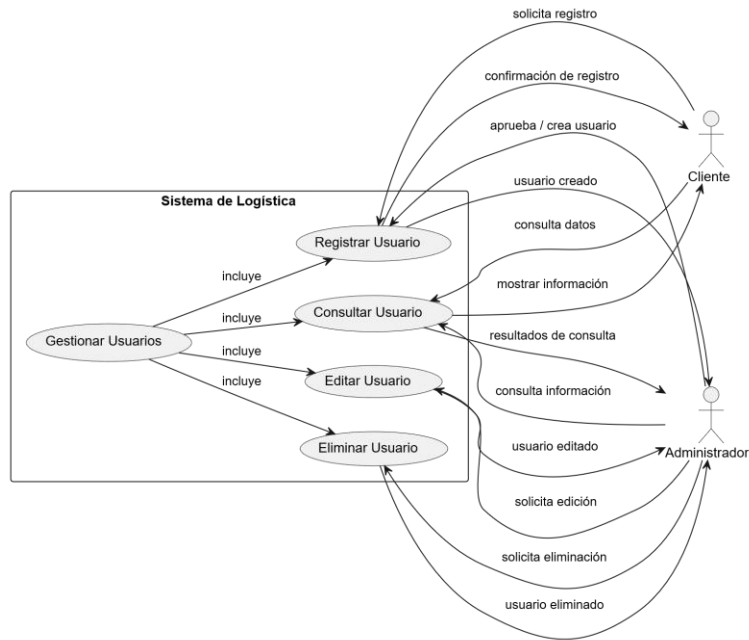
Casos de Uso Individual

CU01 – Registro de Datos usuarios

El caso de uso Gestionar Usuarios contempla cuatro subprocesos principales. En primer lugar, Registrar Usuario, que puede ser ejecutado tanto por el cliente como por el administrador, permite ingresar datos personales, de contacto y credenciales de acceso para crear un nuevo usuario, tras lo cual el sistema valida la información, almacena el registro y notifica su creación exitosa. En segundo lugar, Editar Usuario, acción exclusiva del administrador, posibilita la actualización de información como nombre, correo, rol o permisos, guardando los cambios y confirmando la modificación. En tercer lugar, Eliminar Usuario, también a cargo del administrador, consiste en la eliminación de un usuario previa confirmación, dejando el registro inactivo o borrado en la base de datos y notificando la acción realizada. Finalmente, Consultar Usuario, disponible tanto para clientes como para administradores, permite buscar y visualizar información de usuarios registrados, mostrando datos relevantes como perfil, estado e historial. En la ilustración 3 se representa este caso de uso descrito.

Figura 3

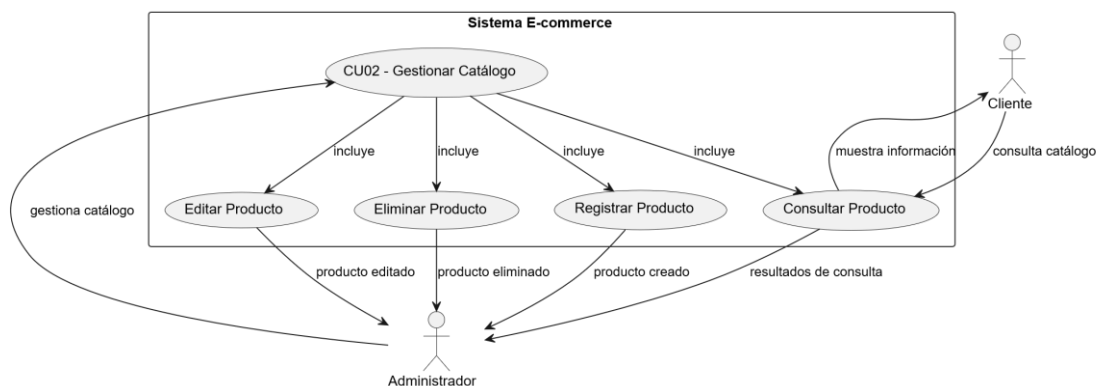
CU01 – Registro de datos Usuarios



Nota. Diagrama de caso de uso individual. Elaborado por el autor (2025).

CU02 – Gestionar Catálogo

El diagrama de caso de uso de gestión de catálogo muestra la interacción entre el administrador y el cliente con el sistema de comercio electrónico. El administrador realiza tareas de mantenimiento del catálogo, como registrar, editar, consultar y eliminar productos, mientras que el cliente accede a la información disponible a través de la consulta del catálogo. De esta manera, se asegura que la información ofrecida a los usuarios esté actualizada y organizada, lo cual es fundamental para la toma de decisiones de compra. Este proceso se encuentra representado en la Ilustración 4.

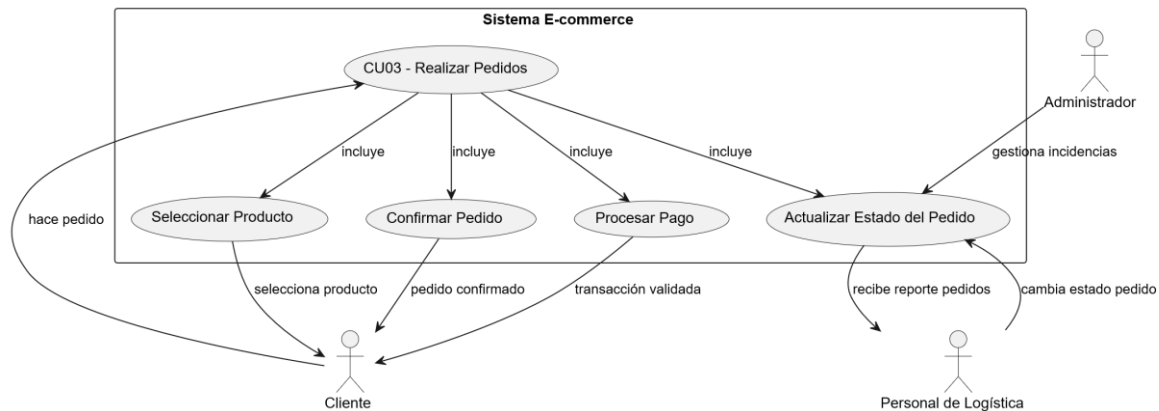
Figura 4**CU02 – Gestionar Catálogo**

Nota. Diagrama de caso de uso individual. Elaborado por el autor (2025).

CU03 – Realizar Compras

El diagrama de caso de uso de realizar pedidos representa el proceso mediante el cual el cliente interactúa con el sistema de comercio electrónico para efectuar una compra. El flujo inicia con la selección de productos, que posteriormente son confirmados por el sistema como parte del pedido. Una vez validada la transacción a través del procesamiento del pago, el pedido queda registrado. En la etapa final, el personal de logística recibe el reporte correspondiente y actualiza el estado del pedido, asegurando la trazabilidad del proceso de compra. Por su parte, el administrador tiene la función de gestionar incidencias relacionadas con los pedidos, lo que garantiza el control y seguimiento de las operaciones. Este proceso integral asegura que los pedidos pasen por todas las fases necesarias, desde la selección hasta la entrega, como se muestra en la Ilustración 5.

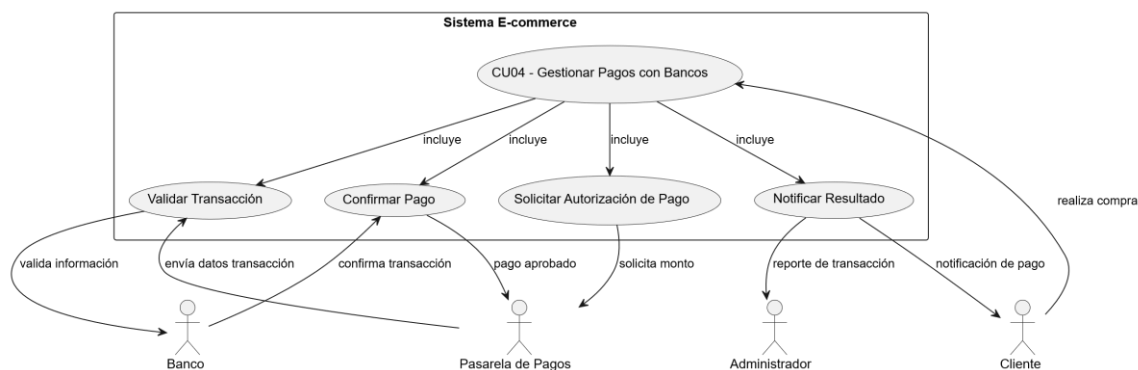
Figura 5

CU03 – Realizar Compras

Nota. Diagrama de caso de uso individual. Elaborado por el autor (2025).

CU04 – Gestión Pagos con Bancos

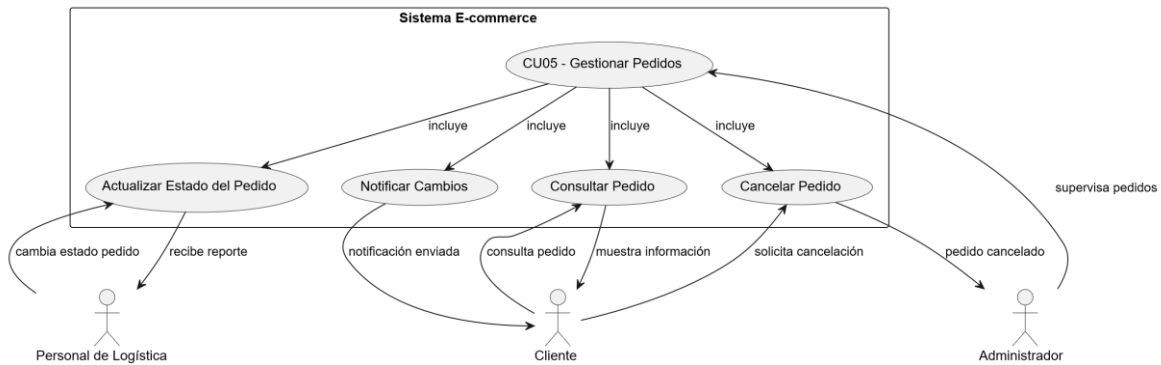
El diagrama de caso de uso de gestión de pagos con bancos ilustra el proceso mediante el cual el cliente, al realizar una compra en el sistema de comercio electrónico, inicia la solicitud de autorización de pago a través de la pasarela de pagos. Esta, a su vez, envía los datos de la transacción al banco, que valida la información y confirma o rechaza la operación. Una vez confirmada la transacción, el sistema emite la aprobación del pago y notifica tanto al cliente como al administrador. El cliente recibe la confirmación de su pago, lo que le garantiza la continuidad del proceso de compra, mientras que el administrador obtiene un reporte de la transacción, lo que permite llevar un control interno sobre las operaciones financieras del sistema. De esta manera, se asegura la integridad del proceso de pago, como se observa en la Ilustración 6.

Figura 6**CU04 – Gestión Pagos con Bancos**

Nota. Diagrama de caso de uso individual. Elaborado por el autor (2025).

CU05 – Gestionar Pedidos

El diagrama de caso de uso denominado CU05, ilustración 7, – Gestionar Pedidos representa las interacciones entre los diferentes actores del sistema de comercio electrónico y las funcionalidades relacionadas con la administración de los pedidos. En este proceso participan tres actores principales: el cliente, el administrador y el personal de logística. El cliente tiene la posibilidad de consultar el estado de sus pedidos, solicitar la cancelación y recibir notificaciones sobre los cambios realizados. El administrador se encarga de supervisar el proceso de gestión y validar la cancelación de pedidos cuando corresponde. Por su parte, el personal de logística es responsable de actualizar el estado de los pedidos en cada etapa, como preparación, envío o entrega, garantizando la trazabilidad del proceso. El sistema incluye funcionalidades específicas como consultar pedido, actualizar estado del pedido, cancelar pedido y notificar cambios, que en conjunto permiten dar un seguimiento eficiente y transparente a las solicitudes de los clientes y mantener el control administrativo de las operaciones.

Figura 7**CU05 – Gestionar Pedidos**

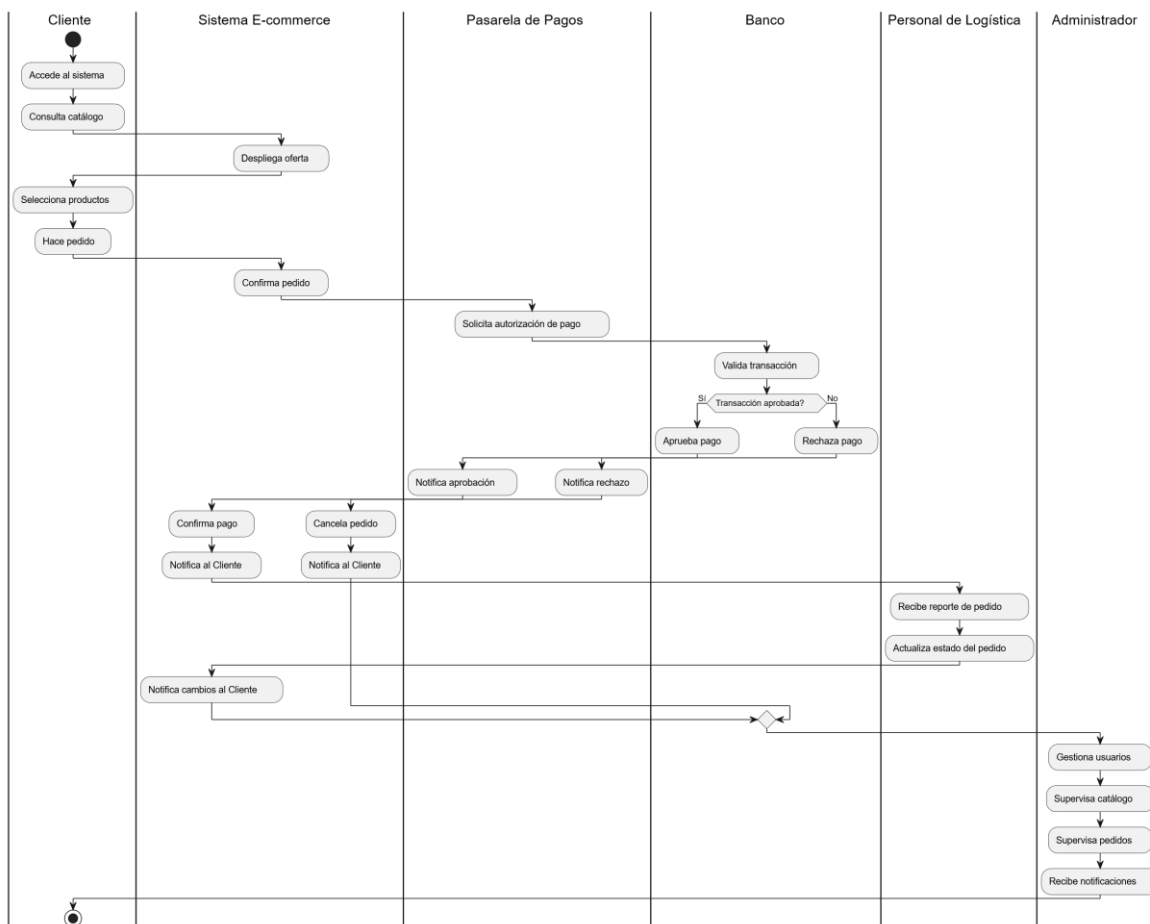
Nota. Diagrama de caso de uso individual. Elaborado por el autor (2025).

Diagrama de Actividades

El diagrama de actividades, ilustración 8, describe el flujo general que sigue un cliente dentro del sistema de comercio electrónico. Inicia con el acceso al sistema, donde el usuario puede registrarse o autenticarse según su estado. Posteriormente, se representa la consulta del catálogo, la selección de productos y la gestión del carrito de compras. El proceso incluye la confirmación del pedido, el pago en línea y la validación de la transacción mediante la pasarela de pagos. En caso de éxito, el sistema confirma el pedido, actualiza el inventario, envía la orden a logística y notifica al cliente. En caso contrario, se notifica un error de pago. Este diagrama proporciona una visión estructurada del flujo de trabajo del cliente, integrando las diferentes interacciones clave hasta completar la compra, como se observa en la ilustración correspondiente.

Figura 8

Diagrama de Actividades



Nota. Diagrama de actividades para el comercio electrónico. Elaborado por el autor (2025).

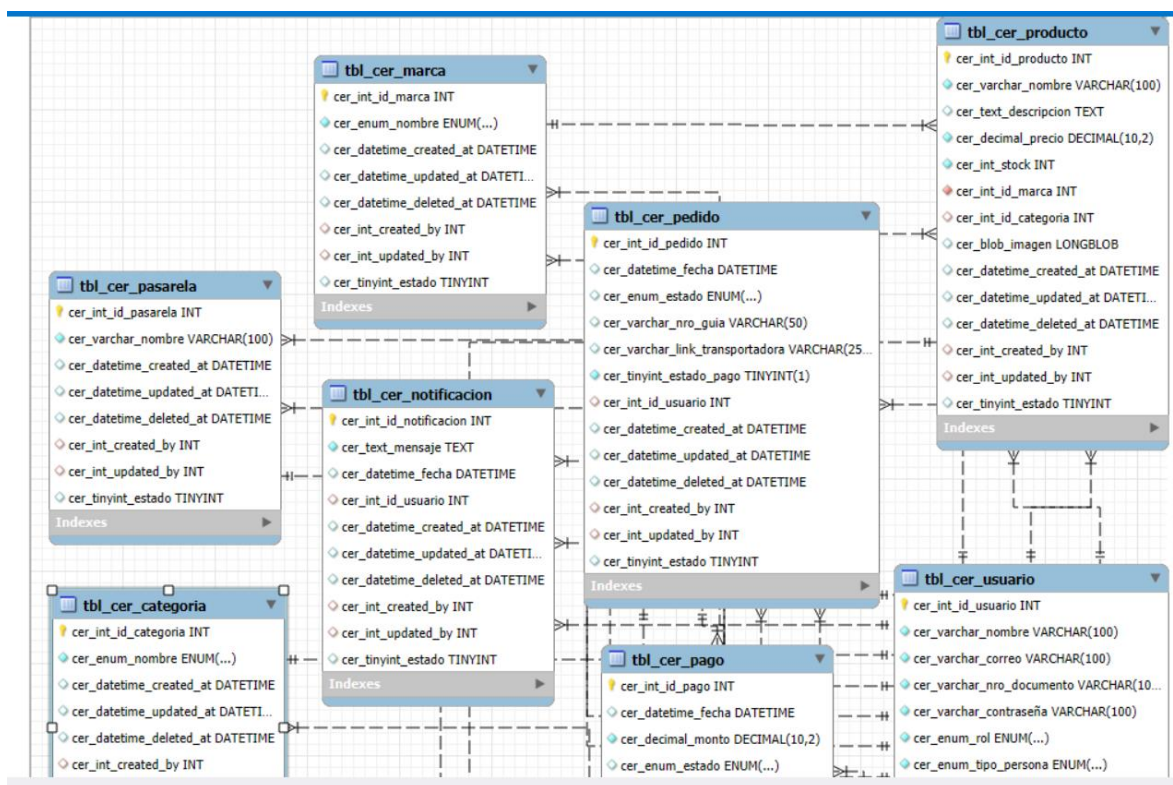
Modelo Entidad Relación

El modelo entidad-relación, ilustración 9, representa la estructura lógica de la base de datos que soporta el funcionamiento del sistema. En este modelo se identifican las entidades principales, tales como usuarios, productos, categorías, carritos de compras, pedidos, pagos e inventario. Se observa que los usuarios pueden poseer uno o varios carritos y realizar múltiples pedidos, los cuales se descomponen en detalles asociados a los productos seleccionados. Cada pedido se vincula de manera única con un pago, garantizando el registro de las transacciones

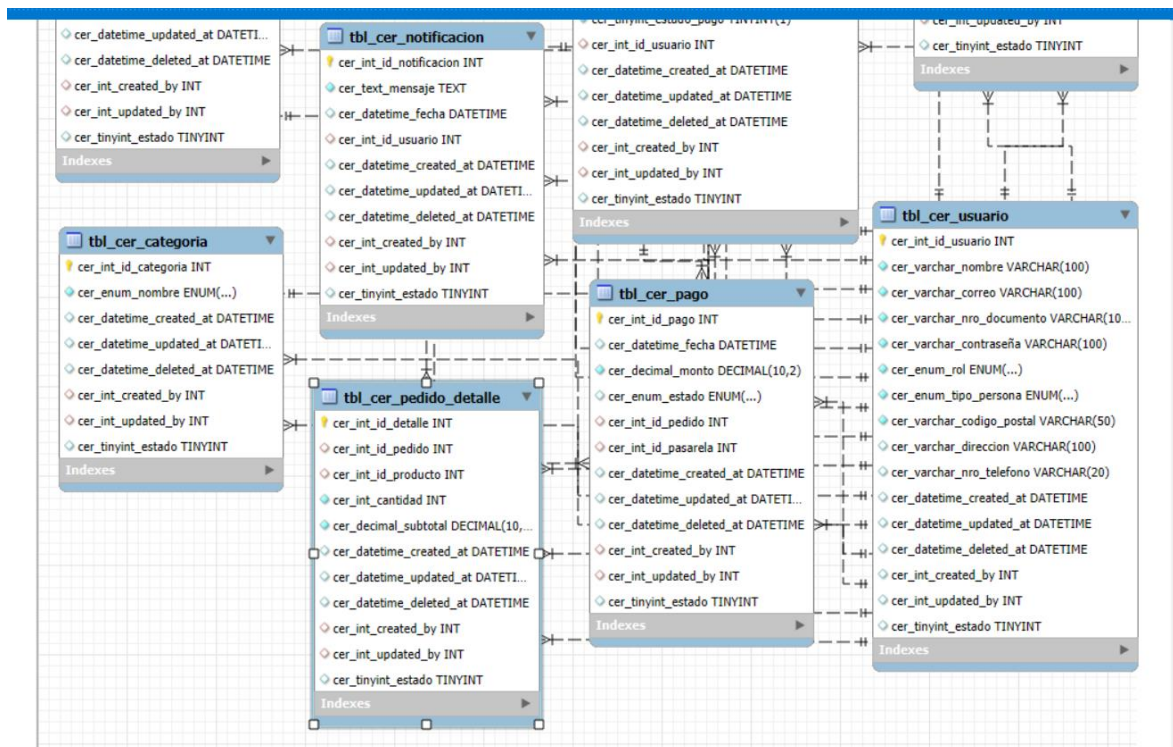
financieras. Asimismo, los productos se relacionan con categorías y generan movimientos de inventario que permiten el control de existencias. Este modelo proporciona una visión clara de la estructura de datos y establece las bases para la integridad, consistencia y escalabilidad de la aplicación web, como se muestra en la ilustración correspondiente.

Figura 9

Modelo Entidad Relación



Nota. Diagrama de modelo entidad relacion para el comercio electrónico. Elaborado por el autor (2025).



Nota. Modelo entidad relación del comercio electrónico. Elaborado por el autor (2025).

MockUp

El uso de mockups en el desarrollo de sistemas resulta fundamental, ya que permite representar de manera visual y temprana la estructura y funcionamiento de una aplicación antes de su implementación técnica. Estos modelos gráficos facilitan la comunicación entre desarrolladores, diseñadores y clientes, reduciendo ambigüedades en los requerimientos y optimizando el tiempo de desarrollo al detectar inconsistencias en fases iniciales. Asimismo, los mockups favorecen la validación temprana de ideas, asegurando que la solución propuesta responda a las necesidades reales de los usuarios.

En este sentido, Bañares Colastra (2016) destaca que las herramientas de prototipado con mockups proporcionan un entorno accesible tanto para desarrolladores como para usuarios sin conocimientos avanzados de programación, ya que permiten transformar ideas en interfaces

visuales intuitivas y funcionales. Dichas representaciones no solo contribuyen a la comprensión global del sistema, sino que también apoyan el análisis, el diseño y la planificación de la aplicación móvil o web, convirtiéndose en un puente entre la concepción del proyecto y su implementación final.

En conclusión, los mockups son una herramienta estratégica dentro del ciclo de desarrollo de software, pues garantizan una mejor alineación entre los objetivos del proyecto y las expectativas del usuario, minimizando riesgos y costos asociados a modificaciones posteriores.

Página Principal “Casa Europea de Repuestos”

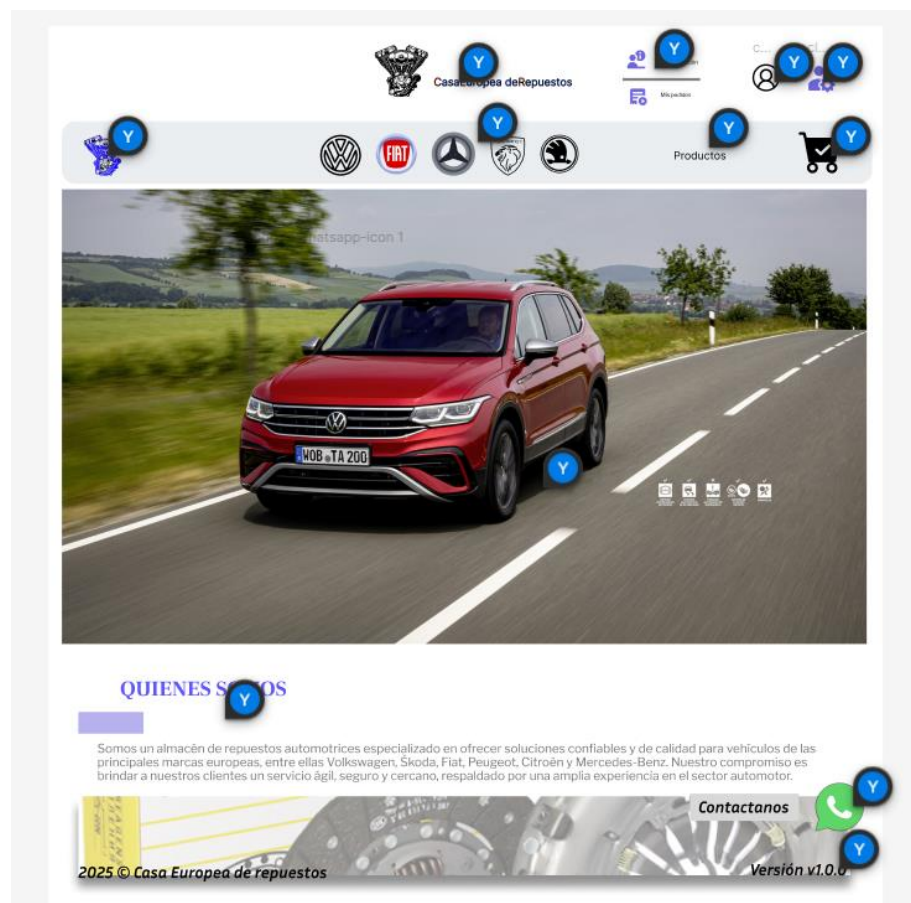
La ilustración 10, muestra el diseño de la página principal del comercio electrónico Casa Europea de Repuestos, en donde se integran elementos clave para la navegación y la experiencia del usuario. En la parte superior se ubica el logo de la empresa junto con el botón que redirige al inicio, permitiendo al usuario regresar fácilmente a la página principal desde cualquier sección. Al costado, se presentan botones de acceso al perfil del cliente, que aparecen cuando el usuario ha iniciado sesión, además de un botón para acceder al módulo de administración.

En el área de navegación principal se encuentran los botones de filtrado por marca, que permiten explorar productos específicos de fabricantes europeos como Volkswagen, Fiat, Mercedes-Benz, Peugeot y otros. Junto a estos, se incluye un botón de catálogo general de productos, acompañado del icono del carrito de compras, el cual despliega un menú horizontal con vista previa de los pedidos seleccionados. Por otro lado, en la sección central destaca un carrusel de imágenes, en este caso mostrando un vehículo Volkswagen, que refuerza la identidad de la tienda como especialista en repuestos europeos.

Más abajo, se incluye la sección “Quiénes Somos”, donde se presenta una breve descripción del almacén y su compromiso con la calidad, la seguridad y la experiencia en el sector automotor; y finalmente, en la parte inferior derecha, se observa el botón de contacto por WhatsApp, que facilita la comunicación directa con la empresa. Además, se muestra la versión actual de la página (v1.0.0) y el aviso de derechos reservados correspondiente al año 2025.

Figura 10

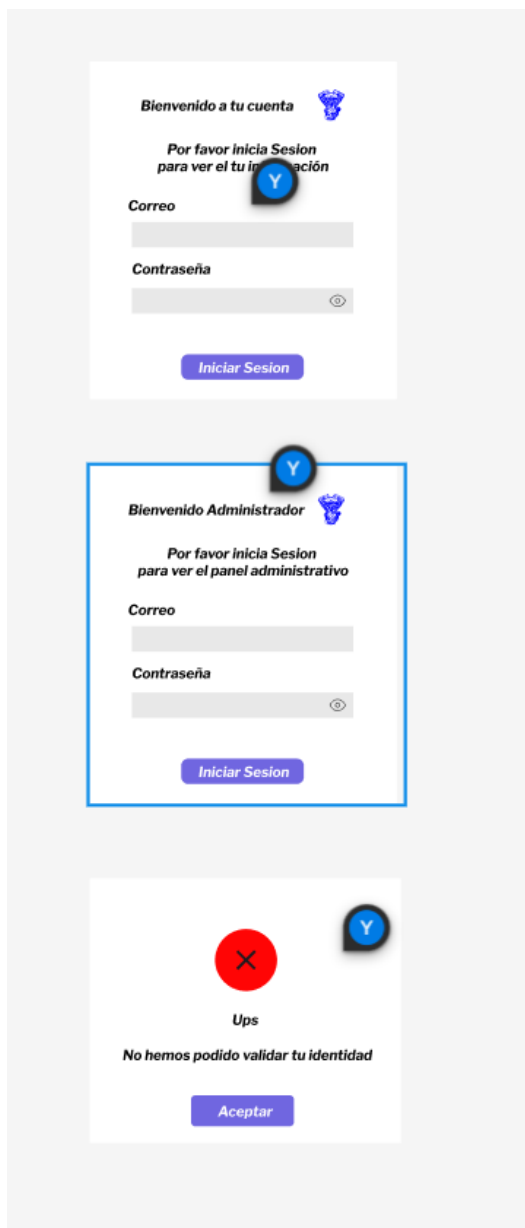
Página Principal Casa Europea de Repuestos



Nota. Home Page comercio electrónico E-commerce Casa Europea de Repuestos. Elaborado por el autor (2025).

La ilustración 11, presenta el diseño del módulo de inicio de sesión para usuarios y administradores en el sistema de Casa Europea de Repuestos, así como el mensaje de error en caso de credenciales inválidas. En el primer cuadro se observa el formulario de acceso para clientes, el cual incluye los campos de correo y contraseña, junto con un botón para mostrar u ocultar la contraseña. El botón inferior permite iniciar sesión y acceder a la cuenta del usuario, donde podrá consultar su información y realizar gestiones dentro del sistema.

En el segundo cuadro se muestra el formulario de acceso exclusivo para administradores. Su diseño es similar al del cliente, con los campos de correo y contraseña, pero el mensaje introductorio especifica que el acceso está destinado al panel administrativo, donde se gestionan los módulos internos del sistema. Finalmente, en el tercer cuadro se encuentra la ventana emergente de error, la cual aparece cuando las credenciales ingresadas no son correctas o no han podido ser validadas. Esta notificación incluye un ícono de advertencia en color rojo, el mensaje “Ups, no hemos podido validar tu identidad” y un botón de aceptación para cerrar el aviso e intentar nuevamente el inicio de sesión.

Figura 11*Modales Inicio de Sesión*

Nota. Modales de inicio de sesión para cliente y administradores. Elaborado por el autor (2025).

Módulo Administrador “Casa Europea de Repuestos”

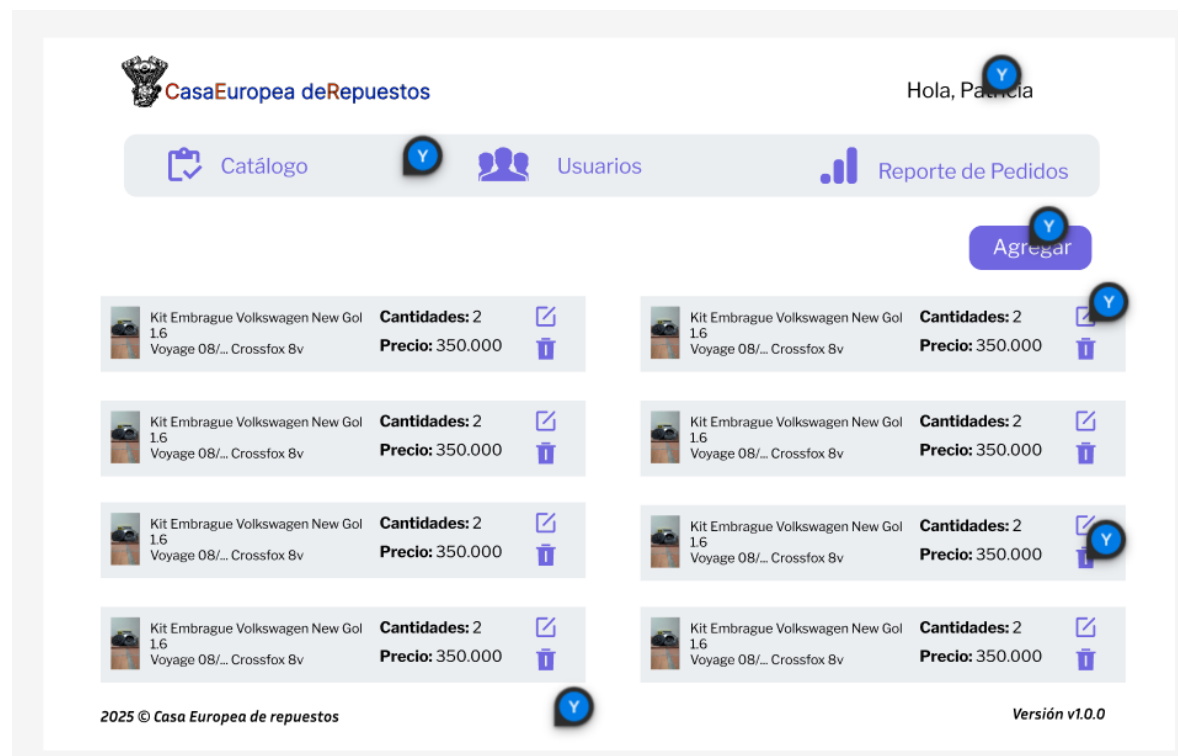
La ilustración 12, presenta el diseño del módulo de inventario en el sistema de Casa Europea de Repuestos, el cual permite la visualización, gestión y control de productos

registrados. En la parte superior se observa el menú de navegación con tres apartados principales: Catálogo, Usuarios y Reporte de pedidos, que facilitan el acceso a los diferentes módulos del sistema. A la derecha se muestra un saludo personalizado al usuario en sesión, lo que permite identificar quién está utilizando la plataforma.

En el centro de la pantalla se encuentra el botón “Agregar”, destinado al registro de nuevos productos en el inventario. La sección principal muestra una lista de productos disponibles. Cada producto cuenta con su respectiva información más relevante. Asimismo, cada tarjeta de producto incorpora dos iconos de acción: uno para editar la información y otro para eliminar el artículo, lo que brinda al administrador un control directo y sencillo sobre el inventario.

Figura 12

Módulo Catálogo de Producto



Nota. Módulo catálogo de Casa Europea de Repuestos. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño de la vista para la creación de productos dentro del módulo de inventario del sistema de Casa Europea de Repuestos. En la parte izquierda de la interfaz se dispone un recuadro destinado a la carga de imágenes, lo cual permite añadir una representación visual del producto a registrar. En el área central y derecha se encuentran los campos de entrada de datos: el nombre del producto, el precio unitario y la cantidad disponible. Adicionalmente, se incluye un campo de texto para ingresar una descripción detallada, que facilita la identificación y clasificación del artículo dentro del inventario.

Más abajo, se despliegan opciones de selección para categoría y marca, que permiten filtrar y organizar los productos según su clasificación. Entre las marcas disponibles se encuentran Skoda, Fiat, Volkswagen, Mercedes Benz y Peugeot, lo que refleja el enfoque especializado de la empresa en repuestos de origen europeo. Finalmente, en la parte inferior derecha se ubica el botón “Crear”, el cual confirma el registro de un nuevo producto dentro del sistema.

Figura 13

Modal Creación de Producto

2025 © Casa Europea de repuestos

Versión v1.0.0

Nota. Modal creación producto. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño del módulo de gestión de usuarios del sistema de Casa Europea de Repuestos. En la parte superior de la interfaz se mantiene el menú de navegación con tres apartados principales: Inventario, Usuarios y Reporte de pedidos, lo que permite desplazarse fácilmente entre los módulos del sistema.

En el centro de la pantalla se encuentra una tabla que organiza la información de los usuarios registrados en el sistema. Esta tabla incluye los campos Nro. Cédula, Nombre, Estado, Cargo y Acciones. En la columna de Acciones, se disponen dos iconos: el primero permite editar la información del usuario, mientras que el segundo posibilita eliminarlo del sistema. De este modo, el administrador puede gestionar de manera eficiente el acceso y los roles de cada integrante dentro de la plataforma.

Figura 14

Módulo Usuarios

The screenshot displays the user management interface. At the top left is the logo for 'Casa Europea de Repuestos'. To the right, it says 'Hola, Patricia'. Below this is a navigation bar with three items: 'Inventario' (with a clipboard icon), 'Usuarios' (with a group of people icon), and 'Reporte de Pedidos' (with a bar chart icon). A blue button labeled 'Agregar' with a plus icon is positioned above the table. The table has five columns: 'Nro. Cedula', 'Nombre', 'Estado', 'Cargo', and 'Acciones'. The first row of data shows '1.023.456', 'Patricia Huertas', 'Activo', 'Administrador', and two icons (edit and delete). At the bottom left, it says '2025 © Casa Europea de repuestos' and at the bottom right, 'Versión v1.0.0'.

Nro. Cedula	Nombre	Estado	Cargo	Acciones
1.023.456	Patricia Huertas	Activo	Administrador	

Nota. Módulo catálogo de Casa Europea de Repuestos. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño de la vista para la creación de usuarios dentro del sistema de Casa Europea de Repuestos. En la parte central se encuentran los campos de entrada correspondientes a los datos generales del nuevo usuario: número de cédula, nombre, correo y contraseña. Estos elementos permiten registrar la información básica necesaria para el acceso al sistema.

En la parte inferior se incluye la opción de asignar un cargo al usuario, con dos alternativas disponibles: administrador o logística. Esta funcionalidad facilita la definición de roles y responsabilidades dentro de la plataforma, asegurando que cada persona cuente con los permisos adecuados según sus funciones.

Finalmente, en la parte inferior derecha se ubica el botón guardar, que confirma el registro de un nuevo usuario en el sistema. De este modo, se garantiza un control organizado y seguro de las cuentas que tienen acceso a la aplicación.

Figura 15

Creación de Usuario

Datos Generales

Nro de Cedula

Nombre

Correo

Contraseña

Cargo

Administrador

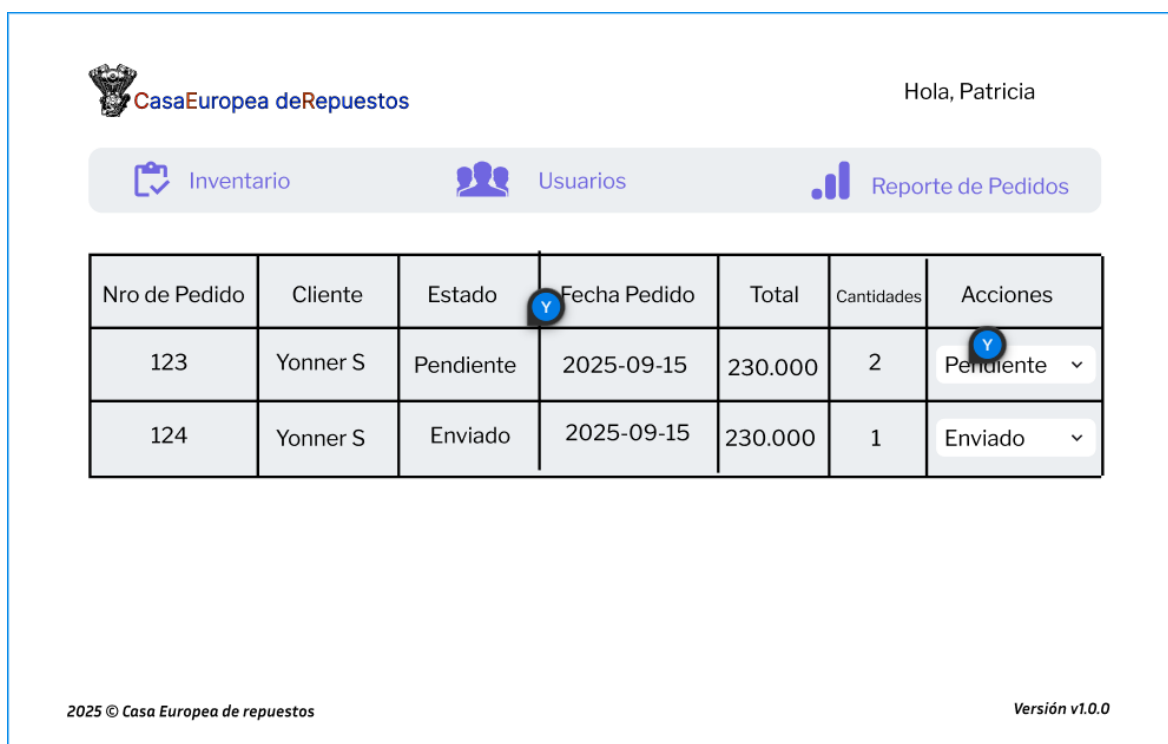
Logística

Nota. Modal creación de usuario. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño del módulo de reporte de pedidos en el sistema de Casa Europea de Repuestos. En el centro de la pantalla se encuentra una tabla que organiza los pedidos registrados. Cada fila muestra información relevante, como el número de pedido, cliente, estado, fecha del pedido, valor total, cantidades y acciones disponibles. En la columna de acciones, el sistema ofrece un menú desplegable que permite cambiar el estado del pedido, ya sea pendiente o enviado. Esto otorga flexibilidad al administrador para actualizar el progreso de las órdenes y mantener un control en tiempo real de la gestión logística.

Figura 16

Módulo Reporte de Pedidos



Nro de Pedido	Cliente	Estado	Fecha Pedido	Total	Cantidades	Acciones
123	Yonner S	Pendiente	2025-09-15	230.000	2	Pendiente ▾
124	Yonner S	Enviado	2025-09-15	230.000	1	Enviado ▾

2025 © Casa Europea de repuestos Versión v1.0.0

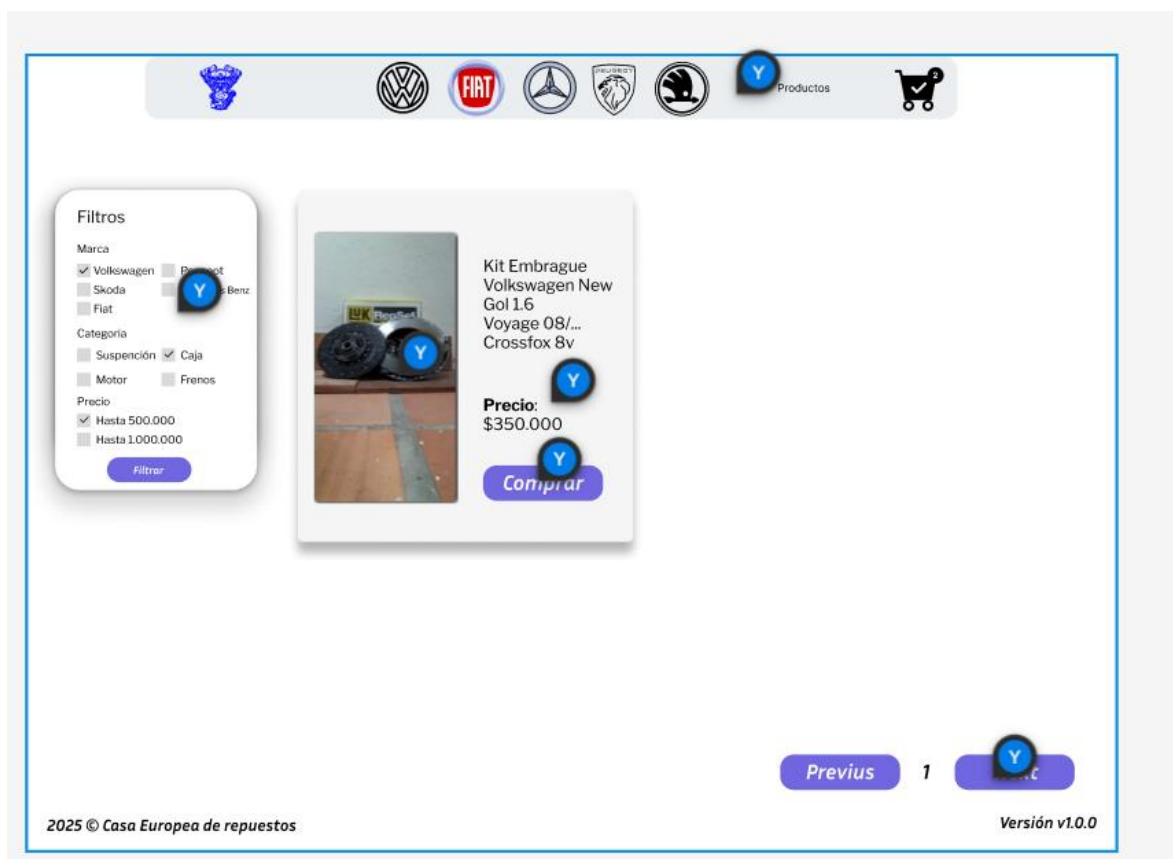
Nota. Módulo reporte de pedidos de Casa Europea de Repuestos. Elaborado por el autor (2025).

Catálogo Productos “Casa Europea de Repuestos”

La siguiente ilustración presenta el diseño del módulo de catálogo de productos en el sistema de Casa Europea de Repuestos. En la parte superior se ubica una barra de navegación con los logotipos de diferentes marcas de automóviles europeos, entre ellos Volkswagen, Fiat, Mercedes Benz, Peugeot y Skoda, lo que facilita el acceso rápido a los productos específicos de cada fabricante. A la derecha se encuentran los apartados de productos y el ícono del carrito de compras, que permite visualizar los artículos seleccionados por el cliente.

En el costado izquierdo de la vista se encuentra un panel de filtros que ofrece diversas opciones de búsqueda. Estos filtros incluyen la marca del vehículo, la categoría del repuesto (suspensión, caja, motor o frenos) y el rango de precios, con dos intervalos predefinidos: hasta 500.000 y hasta 1.000.000. Esta funcionalidad contribuye a una navegación más precisa y personalizada dentro del catálogo.

En el centro de la pantalla se presenta la tarjeta de un producto disponible. La tarjeta incluye la imagen del producto, una breve descripción y el precio de venta establecido en 350.000 pesos colombianos. En la parte inferior de la tarjeta se encuentra el botón comprar, que permite al usuario añadir el artículo al carrito y continuar con el proceso de adquisición. Finalmente, en la parte inferior de la vista se muestra un sistema de paginación, que facilita la navegación entre diferentes páginas del catálogo.

Figura 17*Módulo Catálogo de productos*

Nota. Módulo de catálogo de productos del sistema de Casa Europea de Repuestos. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño de la tarjeta de detalles del producto en el sistema de Casa Europea de Repuestos. En la parte superior izquierda se indica la marca del vehículo asociada, en este caso Volkswagen, y en la parte superior derecha se encuentra un icono en forma de “X” que permite cerrar la ventana emergente.

En la sección central se muestran los elementos principales del producto seleccionado: una imagen ilustrativa, el nombre y descripción breve, junto con el precio establecido de pesos colombianos. Debajo del valor se incluyen los controles de cantidad, representados con los

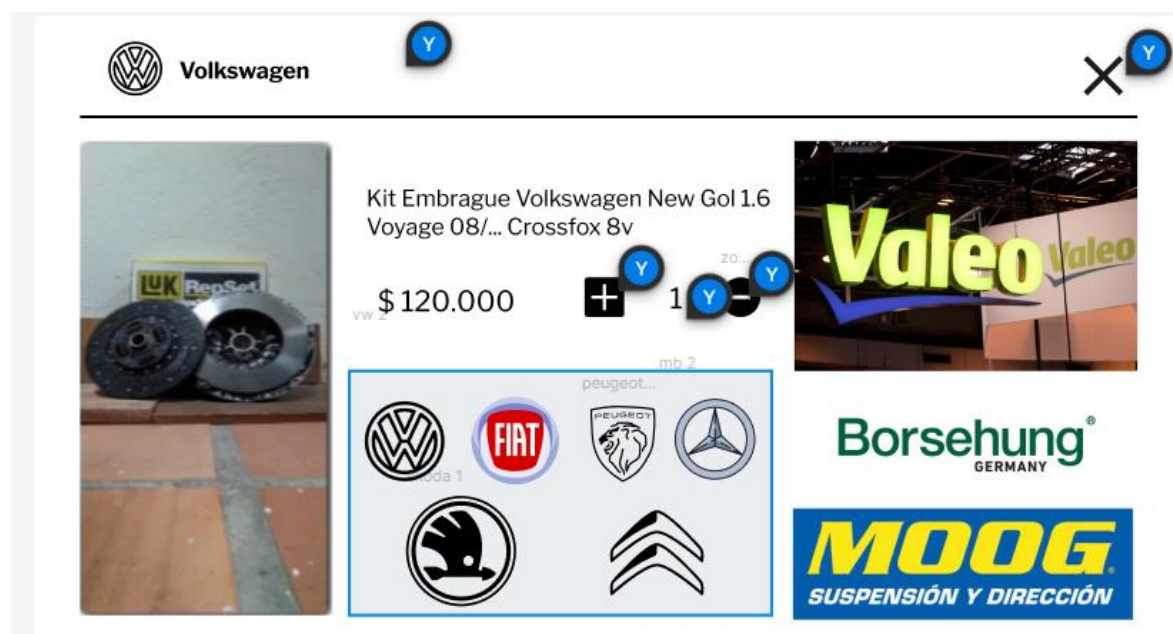
botones de más y menos, que permiten al usuario ajustar el número de unidades que desea adquirir.

A la derecha se encuentran imágenes complementarias de marcas relacionadas con repuestos automotrices, como Valeo, Borsehung y Moog, lo cual refuerza la identificación de fabricantes y proveedores confiables. En la parte inferior central se presentan los logotipos de diferentes marcas europeas, entre ellas Volkswagen, Fiat, Peugeot, Mercedes Benz, Skoda y Citroën.

De esta manera, la tarjeta de producto ofrece una experiencia visual completa, combinando información técnica, opciones de compra y vínculos a marcas relacionadas, todo dentro de un diseño organizado y de fácil navegación.

Figura 18

Detalles del producto



Nota. Detalles de productos del sistema de Casa Europea de Repuestos. Elaborado por el autor (2025).

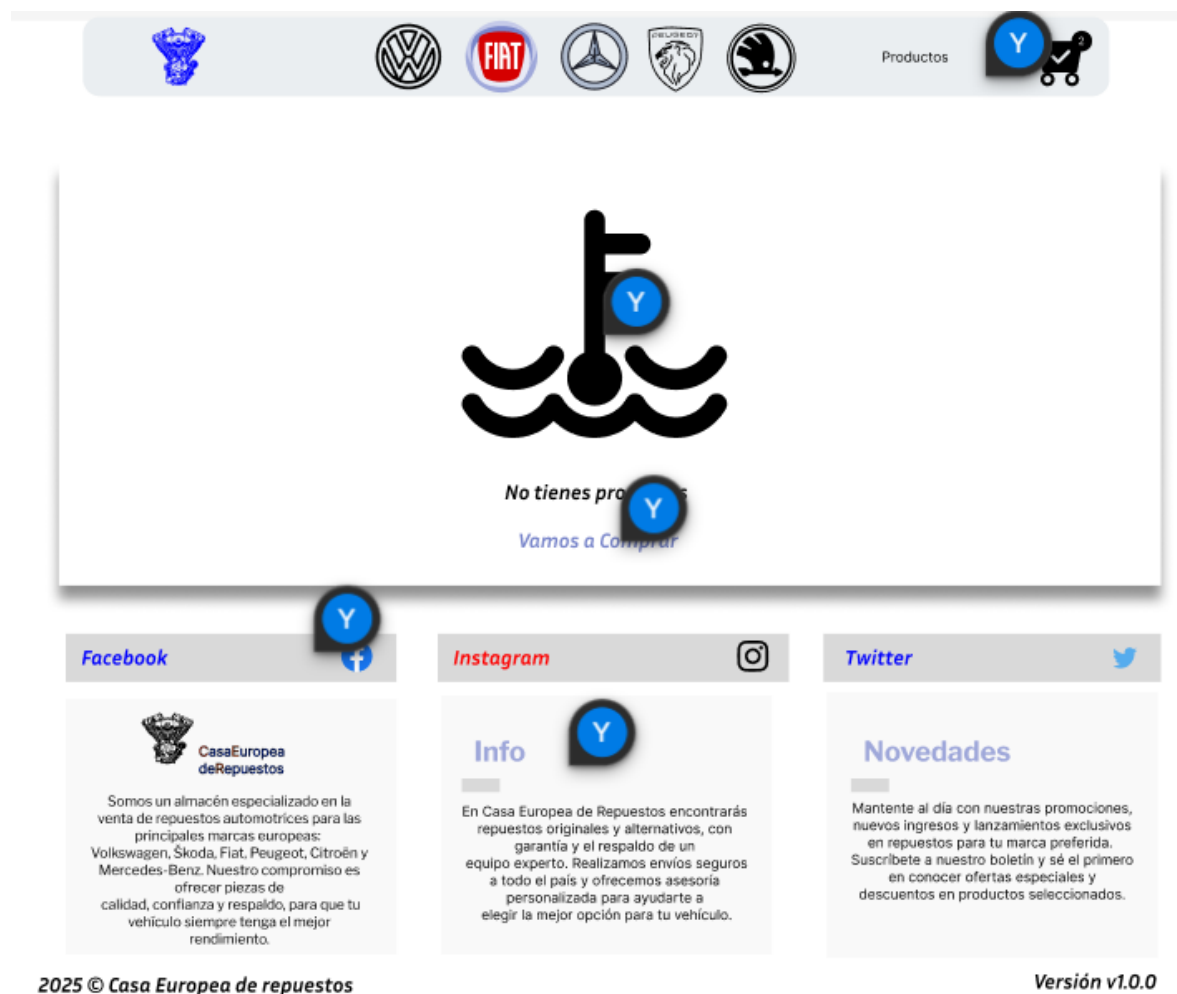
Módulo Compra “Casa Europea de Repuestos”

La siguiente ilustración presenta el diseño del módulo del carrito de compras en el sistema de Casa Europea de Repuestos. En la parte superior se ubica la barra de navegación fija. En el área central de la interfaz se observa un ícono ilustrativo acompañado del mensaje “No tienes productos” y un enlace con la frase “Vamos a comprar”, que dirige al usuario nuevamente al catálogo de artículos disponibles. Este diseño indica de manera clara que el carrito de compras se encuentra vacío y facilita la navegación hacia el módulo de productos para continuar con la selección.

La sección inferior de la vista está organizada en tres columnas que complementan la experiencia del usuario. La primera columna corresponde a Facebook e incluye una breve descripción de la empresa, resaltando su especialización en la venta de repuestos europeos de marcas reconocidas y su compromiso con la calidad. La segunda columna corresponde a Instagram, identificada con el título “Info”, en la que se presentan los beneficios de la tienda, como la disponibilidad de repuestos originales y alternativos, los envíos seguros y la asesoría personalizada. La tercera columna corresponde a Twitter, bajo el título “Novedades”, donde se destacan promociones, lanzamientos exclusivos y descuentos en productos seleccionados.

Figura 19

Módulo de Carrito del Sistema (Vacio)



Nota. Módulo de carrito de compras del sistema de Casa Europea de Repuestos. Elaborado por el autor (2025).

La siguiente ilustración presenta el diseño del módulo del carrito de compras en el sistema de Casa Europea de Repuestos. En la parte superior de la vista se mantiene una barra de navegación fija con los accesos a las diferentes secciones del sistema. En el área central se observa el listado de productos añadidos al carrito, acompañado de su imagen, nombre, cantidad seleccionada y los controles para aumentar o disminuir dicha cantidad. Este diseño facilita la

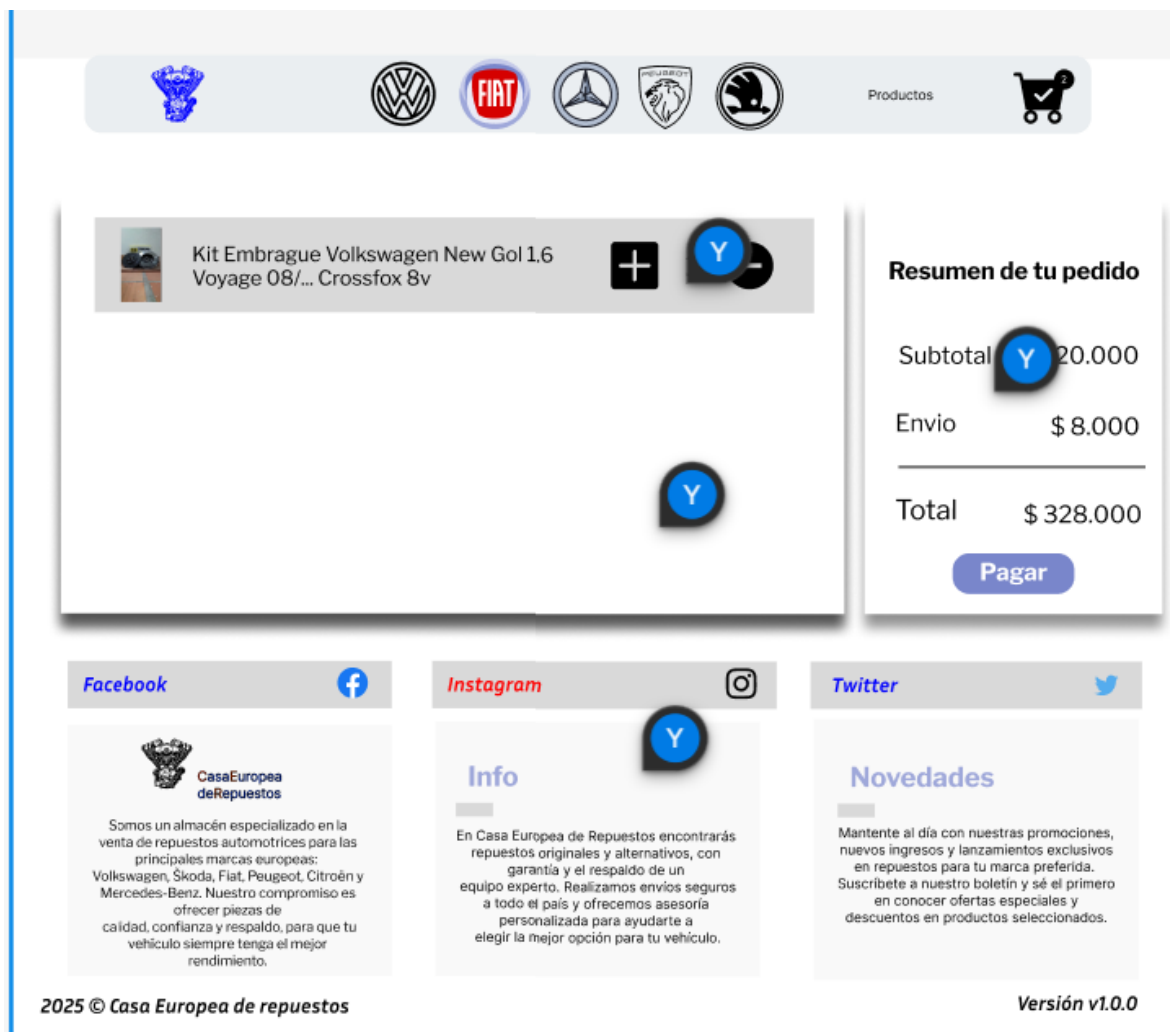
gestión de los artículos seleccionados por el usuario y proporciona una experiencia clara e intuitiva en el proceso de compra.

A la derecha del área principal se incluye un recuadro denominado “Resumen de tu pedido”, donde se presenta de manera detallada el subtotal de los productos, el valor correspondiente al envío y el monto total a pagar. Esta sección finaliza con un botón identificado como “Pagar”, que permite al usuario continuar con el proceso de compra en línea.

La parte inferior de la interfaz mantiene la organización en tres columnas, complementando la experiencia del usuario. La primera columna corresponde a Facebook e incluye una breve descripción de la empresa y su especialización en repuestos europeos. La segunda columna corresponde a Instagram, bajo el título “Info”, donde se destacan los beneficios de adquirir repuestos originales y alternativos, así como la asesoría personalizada. Finalmente, la tercera columna corresponde a Twitter, titulada “Novedades”, que presenta promociones, lanzamientos exclusivos y descuentos disponibles. Todo este diseño refuerza la confianza y el vínculo con los usuarios, integrando la funcionalidad del carrito con la comunicación institucional de la empresa.

Figura 20

Módulo de Carrito del Sistema (con productos)



Nota. Módulo de carrito de compras con productos del sistema de Casa Europea de Repuestos.

Elaborado por el autor (2025).

La siguiente ilustración muestra el diseño del módulo de facturación y registro de usuarios en el sistema de Casa Europea de Repuestos. En la parte superior se mantiene la barra de navegación fija con los accesos a las diferentes secciones del sistema y el ícono del carrito de compras, lo que facilita la continuidad en la navegación del usuario.

En el área central se presenta la sección de inicio de sesión bajo la pregunta “¿Ya has comprado con nosotros?”, donde se ofrecen los campos para ingresar correo y contraseña, acompañados de un botón identificado como “Ingresar”. Justo debajo se ubica el formulario de registro para nuevos clientes, que solicita datos de facturación como nombres, apellidos, correo electrónico, número de teléfono, tipo de persona, contraseña, código postal y dirección. Estos campos permiten recopilar la información necesaria para gestionar tanto la creación de la cuenta como la facturación del pedido.

Finalmente, en la parte inferior del formulario se incluyen dos elementos destacados: un botón identificado como “Registrar”, que facilita el envío de la información ingresada, y el logotipo de Wompi, lo cual indica la disponibilidad de este medio de pago para completar las transacciones. Este diseño combina de manera clara la funcionalidad de autenticación y la gestión de datos de facturación, brindando al usuario una experiencia organizada y eficiente.

Figura 21*Módulo de Pagos del Sistema*

¿Ya has comprado con nosotros?

Ingresar correo Ingresar contraseña

Datos de facturación

Nombres **Apellidos** **Ingresar**

Correo **Nro. Telefono** **Tipo persona**

Contraseña **Código postal** **Dirección**

Registrar **Mercado pago**

2025 © Casa Europea de repuestos Versión v1.0.0

Nota. Módulo de pago del sistema de Casa Europea de Repuestos. Elaborado por el autor (2025).

Código Fuente**FrontEnd**

El desarrollo del frontend constituye uno de los componentes esenciales en un sistema de comercio electrónico, ya que representa la capa de interacción directa entre el usuario y la plataforma. Su diseño e implementación permiten garantizar una experiencia de navegación fluida, accesible y segura, aspectos fundamentales en la aceptación de este tipo de soluciones tecnológicas. Para el presente proyecto se seleccionó la biblioteca React, la cual proporciona una arquitectura basada en componentes que favorece la modularidad y la reutilización del código. Esta decisión se sustenta en la necesidad de contar con una interfaz moderna, escalable, alineada con los lineamientos propuestos por Moncayo Ronquillo (2020), quien enfatiza que el éxito de

una aplicación de comercio electrónico depende en gran medida de su usabilidad y de la organización del código fuente.

La arquitectura del proyecto se estructuró en directorios específicos dentro de la carpeta denominada `src`, los cuales cumplen funciones bien definidas. En este sentido, la carpeta `components` contiene elementos reutilizables como menús, formularios y tarjetas de productos; `pages` organiza las vistas principales del sistema tales como catálogo, carrito de compras, login y registro; `redux` administra el estado global de la aplicación; mientras que `helpers` y `data` se emplean para funciones auxiliares y carga de información estática. Esta organización sigue la lógica de separación de responsabilidades planteada por Moncayo (2020), quien en su propuesta de aplicación móvil define módulos específicos para usuarios, productos y transacciones, con el fin de mantener un sistema mantenible y escalable.

Dentro de los archivos centrales se encuentra `App.jsx`, que actúa como punto de inicio del frontend al renderizar el layout general y administrar la navegación entre páginas. Asimismo, `main.jsx` cumple la función de inicializar la aplicación y enlazarla con el archivo `index.html`, lo que posibilita la carga inicial del sistema. En cuanto a la gestión del estilo, se emplearon hojas de estilo modulares como `App.css` e `index.css`, lo cual permite mantener la coherencia visual. De esta manera, el sistema ofrece al usuario final una experiencia adaptable en ordenadores, lo que concuerda con los principios de usabilidad planteados en la literatura sobre comercio electrónico (García & Martínez, 2019).

La gestión del estado constituye otro aspecto relevante del desarrollo. En este proyecto se utilizó Redux, herramienta que centraliza la información relacionada con los productos, el carrito de compras y la autenticación de usuarios. Esta estrategia resulta fundamental en aplicaciones de

comercio electrónico, donde la consistencia de los datos debe estar garantizada a lo largo de toda la navegación. En la investigación de Moncayo (2020), aunque la solución se desarrolló en React, se observa un enfoque similar al emplear servicios y librerías externas para la gestión de datos de usuarios, compras y pagos. De esta manera, la solución aquí planteada mantiene coherencia con enfoques previos de investigación, adaptándolos a un contexto web con React.

En cuanto a la comunicación con el backend, se implementaron funciones en la carpeta helpers que facilitan la integración con la API REST del sistema. Estas funciones permiten realizar operaciones de registro de usuarios, consulta de productos, gestión de pedidos y procesamiento de pagos. Tal como señala Moncayo (2020), el consumo eficiente de servicios externos y el manejo seguro de transacciones son elementos que incrementan la confiabilidad de la plataforma y favorecen la aceptación de los usuarios en contextos de comercio electrónico. En el caso particular de este sistema, las credenciales y rutas de conexión se gestionan a través de un archivo .env, lo que asegura la protección de información sensible y la facilidad de configuración en distintos entornos de despliegue.

En síntesis, el frontend desarrollado en React constituye una implementación coherente con los principios de modularidad, escalabilidad y seguridad. La arquitectura del código, basada en la separación de componentes, páginas y estado global, responde a las necesidades de un sistema de comercio electrónico que busca ser intuitivo para el usuario y sostenible en el tiempo. Como se puede apreciar en su [repositorio](#).

Gateway (Ocelot)

Dentro de la arquitectura basada en microservicios, el gateway constituye un componente central que permite gestionar el enrutamiento de las solicitudes, aplicar políticas de seguridad y centralizar la comunicación entre los distintos servicios. En el presente proyecto se implementó un gateway utilizando la biblioteca Ocelot, desarrollada sobre .NET Core, la cual proporciona funcionalidades de enrutamiento, balanceo de carga, agregación de solicitudes, autenticación y control de acceso. Este enfoque concuerda con la propuesta de Wang, Zhang, Jia y Liu (2023), quienes señalan que el uso de Ocelot en un entorno de microservicios simplifica la interacción entre cliente y servidor al consolidar los múltiples puntos de acceso en una única interfaz.

La estructura del gateway en este sistema se organizó bajo el proyecto denominado *cer-gateway*, el cual contiene archivos y carpetas que reflejan su configuración y lógica de operación. Entre los elementos principales se encuentran los controladores, modelos y utilitarios que gestionan la respuesta de las solicitudes, así como la configuración del pipeline de Ocelot en los archivos *Program.cs* y *ocelot.json*. Este último define las reglas de enrutamiento, los esquemas de comunicación (HTTP/HTTPS) a los que se dirige cada petición. De acuerdo con Wang et al. (2023), esta configuración permite a Ocelot actuar como un intermediario capaz de distribuir las solicitudes entrantes hacia los microservicios correspondientes, optimizando así el rendimiento y la escalabilidad del sistema.

El archivo *appsettings.json* contiene la configuración complementaria del gateway, incluyendo parámetros de logging, control de dependencias y perfiles de entorno. A su vez, el archivo *ocelot.json* describe las rutas y reglas de acceso, especificando las plantillas de las rutas upstream y downstream, los hosts y puertos de los servicios, así como los métodos HTTP

permitidos. Esta separación entre configuración y lógica de negocio facilita la mantenibilidad del sistema, ya que cualquier cambio en la infraestructura puede gestionarse sin necesidad de modificar el código fuente.

El gateway implementado no integra directamente mecanismos de autenticación mediante tokens JWT, sino que actúa como intermediario en la comunicación hacia los microservicios. Son los microservicios los encargados de validar las credenciales y tokens emitidos, y dado que estos exigen la autenticación JWT, el gateway también debe garantizar que las solicitudes incluyan la información necesaria para cumplir con dichas validaciones. De esta manera, el control de seguridad se mantiene descentralizado en los microservicios, mientras que el gateway se limita a enrutar las peticiones. Según Wang et al. (2023), este enfoque resulta adecuado en entornos de alta concurrencia, ya que asegura que solo los usuarios autorizados accedan a los servicios internos, manteniendo la eficiencia en el procesamiento de las solicitudes.

En síntesis, la implementación de Ocelot como gateway en la solución propuesta proporciona un punto de entrada único y seguro para el ecosistema de microservicios. La capacidad de enrutamiento dinámico, balanceo de carga y aplicación de políticas de seguridad contribuye a la escalabilidad y robustez del sistema, en concordancia con las mejores prácticas descritas en la literatura reciente sobre arquitectura de microservicios. Como se puede apreciar en su [repositorio](#)

Microservicio Usuarios

El microservicio de usuarios constituye uno de los componentes esenciales de la arquitectura de comercio electrónico, ya que centraliza las operaciones relacionadas con la

gestión de clientes y empleados dentro de la plataforma. Este servicio se diseñó siguiendo los principios de independencia, escalabilidad y bajo acoplamiento, los cuales son fundamentales en un esquema basado en microservicios. De acuerdo con Sandoval Pérez y Rojas Pérez (2024), la descomposición de una aplicación en servicios autónomos permite mejorar la mantenibilidad del sistema y facilita su despliegue en entornos de computación en la nube, garantizando al mismo tiempo una respuesta más eficiente frente a la demanda de los usuarios.

La estructura del microservicio `microUsuarios.API` refleja una organización modular que facilita la separación de responsabilidades. En la carpeta `Controllers` se encuentran las clases `ClienteController` y `EmpleadoController`, las cuales actúan como puntos de entrada para las peticiones HTTP relacionadas con las operaciones de clientes y empleados, respectivamente. Estos controladores gestionan las solicitudes entrantes, validan los datos y delegan la lógica de negocio a las capas correspondientes.

La capa `Dao` (`Data Access Object`) está compuesta por las clases `DACliente`, `DAEmpleado` y `DAUtil`, encargadas de la comunicación con la base de datos. Esta capa abstrae las operaciones de acceso a la información, asegurando que la lógica de negocio permanezca desacoplada de los detalles de persistencia. Según Sandoval y Rojas (2024), este enfoque permite mantener la coherencia de los datos y facilita la integración con sistemas de almacenamiento distribuidos en la nube.

En la capa `Logic` se encuentran las clases `BLCliente` y `BLEmpleado`, que encapsulan la lógica de negocio asociada a la gestión de usuarios. Estas clases procesan las reglas específicas de validación, transformación de datos y flujo de operaciones, lo cual garantiza que los

controladores mantengan una función ligera y orientada únicamente a la recepción y respuesta de solicitudes.

El directorio Model se encuentra dividido en subcarpetas Request y Response, donde se definen los objetos de transferencia de datos (DTO) que representan tanto las entradas como las salidas del microservicio. Adicionalmente, la clase GeneralResponse proporciona una estructura uniforme para estandarizar las respuestas hacia el cliente, lo que contribuye a la consistencia de la comunicación entre servicios.

Finalmente, la carpeta Utils contiene clases auxiliares que apoyan la ejecución del microservicio mediante funciones de utilidad reutilizables en diferentes capas. Los archivos de configuración, como appsettings.json, permite establecer parámetros de conexión, variables de entorno y opciones específicas para cada entorno de despliegue. Este mecanismo está en consonancia con las recomendaciones de Sandoval y Rojas (2024), quienes destacan que la parametrización de los microservicios mediante archivos de configuración facilita la portabilidad de las soluciones en escenarios de cloud computing.

En síntesis, el microservicio de usuarios se diseñó con una arquitectura modular que garantiza la separación de responsabilidades y la independencia funcional. Su estructura permite gestionar de forma eficiente las operaciones de clientes y empleados, al tiempo que asegura la escalabilidad del sistema y su alineación con las buenas prácticas de desarrollo de microservicios en entornos de comercio electrónico basados en la nube. Como se observa en su [repositorio](#)

Microservicio Catalogo Productos

Dentro de la arquitectura propuesta, el microservicio de catálogo de productos cumple la función de presentar al usuario la información de los artículos disponibles en la plataforma de comercio electrónico. Su responsabilidad se centra en la visualización del catálogo, ofreciendo mecanismos de filtrado por categorías y marcas, sin encargarse de la gestión directa de los productos. Este diseño responde a los principios de independencia, modularidad y escalabilidad que caracterizan a las arquitecturas basadas en microservicios. De acuerdo con Sandoval Pérez y Rojas Pérez (2024), la segmentación de funcionalidades en servicios autónomos permite mejorar el rendimiento en escenarios de alta demanda y facilita el despliegue en entornos de computación en la nube.

La estructura del proyecto `microCatalogoProductos.API` refleja la organización por capas. En la carpeta `Controllers` se encuentra la clase `CatalogoController`, que gestiona las solicitudes HTTP entrantes relacionadas con la consulta y administración de productos. Esta clase actúa como intermediaria entre el cliente y las capas internas del sistema, delegando la lógica de negocio a la capa `Logic`.

En la carpeta `Dao` se localiza la clase `DACatalogoProductos`, responsable del acceso a datos y de la comunicación con la base de datos que obtiene los registros de productos, categorías y marcas. La abstracción de esta capa garantiza la separación entre persistencia y lógica de negocio. A su vez, la clase `BLCatalogoProductos`, ubicada en la carpeta `Logic`, contiene las reglas de negocio necesarias para la gestión de productos, tales como validaciones, operaciones de consulta y actualizaciones del catálogo.

El directorio Model incluye entidades y objetos de transferencia de datos (DTO). Entre ellos se encuentran las clases Categoría y Marca, que representan la taxonomía del catálogo, así como GeneralResponse, que permite unificar el formato de las respuestas. Las carpetas Request y Response definen los modelos de entrada y salida, asegurando la consistencia en la comunicación entre servicios.

Asimismo, el directorio Utils incorpora componentes de soporte como filtros, resultados de acción y variables de entorno que facilitan la configuración y el manejo de respuestas estándar. Finalmente, los archivos de configuración appsettings.json permite gestionar parámetros específicos para los distintos entornos de despliegue, lo cual contribuye a la portabilidad y flexibilidad del servicio.

De acuerdo con Velasco Navas (2024), la correcta implementación del microservicio de catálogo no solo permite al administrador gestionar productos y categorías, sino que también posibilita al cliente explorar de forma estructurada la oferta disponible. Este servicio constituye, por tanto, un punto neurálgico en la experiencia de compra, al garantizar que la información de los artículos se presente de manera clara, actualizada y accesible desde múltiples interfaces de usuario. Como se evidencia en el [repositorio](#).

MicroServicio Inventario

El microservicio de inventario cumple la función de administrar el ciclo de vida de los productos en la plataforma de comercio electrónico, siendo el encargado de gestionar el catálogo mediante la creación, actualización, eliminación y edición de artículos disponibles para los clientes. A diferencia del microservicio de catálogo de productos, que únicamente muestra la

información a los usuarios finales, el microservicio de inventario concentra las operaciones internas de mantenimiento del catálogo, asegurando que la oferta de artículos en el sistema se mantenga actualizada y consistente. Según Sandoval Pérez y Rojas Pérez (2024), la división de responsabilidades entre microservicios especializados fortalece la escalabilidad y el control de los sistemas, ya que permite que cada módulo responda a necesidades específicas sin generar dependencia directa entre componentes.

La estructura del proyecto `microInventario.API` se organiza en distintas capas con responsabilidades claramente delimitadas. En la carpeta `Controllers` se encuentra `InventarioController`, que recibe las solicitudes HTTP relacionadas con las operaciones de gestión de productos y actúa como punto de acceso al servicio. La carpeta `Dao` contiene la clase `DAInventario`, responsable de realizar las operaciones de persistencia en la base de datos, mientras que la carpeta `Logic` alberga la clase `BLInventario`, que concentra la lógica de negocio, incluyendo las reglas necesarias para validar y procesar las operaciones de inventario.

El directorio `Model` se divide en subcarpetas `Request` y `Response`, que representan los modelos de entrada y salida de datos, complementados por la clase `GeneralResponse`, encargada de estandarizar las respuestas del microservicio. Adicionalmente, en la carpeta `Util` se incluyen funcionalidades de soporte como el manejo centralizado de errores a través de filtros y clases auxiliares, lo que contribuye a robustecer la estabilidad del sistema.

De acuerdo con Velasco Navas (2024), la correcta gestión del inventario en un entorno de microservicios no solo asegura la disponibilidad de información precisa en el catálogo, sino que también habilita a los administradores para tomar decisiones sobre el ciclo de vida de los productos, optimizando la oferta y alineándola con la demanda del mercado. En este sentido, el

microservicio de inventario constituye un eje fundamental dentro de la plataforma, ya que permite mantener actualizado el catálogo mostrado al usuario y garantiza la coherencia entre los procesos internos de la organización y la experiencia del cliente final. Como se presenta en su [repositorio](#).

Microservicio Pedidos

El microservicio de pedidos constituye el componente responsable de gestionar las órdenes realizadas por los clientes dentro de la plataforma de comercio electrónico. Su función principal es administrar el ciclo de vida de los pedidos, desde su creación hasta la actualización de su estado, lo cual incluye operaciones como la confirmación, el despacho, la entrega. Además, este servicio permite la generación de reportes y consultas que facilitan a los administradores el monitoreo del comportamiento de las ventas y la trazabilidad de los pedidos. De acuerdo con Sandoval Pérez y Rojas Pérez (2024), la definición de microservicios especializados en la administración de procesos críticos mejora la escalabilidad del sistema y favorece la adaptación de la plataforma a cambios en la demanda de los usuarios.

La estructura del proyecto microPedidos.API refleja un diseño modular orientado a la separación de responsabilidades. En la carpeta Controllers se encuentra la clase PedidoController, encargada de gestionar las solicitudes HTTP relacionadas con la creación y administración de pedidos. La capa Dao, representada por la clase DAPedidos, se encarga de las operaciones de persistencia y comunicación con la base de datos. Por su parte, la capa Logic incluye la clase BLPedido, donde se concentra la lógica de negocio vinculada con la validación, actualización de estados y manejo de las reglas específicas de procesamiento de órdenes.

El directorio Model define las entidades y objetos de transferencia de datos (DTO), tales como Pedido y CarritoDetalleDto, que representan tanto los pedidos en sí como los detalles de los artículos incluidos en cada transacción. Asimismo, la clase GeneralResponse garantiza la estandarización en las respuestas generadas por el servicio. Complementariamente, en la carpeta Utils se incluyen herramientas para el manejo de errores, resultados de acción y filtros de excepción, lo cual refuerza la robustez del sistema y asegura una experiencia uniforme en el consumo de la API.

En concordancia con lo planteado por Velasco Navas (2024), la gestión eficiente de los pedidos en un ecosistema de comercio electrónico no solo garantiza el cumplimiento de las expectativas de los clientes, sino que también ofrece a los administradores una visión estratégica mediante reportes y estadísticas que apoyan la toma de decisiones. Este microservicio, al actualizar de manera dinámica los estados de las órdenes, facilita tanto la comunicación con el cliente respecto al avance de su pedido como la supervisión del proceso logístico por parte del personal administrativo.

En síntesis, el microservicio de pedidos constituye un módulo clave dentro de la arquitectura basada en microservicios, ya que asegura la correcta administración de las órdenes de compra, integra la gestión de estados y facilita la obtención de reportes. Su diseño modular y desacoplado contribuye a la eficiencia, la escalabilidad y la trazabilidad de los procesos de comercio electrónico, aspectos fundamentales en plataformas modernas que operan bajo arquitecturas distribuidas en la nube. Como se puede apreciar en el [repositorio](#).

Microservicio Pagos

El microservicio de pagos constituye el módulo encargado de gestionar las transacciones económicas dentro de la plataforma de comercio electrónico. Su función principal es la generación de órdenes de pago a través de la integración con la pasarela Wompi, lo que permite ofrecer a los clientes un mecanismo seguro y confiable para completar sus compras. Este servicio no solo crea las órdenes de pago, sino que también se comunica con el microservicio de pedidos con el fin de actualizar el estado de las órdenes una vez que la transacción ha sido confirmada. De acuerdo con Sandoval Pérez y Rojas Pérez (2024), la incorporación de microservicios especializados en el procesamiento de pagos favorece la descentralización de responsabilidades y garantiza mayor escalabilidad en entornos distribuidos.

La estructura del proyecto `microPagos.API` se organiza en diferentes capas que reflejan una separación clara de funciones. En la carpeta `Controllers` se incluyen las clases `PagoController` y `NotificacionController`, que administran las solicitudes HTTP asociadas a la creación de órdenes de pago y a la recepción de notificaciones de estado provenientes de la pasarela externa. La capa `Dao`, representada por la clase `DAPagos`, se encarga de la persistencia de información de las transacciones, mientras que la carpeta `Logic`, con la clase `BLPagos`, centraliza la lógica de negocio vinculada a la validación de estados y la interacción con servicios externos.

El directorio `Model` contiene los objetos de transferencia de datos (DTO) en las carpetas `Request` y `Response`, además de la clase `GeneralResponse`, que estandariza el formato de las respuestas. En la carpeta `Utils` se localizan componentes auxiliares como `ActionResults`, filtros de validación y la subcarpeta `ExternalAPI`, que incluye las clases `MercadoPago` y `PedidosClient`,

encargadas de la comunicación directa con la pasarela de pago y el microservicio de pedidos, respectivamente. Esta integración asegura que cada transacción financiera esté asociada de manera inmediata con el pedido correspondiente, lo que permite actualizar su estado de forma automática.

Según Velasco Navas (2024), la integración de plataformas de pago externas en una arquitectura de microservicios resulta fundamental para garantizar la seguridad y confiabilidad de las transacciones, al tiempo que mejora la experiencia del usuario mediante procesos de validación en tiempo real. En este sentido, el microservicio de pagos implementado no solo verifica los estados de las órdenes en la plataforma de Mercado Libre, sino que también asegura la coherencia de los registros internos de la aplicación mediante la sincronización con el microservicio de pedidos.

En síntesis, el microservicio de pagos proporciona un mecanismo confiable para la generación, validación y actualización de transacciones económicas en la plataforma. Su diseño modular y su integración con pasarelas externas permiten un procesamiento seguro, mientras que la comunicación directa con el microservicio de pedidos asegura la consistencia entre el estado de pago y el ciclo de vida de las órdenes. De esta manera, este servicio constituye un eje estratégico dentro de la arquitectura de comercio electrónico basada en microservicios. Como se aprecia en su [repositorio](#).

Pruebas del Sistema

El proceso de pruebas de software constituye una de las fases más críticas dentro del ciclo de vida del desarrollo, ya que permite garantizar la calidad, fiabilidad y correcto funcionamiento del sistema antes de su implementación definitiva. Según Paz (2016), las pruebas de calidad de software deben entenderse como una actividad sistemática orientada a identificar errores, asegurar el cumplimiento de los requerimientos establecidos y verificar que el producto final satisfaga las expectativas del usuario. En este sentido, el análisis del proceso de pruebas no solo busca la detección de fallas técnicas, sino también la validación de aspectos relacionados con la usabilidad, el rendimiento y la estabilidad del sistema.

En el contexto del sistema de comercio electrónico Casa Europea de Repuestos, las pruebas desempeñaron un papel esencial para garantizar que la plataforma cumpla con los criterios de funcionalidad, eficiencia y seguridad definidos durante su diseño. Estas pruebas abarcaron tanto los requerimientos funcionales, asociados con el comportamiento esperado del sistema frente a acciones específicas del usuario, como los no funcionales, enfocados en la calidad del desempeño, la protección de la información y la experiencia de uso.

De acuerdo con Paz (2016), un proceso de pruebas bien estructurado permite reducir riesgos, optimizar recursos y mejorar la percepción de calidad del software entregado. Por esta razón, las pruebas realizadas en esta unidad se planificaron y ejecutaron bajo un enfoque metodológico, garantizando la trazabilidad entre los requerimientos y los resultados obtenidos. Asimismo, se documentaron de manera detallada las evidencias de validación, contribuyendo a la transparencia del proceso y al aseguramiento de la calidad global del sistema.

Pruebas Requerimientos Funcionales

Caso de prueba 1: Registro de usuarios (REQ-F01)

Nombre de Caso de Prueba: Registro de usuarios en el sistema e-commerce.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

El servidor debe estar en ejecución y la base de datos operativa.

El usuario debe tener acceso al formulario de registro desde la página principal.

Ingresar al módulo “Registrarse”.

Diligenciar los campos obligatorios: nombre, correo electrónico, contraseña, dirección, código postal, número de identificación, entre otros.

Presionar el botón “Crear cuenta”.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

Resultado de la prueba:

El sistema permitió el registro de un nuevo usuario, mostrando un mensaje de confirmación y almacenando los datos correctamente en la base de datos.

El comportamiento fue consistente con lo esperado.

Figura 22*Registro Usuario*

¿Ya has comprado con nosotros? patriciahuertas@gmail.com Ingresar

Datos de facturación

Nombres: usuario de prueba Correo: usuarioprueba@gmail.com

Nro. Teléfono: 1234567890 Nro de documento: 123456798 Tipo persona: Jurídica

Código Postal: 15001 Dirección: dirección de prueba Contraseña: contraseñaprueba

Confirmar Contraseña: contraseñaprueba

Registrar

Éxito
Cliente creado correctamente.
OK

Actualizar Pagar con Wompi

© 2025 CasaEuropeaDeImpuestos Hecho por Yonner Suarez Versión v1.0.4

Nota. Corresponde al Caso de prueba 1 “Registro de usuarios en el sistema e-commerce”.

Elaborado por el autor (2025).

Caso de prueba 2: Prueba de Autenticación de Administradores (REQ-F02)

Nombre de Caso de Prueba: Inicio de sesión de administradores registrados.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

El administrador debe tener una cuenta activa en la base de datos.

El sistema debe encontrarse en línea y operativo.

Acceder al formulario de inicio de sesión.

Ingresar credenciales válidas (correo y contraseña).

Presionar “Iniciar sesión”.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

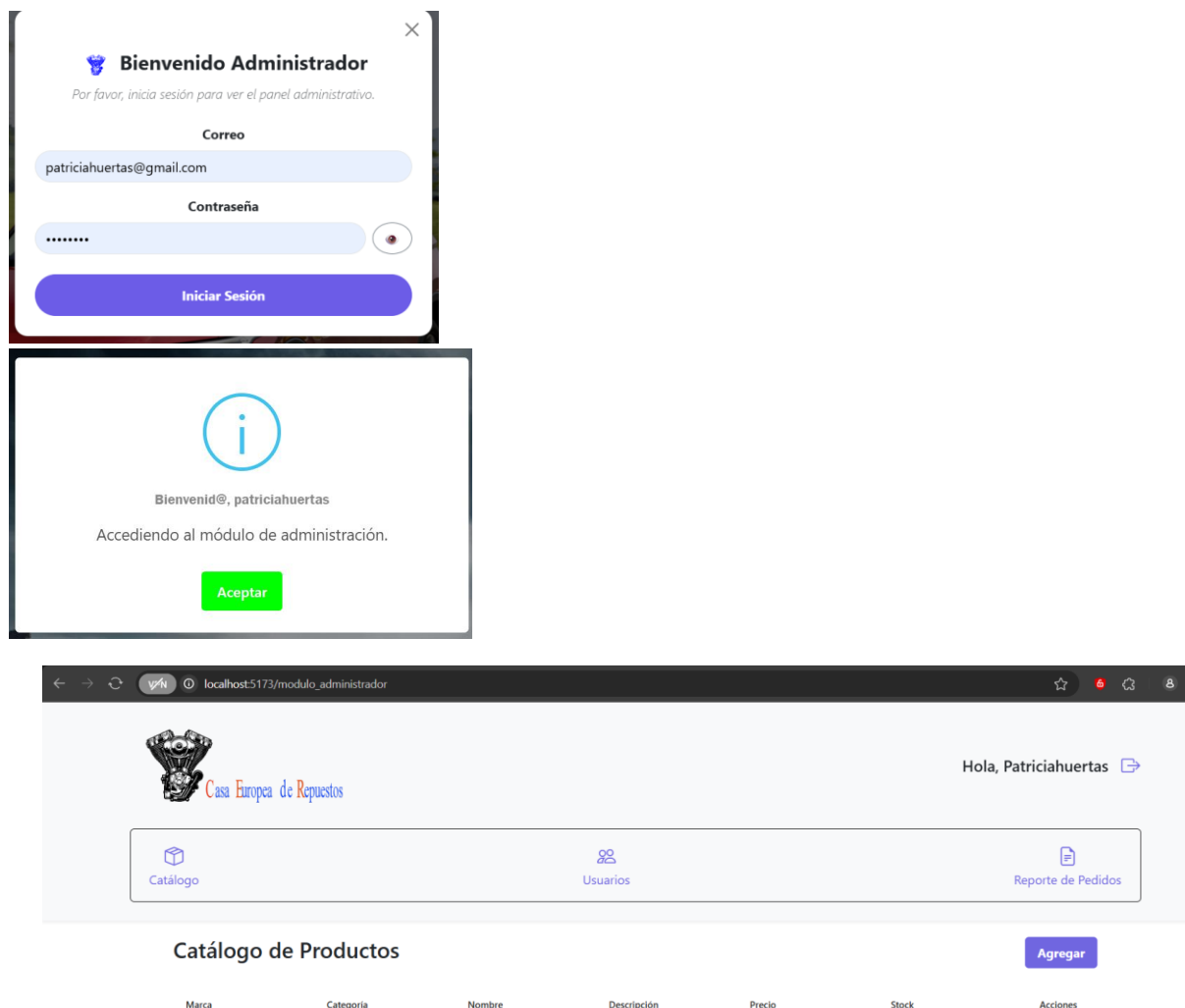
Resultado de la prueba:

El sistema permitió el ingreso del administrador y redirigió correctamente al panel de control.

Las credenciales inválidas fueron rechazadas, asegurando la autenticación adecuada.

Figura 23

Ingreso Modulo Administrativo



Nota. Corresponde al Caso de prueba 2 “Autenticación de administradores registrados”.

Elaborado por el autor (2025).

Caso de prueba 3: Prueba de Gestión de Productos (REQ-F03)

Nombre de Caso de Prueba: Agregar, editar y eliminar productos del catálogo.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

El administrador debe estar autenticado.

Debe existir conexión con la base de datos.

Ingresar al módulo “Gestión de productos”.

Agregar un nuevo producto con todos sus campos obligatorios.

Editar el producto creado.

Eliminar un producto existente.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

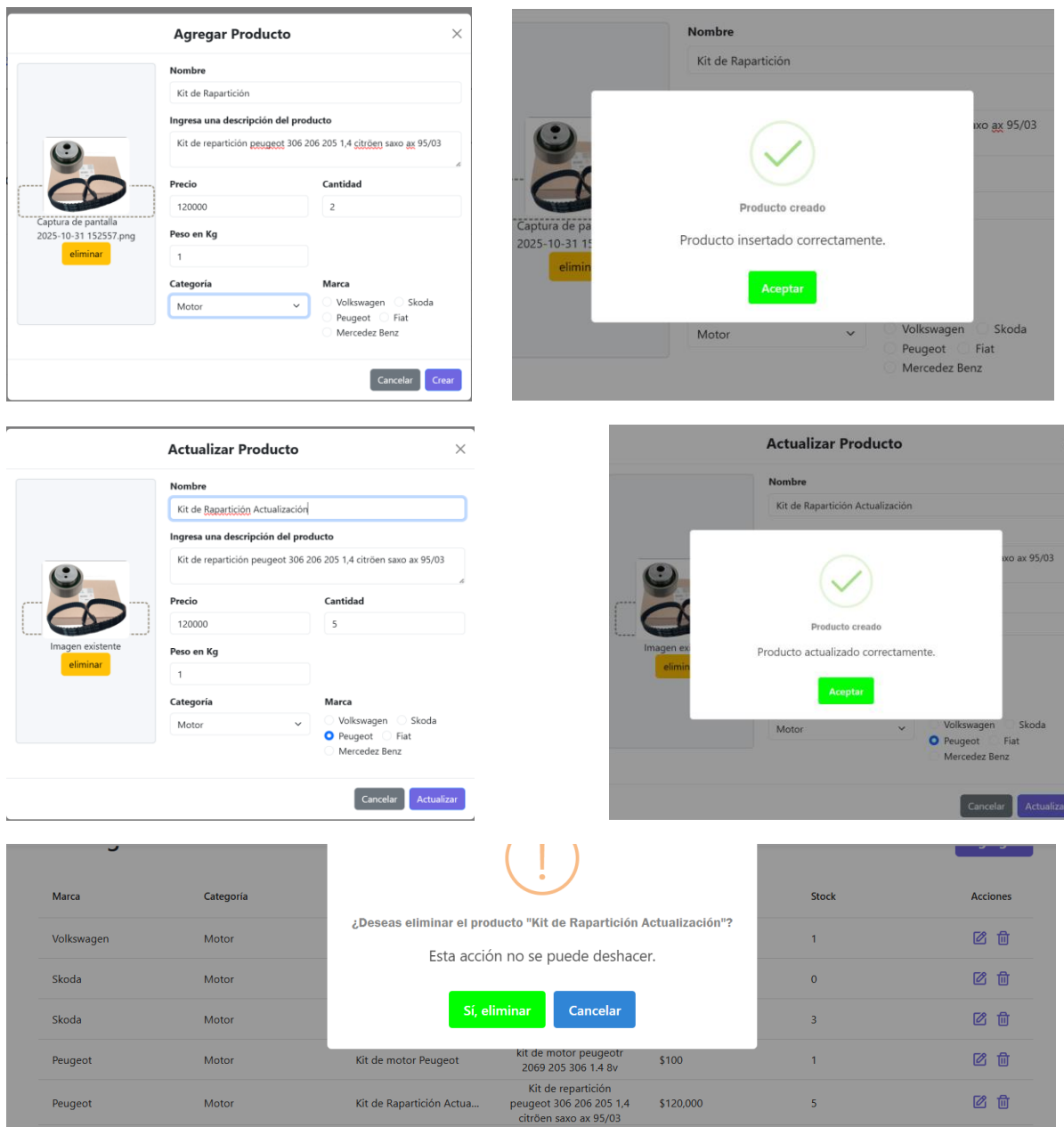
Resultado de la prueba:

El sistema ejecutó correctamente las operaciones de agregar, editar y eliminar productos.

Los cambios se reflejaron de inmediato en la interfaz del cliente.

Figura 24

Resultado de la prueba para el requerimiento identificado con REQ-F03



Nota. Corresponde al Caso de prueba 3 “Gestión de productos en el sistema e-commerce”.

Elaborado por el autor (2025).

Caso de prueba 4: Prueba de Carrito de Compras (REQ-F04)

Nombre de Caso de Prueba: Agregar y eliminar productos del carrito de compras.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

El usuario debe tener conexión a internet.

Debe existir al menos un producto en el catálogo.

Ingresar al catálogo de productos.

Seleccionar un artículo y añadirlo al carrito.

Eliminar un producto del carrito.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

Resultado de la prueba:

El sistema actualizó correctamente el subtotal y el total al agregar o eliminar productos, mostrando la información de forma dinámica.

Figura 25

Resultado de la prueba para el requerimiento identificado con REQ-F04

The image displays a web application interface for a car parts store. At the top, there is a search bar with the text "Ejemplo: Amortiguador - embrague" and a "Productos" button. Below the search bar, there are several product cards. The first row shows "Kit De Motor VW" for \$1,000, "KIT De Motor Sk" for \$1,000 (marked "Agotado"), and "Bomba De Agua" for \$100. The second row shows "KIT De Motor Peugeot" and "KIT De Motor Fiat".

On the right side, there are two screenshots of a shopping cart. The top one shows the cart with "Kit de motor VW" (quantity 1, price \$1000) and "bomba de agua" (quantity 1, price \$100). The subtotal is \$1,100, shipping is \$0, and the total is \$1,100. The bottom one shows the cart with "Kit de motor VW" (quantity 1, price \$1000) and "bomba de agua" (quantity 0, price \$100). The subtotal is \$1,000, shipping is \$0, and the total is \$1,000.

At the bottom, there is a product card for "Skoda bomba de agua" for \$100. To its right is a "Resumen" (Summary) section with the following details:

Resumen	
Sub-Total	\$ 100
Envío	\$ 0
Total	\$ 100

Below the summary is a "Calcular Envío" button. To the right of the summary is another "Ver Carrito" button.

Nota. Corresponde al Caso de prueba 4 “Gestión del carrito de compras”. Elaborado por el autor (2025).

Caso de prueba 5: Prueba de Procesamiento de Pagos (REQ-F05)

Nombre de Caso de Prueba: Procesamiento de pagos en línea mediante pasarela simulada.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba**Precondiciones de la Prueba:**

El usuario debe tener productos en el carrito.

La pasarela de pago debe estar activa.

Proceder al pago desde el carrito.

Ingresar los datos de la tarjeta simulada.

Confirmar la compra.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

Resultado de la prueba:

El sistema procesó correctamente la transacción, generó la factura digital y registró la venta en la base de datos.

Figura 26

Resultado de la prueba para el requerimiento identificado con REQ-F05

Actualizar **Pagar con Wompi**

Pago a
Casa Europea de Repuestos
\$8.600 COP

Ingresa tus datos

Correo electrónico
yonnerhazziel@gmail.com

Nombres y apellidos
Yonner Hazziel Duvan Suárez Huertas

Celular o Número telefónico
+57 3186750864

Continuar con tu pago

PAGOS SEGUROS POR **Wompi** **Grupo Cibest**

The image shows two parts of the Wompi interface. On the left is the merchant dashboard for 'Casa Europea de Repuestos', displaying account balances and a sidebar menu. On the right is a detailed view of a transaction.

Dashboard Data:

- Acreditado: \$7,495,80
- Saldo disponible: \$0,00

Transaction Details:

- Nombres y apellidos: Yonner Hazziel Duvan Suárez Huertas
- Número telefónico o celular: +573186750868
- Número celular usado: 3186750868
- ID Transacción Nequi: 350-123-12041672-1761065991us7H

Entradas contables:

Concepto	Monto
Pago	\$8.600,00
Comisión	-\$927,90
IVA de la comisión	-\$176,30

Transaction Summary Table:

Estado	Monto y cliente	Datos del pago	Hora y fecha
Pagada	COP \$8.600,00 yonnerhazziel@gmail.com	#1330662-1761065991-36512 Ref: PEDIDO_32	oct. 21 11:59 AM Ver más

Nota. Corresponde al Caso de prueba 5 “Procesamiento de pagos”. Elaborado por el autor (2025).

Pruebas Requerimientos No Funcionales

Caso de prueba 7: Prueba de Seguridad de la Información (REQ-NF02)

Nombre de Caso de Prueba: Verificación de cifrado y protección de datos personales.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

La base de datos debe contener usuarios registrados.

Acceder a la base de datos del sistema.

Verificar el formato de almacenamiento de contraseñas.

Fecha de inicio de la prueba: 20/10/2025.

Fecha de finalización de la prueba: 20/10/2025.

Resultado de la prueba:

Las contraseñas se almacenan cifradas mediante el algoritmo SHA256, garantizando la confidencialidad y la integridad de la información.

Figura 27

Resultado de la prueba para el requerimiento identificado con REQ-NF02

```
SELECT * FROM casa_europea_repuestos_ecommerce.tbl_cer_usuario;
```

har_correo	cer_varchar_nro_documento	cer_varchar_contrasena
iertas@gmail.com	40035449	1463286264534ACF046FE1D9F42316CA
@transportesrapidos.com	1002394576	D9765AD4FC979CE3FD4C3D5AC925EC3D46C1293F931D191182B4964BAE4F61A0
@gmail.com	12347	B42EE1120213684B409086083D3896F0
@gmail.com	123456789	692A9EBE9588CCDF1DCD928BEE11CBF8
@gmail.com	12345678	61664FCF8916A3BFD2D465D791E9A6D
@gmail.com	12345	D6D8503B6D13C6D8DDFB6FD565F12F10
@gmail.com	123456	9FF9696E96D31E3060EBE61ECS8980DC
@gmail.com	414	E1450D4BDAECC5247A9ECBD3D5884954
@gmail.com	5745	DDF807A868AD6499EDD2FCA8C8691D72
zzziel@gmail.com	12345654	BA6DBC0578E2ECBD2263AB9C341285A5
@gmail.com	252352352	0071638C16B0B4A4C7948E4597695DBC

Nota. Corresponde al Caso de prueba 7 “Seguridad de la información”. Elaborado por el autor (2025).

Caso de prueba 8: Prueba de Compatibilidad (REQ-NF04)

Nombre de Caso de Prueba: Funcionamiento en navegadores.

Responsable: Yonner Duván Suárez Huertas.

Casos de Prueba

Precondiciones de la Prueba:

El sistema debe estar publicado en un entorno accesible.

Ejecutar la aplicación en Google Chrome, Edge y Firefox.

Fecha de inicio de la prueba: 20/10/2025.

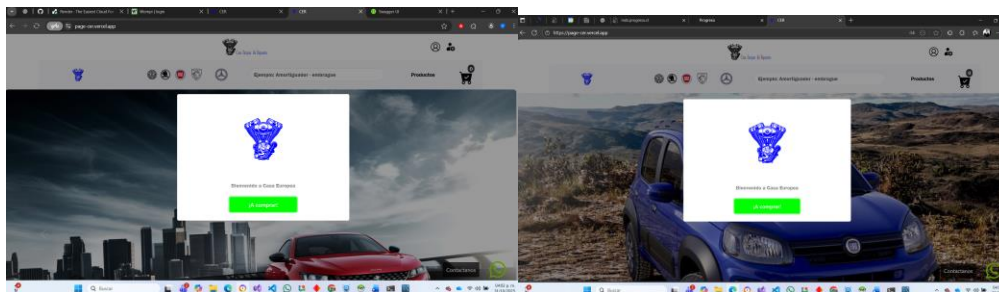
Fecha de finalización de la prueba: 20/10/2025.

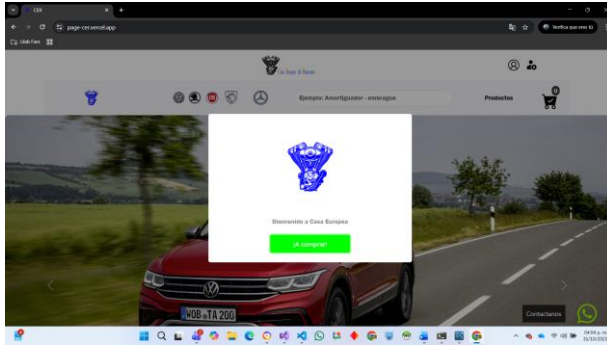
Resultado de la prueba:

El sistema funcionó correctamente en todos los navegadores.

Figura 28

Resultado de la prueba para el requerimiento identificado con REQ-NF04





Nota. Corresponde al Caso de prueba 10 “Disponibilidad del sistema”. Elaborado por el autor (2025).

Despliegue del Sistema

El despliegue del sistema constituye una etapa determinante en el ciclo de vida del desarrollo de software, ya que en ella se materializan los resultados del análisis, diseño e implementación para poner en marcha el producto en un entorno operativo funcional. Este proceso abarca la configuración de la infraestructura, la integración de los componentes de software, la validación de los servicios y la puesta en funcionamiento del sistema bajo condiciones reales de operación. En la actualidad, la adopción de modelos de despliegue basados en la nube ha transformado este proceso, al ofrecer soluciones escalables, seguras y de bajo costo operativo.

De acuerdo con Blázquez León (2022), los entornos cloud permiten alcanzar altos niveles de disponibilidad y flexibilidad, facilitando la implementación de servicios de comercio electrónico accesibles para pequeñas y medianas empresas. De forma complementaria, Arango Ortiz (2022) destaca la pertinencia del enfoque *Cloud Native*, basado en arquitecturas de microservicios, que favorece la escalabilidad y el mantenimiento continuo mediante el uso de prácticas DevOps y metodologías ágiles como SCRUM. En consonancia con ambos autores, el presente proyecto aborda el despliegue de un sistema de comercio electrónico probado tanto en un entorno local como en la nube, con el fin de garantizar su operatividad, estabilidad y futura transición al entorno de producción.

Requisitos de hardware

Para garantizar el correcto funcionamiento del sistema, se establecen tres entornos de ejecución diferenciados: local, de pruebas (QA) y de producción (PROD). Esta segmentación permite validar progresivamente el comportamiento del sistema, detectar posibles fallos antes de

la puesta en marcha definitiva y asegurar la coherencia entre las distintas etapas del ciclo de despliegue.

En el entorno local, el sistema se ejecuta en equipos de desarrollo con especificaciones técnicas que facilitan la virtualización de servicios y el procesamiento simultáneo de los diferentes componentes. Se recomiendan las siguientes características mínimas: procesador Intel Core i5 o equivalente, 8 GB de memoria RAM, 250 GB de almacenamiento tipo SSD, conectividad a Internet estable y sistema operativo Windows 10 o Ubuntu 20.04 LTS. Este entorno se utiliza principalmente para la depuración inicial, la integración de módulos y la validación de dependencias.

El entorno de pruebas o QA (Quality Assurance) se emplea para verificar la estabilidad, seguridad y rendimiento del sistema bajo condiciones controladas que simulan su comportamiento en producción. En este nivel se ejecutan pruebas automatizadas, de integración y de carga, con el fin de identificar errores antes del despliegue final.

Finalmente, el entorno de producción (PROD) corresponde a la infraestructura definitiva en la que el sistema se encuentra disponible para los usuarios finales. En este caso se recomienda una infraestructura similar a la siguiente.:

Tabla 16*Arquitectura de la Plataforma en la Nube*

Componente / Servicio	Función o Propósito	Tecnología o Tipo de Servicio	Características Clave
Infraestructura general	Plataforma de implementación y ejecución de todos los servicios	Nube pública con aprovisionamiento dinámico	Escalabilidad automática, alta disponibilidad y asignación de recursos bajo demanda
Backend	Ejecución de la lógica de negocio bajo arquitectura sin servidor	Funciones <i>serverless</i> para microservicios	Despliegue ágil, reducción de costos operativos, facturación por uso real
Frontend	Almacenamiento y distribución de la interfaz web	Almacenamiento de objetos con red de distribución de contenido (CDN)	Entrega global, baja latencia, acceso rápido desde distintas ubicaciones
Base de datos	Gestión de datos transaccionales	Servicio de base de datos relacional administrado	Configuración en múltiples zonas de disponibilidad, respaldo automático y alta tolerancia a fallos
Gestión de API	Enrutamiento y control de solicitudes HTTP	Puerta de enlace de API administrada	Integración centralizada, control de tráfico y

Componente / Servicio	Función o Propósito	Tecnología o Tipo de Servicio	Características Clave
	hacia los servicios backend		soporte para autenticación y seguridad
Monitoreo y registros	Supervisión del rendimiento y registro de eventos del sistema	Servicio de monitoreo y gestión de logs en la nube	Métricas en tiempo real, generación de alertas y análisis de registros
Gestión de seguridad y accesos	Control de identidades, permisos y credenciales	Servicio de administración de identidades y accesos	Autenticación segura, políticas de acceso granular y control de usuarios y servicios

Nota. Tabla de componentes claves para los servicios en la nube. *Fuente.* Autor

Esta configuración permite desplegar una arquitectura distribuida, escalable y segura, capaz de adaptarse a variaciones en la carga de trabajo, garantizar la continuidad del servicio y mantener niveles óptimos de rendimiento y disponibilidad (Arango Ortiz, 2022).

Requisitos de Software

El despliegue del sistema requiere un ecosistema de software que permita integrar de forma coherente las etapas de desarrollo, pruebas y producción, garantizando la automatización, portabilidad y administración eficiente de los componentes. Para ello, la solución se fundamenta en tecnologías de código abierto y servicios administrados en la nube, los cuales facilitan la implementación de buenas prácticas de ingeniería de software y despliegue continuo.

El desarrollo del sistema se llevó a cabo empleando .Net para la capa de backend y React para la capa de frontend, tecnologías que permiten una arquitectura modular y la separación lógica entre la presentación y la lógica de negocio. La base de datos se gestiona con MySQL, debido a su estabilidad, escalabilidad y compatibilidad con entornos locales y en la nube. Esta configuración facilita la replicación entre los tres entornos definidos: desarrollo local, entorno de pruebas (QA) y entorno de producción (PROD), garantizando la coherencia de la aplicación a lo largo del ciclo de vida del despliegue.

Para el control de versiones y la gestión del código fuente se utiliza Git, con repositorios centralizados en GitHub, lo que permite la colaboración entre desarrolladores, la trazabilidad de los cambios y la automatización de procesos a través de flujos de trabajo (*workflows*) integrados. El sistema de integración y entrega continua (CI/CD) se implementa mediante GitHub Actions, herramienta que automatiza las tareas de construcción, pruebas y despliegue en los diferentes entornos, asegurando una entrega controlada y progresiva de las versiones del sistema.

La contenedorización se realiza mediante Docker, lo que permite empaquetar los servicios y sus dependencias para asegurar la portabilidad entre los entornos. Se optimiza la gestión de recursos y el escalado automático. Además, la infraestructura como código (IaC) se gestiona con plataformas cloud, herramientas que facilitan la automatización, replicabilidad y consistencia en la configuración de la infraestructura en la nube (Arango Ortiz, 2022).

Este conjunto tecnológico proporciona una base sólida y coherente para el despliegue del sistema, al reducir la intervención manual, incrementar la confiabilidad del proceso y permitir una integración fluida entre las fases de desarrollo, pruebas y operación, en conformidad con las prácticas actuales de la ingeniería de software moderna (Blázquez León, 2022).

Los requerimientos de software para el mantenimiento y desarrollos adicionales del proyecto son los siguientes:

Tabla 17

Inventario de Tecnologías y Herramientas Utilizadas en el Proyecto

Categoría	Componente / Herramienta	Propósito o Función	Requerimiento Específico
Lenguajes y Frameworks de Desarrollo	Backend (.NET)	Desarrollo de la lógica de negocio y servicios del sistema	Soportar arquitectura modular, interoperabilidad con bases de datos SQL, y despliegue en contenedores
	Frontend (React)	Desarrollo de la interfaz de usuario	Permitir una arquitectura de componentes, integración con API REST y despliegue en entornos web escalables
Gestión de Datos	MySQL	Base de datos relacional para almacenamiento de información	Soportar replicación entre entornos (DEV, QA, PROD), garantizar integridad y escalabilidad

Categoría	Componente / Herramienta	Propósito o Función	Requerimiento Específico
Control de Versiones	Git (repositorios centralizados)	Gestión del código fuente y trabajo colaborativo	Permitir control de versiones, trazabilidad de cambios y manejo de ramas (branches) Ejecutar pipelines
Integración y Entrega Continua (CI/CD)	Plataforma de automatización de flujos de trabajo	Construcción, pruebas y despliegue continuo del sistema	automáticos para build, test y deploy; garantizar entregas seguras y progresivas
Contenedorización	Docker	Empaquetado de aplicaciones y dependencias	Asegurar portabilidad entre entornos, aislamiento de servicios y despliegue homogéneo
Infraestructura como Código (IaC)	Herramientas de definición de infraestructura (ej. CloudFormation / CDK o equivalentes)	Automatización de la creación y gestión de recursos en la nube	Permitir despliegues reproducibles, configuraciones consistentes y versionamiento de la infraestructura

Categoría	Componente / Herramienta	Propósito o Función	Requerimiento Específico
Entornos de Ejecución	Desarrollo (DEV), Pruebas (QA), Producción (PROD)	Separación de fases del ciclo de vida del sistema	Mantener coherencia y sincronización de configuraciones entre entornos
Automatización y Monitoreo	Herramientas de integración y supervisión	Supervisión de procesos y rendimiento del sistema	Facilitar la observabilidad, la detección temprana de fallos y la trazabilidad de eventos
Gestión y Buenas Prácticas	Ecosistema de desarrollo colaborativo	Asegurar calidad y coherencia en el ciclo de vida del software	Integrar prácticas de desarrollo ágil, control de calidad y despliegue continuo

Nota. Tabla de componentes inventario de tecnologías y herramientas utilizadas en el proyecto.

Fuente. Autor

Configuración de Entorno

La configuración del entorno de despliegue constituye una fase crítica dentro del ciclo de vida del sistema, ya que permite garantizar su estabilidad, disponibilidad y correcto funcionamiento en cada una de las etapas previas a la puesta en producción. Este proceso comprende la definición de variables de entorno, la configuración de dependencias, el

establecimiento de conexiones entre servicios y la asignación de recursos de infraestructura tanto en entornos locales como en la nube.

En el entorno local, la configuración se centra en la preparación de las variables de entorno necesarias para la ejecución del backend, el frontend y la base de datos. Los microservicios, desarrollados bajo la plataforma .NET Core, se ejecutan de manera independiente dentro de contenedores Docker, lo que permite la virtualización y aislamiento de cada componente. Estos contenedores se comunican entre sí mediante redes internas configuradas en Docker Compose, asegurando la interoperabilidad de los servicios y el acceso controlado a la base de datos. Para la gestión de las peticiones HTTP y el enrutamiento entre los diferentes microservicios, se implementa un API Gateway utilizando Ocelot, lo cual facilita la centralización del tráfico, el control de seguridad y la optimización del consumo de recursos.

El frontend, desarrollado con React, conectándose a los servicios expuestos por el gateway mediante endpoints REST. Esta arquitectura permite mantener una clara separación entre la interfaz de usuario y la lógica de negocio, promoviendo la escalabilidad y el mantenimiento independiente de cada módulo. La base de datos, administrada con MySQL, se configura localmente con el servidor local de MySQL.

En el entorno de pruebas o QA, la configuración se replica a partir del entorno local, incorporando ajustes orientados a la validación del rendimiento, la verificación de la seguridad y la estabilidad de las integraciones entre componentes. En esta fase, los microservicios desarrollados en .NET se despliegan en la plataforma Render, que ofrece un entorno administrado para la ejecución de contenedores y servicios web, permitiendo simular condiciones reales de operación y evaluar la respuesta del sistema ante cargas variables. La base

de datos se aloja en Cloud Clever Database, servicio que proporciona instancias MySQL escalables y seguras, con capacidad de respaldo automático y acceso remoto controlado.

El frontend, desarrollado en React, se despliega en Vercel, plataforma que facilita la integración continua y la entrega automatizada desde el repositorio GitHub, generando entornos de previsualización y versiones intermedias para pruebas funcionales. Esta configuración permite verificar el comportamiento del sistema completo en un entorno que emula la infraestructura de producción, sin comprometer la estabilidad del entorno principal.

Durante esta etapa se ejecutan pruebas automatizadas de integración, rendimiento y carga, con el objetivo de validar la correcta comunicación entre los microservicios, la disponibilidad del gateway Ocelot y la eficiencia en el intercambio de datos con la base de datos. Los resultados de estas pruebas permiten ajustar parámetros de rendimiento, identificar posibles cuellos de botella y optimizar la configuración antes de la puesta en marcha en producción.

El entorno de producción se recomienda implementar sobre una plataforma de computación en la nube, utilizando una arquitectura distribuida basada en contenedores y servicios administrados. Los microservicios desarrollados en .NET se despliegan en un entorno sin servidor o mediante un orquestador de contenedores, según los requerimientos de procesamiento. El gateway de servicios se configura dentro del entorno de contenedores o en una plataforma de aplicaciones gestionada, mientras que el frontend, desarrollado en React, se aloja en un servicio de almacenamiento de objetos, con distribución global a través de una red de entrega de contenido (CDN).

La base de datos se administra mediante un servicio relacional gestionado con el motor MySQL, configurado en modo multizona de disponibilidad, lo que garantiza redundancia y recuperación ante fallos.

La configuración de los entornos incorpora políticas de seguridad fundamentales, que incluyen el cifrado de datos en tránsito mediante protocolos SSL/TLS, la gestión segura de credenciales mediante un servicio de almacenamiento de secretos, y el control de acceso basado en identidades y permisos.

De igual manera, se implementan prácticas de observabilidad que permiten monitorear métricas de rendimiento, generar alertas y realizar un seguimiento detallado del comportamiento del sistema, contribuyendo a la detección temprana de incidentes y a la mejora continua de la operación.

Esta configuración modular y escalable, sustentada en la arquitectura de microservicios, asegura la continuidad del servicio, la recuperación ante fallos y la adaptabilidad a las necesidades futuras del sistema. De acuerdo con Arango Ortiz (2022), el uso de servicios nativos en la nube permite optimizar recursos y mantener la disponibilidad, mientras que Blázquez León (2022) resalta la importancia de una configuración reproducible y automatizada para garantizar la estabilidad del sistema en entornos productivos.

Recomendaciones para despliegue

Para lograr una implementación exitosa y sostenible del sistema, es indispensable aplicar un conjunto de buenas prácticas que garanticen la eficiencia operativa, la seguridad de la información y la estabilidad de los servicios desplegados. Las siguientes recomendaciones se

fundamentan en la experiencia adquirida durante el desarrollo del proyecto y en la revisión de los trabajos de Blázquez León (2022) y Arango Ortiz (2022), quienes destacan la importancia de la automatización, la escalabilidad y la observabilidad dentro de los procesos de despliegue en la nube.

En primer lugar, se recomienda la automatización integral del ciclo de despliegue, utilizando herramientas de integración y entrega continua (CI/CD) configuradas desde GitHub Actions. Este enfoque permite generar versiones controladas del sistema, ejecutar pruebas automatizadas y desplegar actualizaciones de forma progresiva en los distintos entornos (local, QA y producción), minimizando los errores manuales y garantizando la trazabilidad de los cambios.

En segundo lugar, es esencial mantener una escalabilidad controlada en el entorno de producción. Para ello, se recomienda implementar mecanismos de autoescalado y balanceo de carga que ajusten dinámicamente la capacidad de cómputo en función de la demanda. De esta manera, el sistema puede responder eficientemente a los incrementos de tráfico, garantizando un rendimiento óptimo y una alta disponibilidad en todo momento.

Antes del paso a producción, debe llevarse a cabo una fase exhaustiva de validación de rendimiento, que incluya pruebas de carga, estrés y disponibilidad en los entornos de calidad (QA) y preproducción. Estas pruebas permiten evaluar la capacidad del sistema para manejar múltiples solicitudes concurrentes, detectar posibles cuellos de botella y ajustar los parámetros de configuración de los microservicios, del gateway y de las instancias de ejecución antes del lanzamiento definitivo.

La aplicación sistemática de estas recomendaciones, en concordancia con los principios de la cultura DevOps, permite optimizar el ciclo de vida del despliegue, mejorar la eficiencia operativa y reducir los riesgos asociados al mantenimiento de la infraestructura. En conjunto, estas prácticas fortalecen la fiabilidad, seguridad y escalabilidad del sistema, asegurando una operación continua y sostenible en entornos productivos (Arango Ortiz, 2022; Blázquez León, 2022).

Manual de usuario

El manual de usuario constituye un componente fundamental dentro del proceso de implementación y despliegue de sistemas de software, ya que proporciona las directrices necesarias para la correcta interacción entre los usuarios y la aplicación desarrollada. Su objetivo principal es ofrecer una guía clara, estructurada y accesible que facilite la comprensión de las funcionalidades del sistema, así como el adecuado aprovechamiento de sus recursos. De acuerdo con Pressman y Maxim (2020), la documentación orientada al usuario final cumple una función esencial en la aceptación del sistema, al garantizar que las operaciones se realicen de forma eficiente y segura.

En el contexto de este proyecto, el manual de usuario tiene como propósito describir los procedimientos de acceso, navegación y operación del sistema de comercio electrónico desarrollado bajo una arquitectura de microservicios en .NET, con una interfaz construida en React y desplegada en servicios en la nube de Amazon Web Services (AWS). Este documento detalla los pasos necesarios para el registro de usuarios, la gestión de productos, la realización de transacciones y la administración de la plataforma, diferenciando las funcionalidades disponibles para los distintos tipos de usuarios, como clientes y administradores.

Asimismo, el manual busca promover la autonomía del usuario final y reducir la dependencia del soporte técnico, permitiendo una experiencia de uso intuitiva y confiable. Según Sommerville (2016), una adecuada documentación incrementa la satisfacción del usuario y contribuye a la reducción de errores operativos, lo que repercute positivamente en la calidad del servicio. En este sentido, el presente manual se constituye como una herramienta clave para la correcta adopción y operación del sistema dentro del entorno productivo.

Enalce al Manual: <https://manual-usuario-cer.vercel.app/>

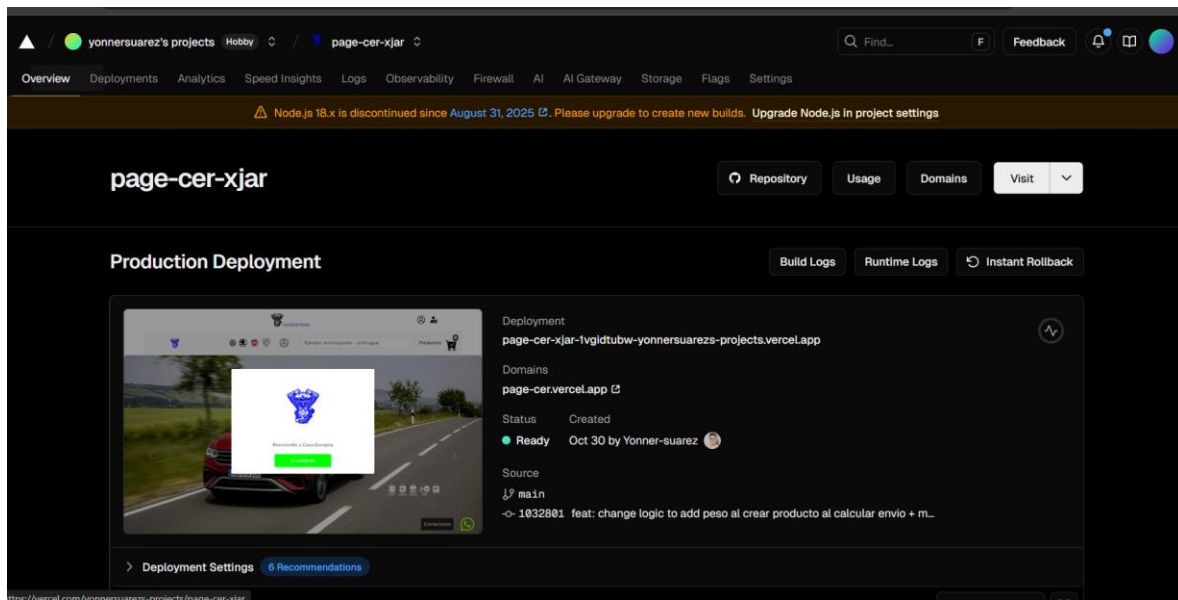
El sistema ha sido sometido a pruebas exhaustivas en entornos locales y en la nube, comprobándose su operatividad, compatibilidad y estabilidad. Las pruebas funcionales, de carga y rendimiento se realizaron sobre las plataformas render, cloud clever y vercel, evidenciando tiempos de respuesta óptimos y correcta integración entre los módulos del sistema.

Actualmente, el sistema se encuentra preparada para su despliegue en producción en AWS, a la espera de la disponibilidad de recursos institucionales y de la aprobación del departamento de tecnologías de la información (TI). Una vez completadas estas condiciones, el sistema podrá ser puesto en funcionamiento a nivel empresarial, garantizando un servicio de comercio electrónico totalmente operativo y escalable.

En la ilustración 29 se observa una evidencia del proceso de despliegue, que corresponde al panel de control de la plataforma Vercel, el cual muestra los detalles del Despliegue de ambientes de pruebas para el proyecto page-cer-xjar o componente front del comercio electrónico. Donde incluye el commit de origen

Figura 29

Despliegue Front Vercel



Nota. Dashboard del despliegue frontend de Casa Europea de Repuestos. Elaborado por el autor (2025).

En la Ilustración 30 se observa una evidencia del proceso de despliegue de microservicios, que corresponde al panel de gestión de la plataforma *cloud* Render, mostrando una lista de cinco Microservicios y un Gateway.

Figura 30

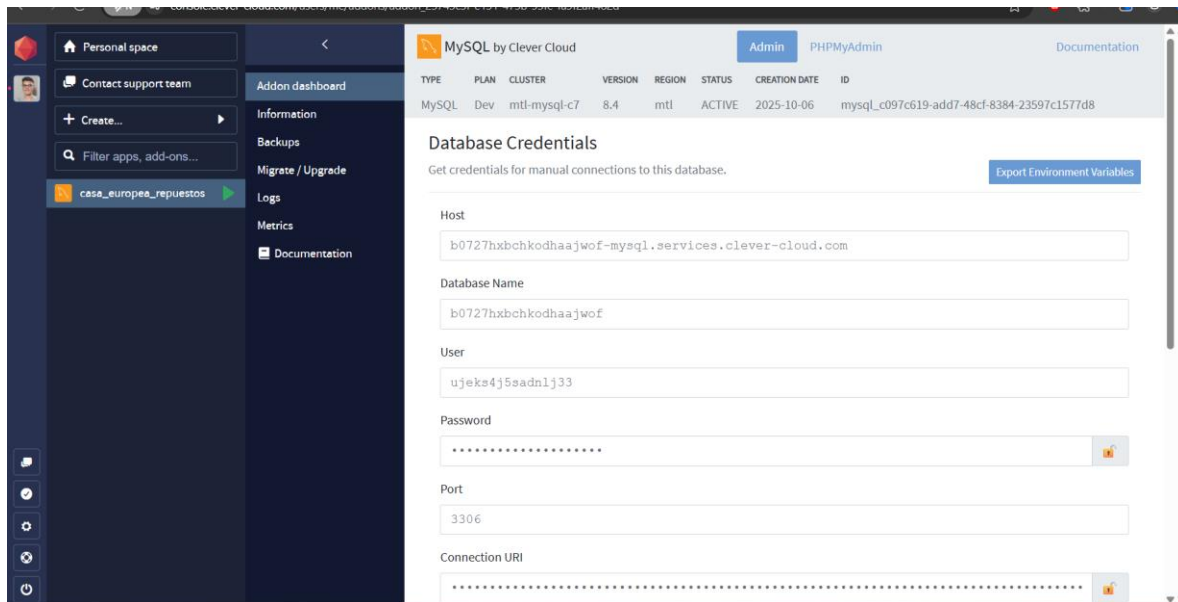
Despliegue de Microservicios en Render

The screenshot displays the Render dashboard interface. On the left is a sidebar with navigation menus for 'My Workspace', 'Projects', 'Blueprints', 'Environment Groups', 'INTEGRATIONS', 'NETWORKING', and 'WORKSPACE'. The main content area is titled 'Ungrouped Services' and features a search bar and a table of services. The table has columns for 'SERVICE NAME', 'STATUS', 'RUNTIME', 'REGION', and 'UPDATED'. All services shown are in a 'Suspended by you' status and are running on 'Docker' in the 'Oregon' region.

SERVICE NAME	STATUS	RUNTIME	REGION	UPDATED
cer-pagos	Suspended by you	Docker	Oregon	11d
cer-inventario	Suspended by you	Docker	Oregon	11d
cer-catalogo-productos	Suspended by you	Docker	Oregon	11d
cer-usuarios	Suspended by you	Docker	Oregon	20d
cer-gateway	Suspended by you	Docker	Oregon	20d
cer-pedidos	Suspended by you	Docker	Oregon	20d

Nota. Dashboard de los despliegues de microservicios de Casa Europea de Repuestos. Elaborado por el autor (2025).

En la Ilustración 31 se observa una evidencia del proceso de despliegue de la base de datos, que corresponde al panel de control de *Addons* de la plataforma Clever Cloud, mostrando las credenciales de conexión de la base de datos MySQL desplegada en la nube.

Figura 31*Despliegue de Base de Datos en Cloud Clever*

The screenshot displays the 'MySQL by Clever Cloud' dashboard. On the left, a dark sidebar contains navigation options: 'Personal space', 'Contact support team', 'Create...', 'Filter apps, add-ons...', 'casa_europea_repuestos', 'Add-on dashboard', 'Information', 'Backups', 'Migrate / Upgrade', 'Logs', 'Metrics', and 'Documentation'. The main content area shows a table with columns: TYPE, PLAN, CLUSTER, VERSION, REGION, STATUS, CREATION DATE, and ID. The table contains one entry: MySQL, Dev, mtl-mysql-c7, 8.4, mtl, ACTIVE, 2025-10-06, and mysql_c097c619-add7-48cf-8384-23597c1577d8. Below the table is the 'Database Credentials' section, which includes a link to 'Export Environment Variables' and several input fields: Host (b0727hxbchkodhaajwof-mysql.services.clever-cloud.com), Database Name (b0727hxbchkodhaajwof), User (ujeks4j5sadnj33), Password (masked with dots), Port (3306), and Connection URI (masked with dots).

Nota. Dashboard del despliegue Base de Datos de Casa Europea de Repuestos. Elaborado por el autor (2025).

Trabajos Futuros

Se una proyección de las líneas de desarrollo futuras para el sistema de gestión de comercio electrónico implementado en Casa Europea de Repuestos. Se abordan cuatro ejes estratégicos: accesibilidad, internacionalización y localización, analítica de datos y generación de reportes.

Accesibilidad

La accesibilidad web constituye un componente esencial en el diseño de aplicaciones inclusivas. Esta se define como el conjunto de prácticas que permiten que personas con diversas capacidades sensoriales, motrices o cognitivas puedan interactuar adecuadamente con los contenidos digitales (Rengifo et al., 2020). Para garantizar el cumplimiento de estos principios, se propone adoptar las Web Content Accessibility Guidelines (WCAG) 2.1 como estándar técnico de referencia, reconociendo la norma ISO/IEC 40500 como su equivalente normativo.

En una primera fase, se recomienda implementar los requisitos del nivel A de WCAG 2.1, lo cual implica asegurar funcionalidades básicas como la navegación por teclado, el uso de texto alternativo en imágenes y la estructuración semántica mínima del contenido. Posteriormente, en una segunda fase, se sugiere avanzar hacia el nivel AA, incorporando mejoras en el contraste visual, la jerarquía lógica del contenido y el etiquetado semántico de los elementos interactivos.

La validación de estos requisitos debe realizarse mediante herramientas automáticas como axe o Pa11y, complementadas con sesiones de prueba con usuarios reales o representantes de colectivos con necesidades de accesibilidad. Los resultados obtenidos deben documentarse en el expediente de calidad del proyecto, asegurando trazabilidad y cumplimiento normativo (Rengifo et al., 2020).

Internacionalización

La internacionalización (i18n) y la localización (l10n) son procesos complementarios que permiten adaptar una aplicación a diferentes contextos lingüísticos y culturales. La internacionalización se refiere a las decisiones de diseño que posibilitan la adaptación del sistema sin modificar su estructura, mientras que la localización implica la traducción y adaptación de contenidos a un idioma y cultura específicos (Corte, 2002).

Como acción inicial, se propone habilitar el idioma inglés como segundo idioma del sistema. Esta decisión estratégica permite ampliar el alcance del mercado objetivo y facilita la integración con servicios externos que operan en dicho idioma. A nivel técnico, se recomienda utilizar codificación UTF-8 en todas las tablas y conexiones, así como modelar entidades traducibles mediante estructuras separadas (por ejemplo, `product_translations` con atributos como `locale`, `name` y `description`).

Asimismo, se sugiere externalizar los textos de la interfaz en archivos de recursos o servicios de internacionalización, y establecer un proceso de revisión y versionado de las traducciones antes de su publicación. La prioridad de despliegue debe centrarse en la interfaz pública y las páginas de catálogo, dejando para fases posteriores la traducción de mensajes administrativos y correos transaccionales.

Módulo de Analítica de Datos

El módulo de analítica de datos tiene como propósito transformar los datos operacionales del sistema en información útil para la toma de decisiones estratégicas. Este proceso se fundamenta en técnicas de extracción, transformación y carga (ETL), análisis exploratorio de

datos (EDA) y minería de datos, orientadas a la gestión comercial y logística (Gómez Ayala, 2020).

La arquitectura propuesta contempla cuatro capas: ingestión de datos (logs y transacciones), almacenamiento analítico (data mart), procesamiento (ETL/ELT) y visualización. Se sugiere diseñar un esquema dimensional básico que incluya hechos de ventas y dimensiones como producto, tiempo y cliente. Este modelo debe alimentarse desde la base de datos operativa del sistema.

Los indicadores clave de desempeño (KPI) iniciales incluyen: ventas por SKU, tasa de conversión, ticket promedio, rotación de inventario y pedidos pendientes. El análisis exploratorio debe identificar patrones de comportamiento, estacionalidades y correlaciones entre promociones y ventas. En cuanto a minería de datos, se recomienda iniciar con técnicas de segmentación de clientes (clustering) y reglas de asociación para identificar relaciones de compra (market basket analysis). Como entregable inicial, se plantea la construcción de un tablero de control con los KPI mínimos y un informe EDA que documente hallazgos y acciones derivadas.

Modulo Reportador

El reportador es un componente funcional que permite generar, programar y exportar informes a partir de los datos del sistema. Su diseño debe contemplar la posibilidad de definir filtros, formatos y frecuencias de entrega, tanto por parte de los usuarios como de los procesos operativos (Rodríguez Noa, 2022).

Se priorizan los siguientes reportes: (1) ventas diarias por SKU y canal, con filtros por fecha y estado del pedido; (2) inventario crítico, con alertas por correo electrónico; (3) resumen financiero de cobros y devoluciones por periodo; y (4) ranking de clientes por volumen y

frecuencia de compra. Desde el punto de vista técnico, se recomienda exponer una API para la generación de reportes y permitir la exportación en formatos CSV y PDF. También se puede considerar el uso de herramientas de inteligencia de negocios (BI) ligeras o librerías server-side para la generación programada de informes.

Conclusiones.

Dentro de la identificación, construcción y documentación de los requerimientos funcionales y no funcionales permitió establecer una base sólida para el desarrollo del sistema e-commerce, garantizando que la solución propuesta respondiera de manera precisa a las necesidades del cliente, el almacén y los usuarios finales. Este proceso aseguró la inclusión de aspectos legales, técnicos y comerciales, lo que facilitó una comprensión integral del entorno operativo y permitió definir un rumbo claro para el diseño y la implementación del sistema, minimizando riesgos y fortaleciendo la alineación entre los objetivos del negocio y las funcionalidades del software.

El diseño de la arquitectura del sistema, acompañado por una interfaz gráfica intuitiva y una experiencia de usuario centrada en la usabilidad, posibilitó la construcción de un entorno digital, escalable y adaptable a las necesidades del comercio electrónico. La estructuración de módulos como catálogo, carrito, pagos, pedidos, soporte e inventario garantizó una distribución funcional ordenada, facilitando la interacción del usuario y la gestión operativa interna. De esta manera, el diseño se consolidó como un pilar fundamental para asegurar la eficiencia del sistema y mejorar la experiencia de navegación y compra.

El desarrollo de los módulos del sistema e-commerce permitió materializar las funcionalidades planteadas, integrando procesos clave como la gestión de productos, administración, control de inventarios y procesamiento seguro de pagos. La implementación técnica demostró la viabilidad del diseño propuesto y permitió construir una plataforma robusta y operativa, capaz de responder a los requerimientos establecidos. Este avance tangible evidenció la importancia de una arquitectura bien definida y del uso de tecnologías adecuadas para garantizar el correcto funcionamiento del ecosistema digital.

La ejecución de pruebas funcionales, de seguridad y usabilidad permitió validar la estabilidad, integridad y confiabilidad del sistema, asegurando que cada componente cumpliera con los requisitos exigidos y respondiera de manera adecuada a distintos escenarios operativos. Estas pruebas fueron fundamentales para identificar oportunidades de mejora, corregir errores y optimizar el rendimiento, garantizando que la plataforma ofreciera una experiencia segura, eficiente y satisfactoria tanto para los administradores como para los usuarios finales.

El sistema fue preparado para su despliegue en un entorno de producción, aunque este paso aún no ha sido aprobado por las directivas de la empresa; sin embargo, se dejó completamente configurado y validado en un ambiente QA destinado a pruebas internas. La capacitación brindada al personal del almacén permitió que los usuarios se familiarizaran con la plataforma y comprendieran su funcionamiento, facilitando una futura adopción efectiva cuando el sistema sea aprobado para su puesta en marcha. Este avance garantiza que la empresa cuente con una solución lista para integrarse en sus procesos operativos, fortaleciendo la gestión comercial, mejorando la visibilidad del negocio y sentando las bases para una atención al cliente más eficiente y sostenible en el tiempo.

Referencias Bibliográficas

Alca, O. (2001). *Comercio electrónico*. Universidad del Pacífico, Centro de Investigación Universidad del Pacífico, Centro de Investigación.

Arturo, G. H. B. (2014). Aplicación de convenios internacionales frente al reglamento propio de un portal de comercio electrónico internacional. *Criterio Jurídico*, 14(2), 107-130.

Ayaviri-Panozo, A., & Ramírez-Correa, P. E. (2019). Teorías más Utilizadas en la Negociación de Precios Colaborativos entre Empresas de la Cadena de Suministros. *Información tecnológica*, 30(6), 201-210.

Bañares Colastra, M. (2016). Herramienta para el prototipado de aplicaciones móviles usando mockups.

Barrientos Felipa, P. (2017). Marketing+ internet= e-commerce: oportunidades y desafíos. *Revista finanzas y política económica*, 9(1), 41-56.

Blanco, C. (2008). *Patrones de Diseño*. Ingeniería del Software I.

Blázquez León, A. (2022). Análisis, diseño e implementación de un servicio de comercio electrónico en la nube (Doctoral dissertation, Telecomunicacion).

Calabrese, J., Muñoz, R., Pasini, A. C., Esponda, S., Boracchia, M., & Pesado, P. M. (2017). Asistente para la evaluación de características de calidad de producto de software propuestas por ISO/IEC 25010 basado en métricas definidas usando el enfoque GQM. En *XXIII Congreso Argentino de Ciencias de la Computación* (La Plata, 2017).

Canchala, A., de Uso, C., de Uso, S., de Uso, D. D. C., de Estructura Estática, D., de Interacción, D., & de Estados, D. (2004). UML, ejemplo sencillo sobre Modelado de un Proyecto. *Developer Network*.

Candela Mantilla, M. J. (2025). *E-commerce para la Gestión de Inventarios de una empresa importadora de productos de ferretería en Lima*.

Castañeda, D. H. T., & Zavala, J. G. (2012). Comercio electrónico. *Contribuciones a la Economía*, 10(2), 35.

Colmenero Seneira, M. (2025). *Comportamiento del consumidor en entornos de comercio online*.

Corte, N. (2002). Localización e internacionalización de sitios web. *Tradumàtica: traducció i tecnologies de la informació i la comunicació*, (1).

Derwey Castellanos, M., Rodríguez Díaz, F. C., & Caicedo Rosero, R. A. (2021). Desarrollo de software para la gestión de inventario, almacenamiento y despachos de producto terminado para la empresa ChoriRico [Trabajo de grado, Universidad Cooperativa de Colombia].

Díaz Doria, A. (2023). Influencia del comportamiento del consumidor en las estrategias de precios utilizadas en el comercio electrónico (Doctoral dissertation, Universidad Nacional de La Plata).

Fernández-Portillo, A., Sánchez-Escobedo, M. C., Jiménez-Naranjo, H. V., & Hernández-Mogollón, R. (2015). La importancia de la Innovación en el Comercio Electrónico. *Universia Business Review*, (47), 106-125.

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2000). *Patrones de diseño*. dotNetManía, 34.
- Gaona García, J. F. (2024). Desarrollo de un sistema e-commerce para la automatización de ventas y envíos de equipos tecnológicos en el establecimiento Aliengamer utilizando la metodología Kanban.
- Gomez Ayala, A. D. (2020). Plan de negocios para la creación de una empresa de bi y analítica de datos dirigida a las pyme del sector Ecommerce en Colombia.
- Gordieiev, O. L. E. K. S. A. N. D. R., Kharchenko, V. Y. A. C. H. E. S. L. A. V., & Gordieieva, D. A. R. I. A. (2022). Software requirements profile quality model. *International Journal of Computing*, 21(1), 111-119.
- Ibarra, J. M. S. (2017). Legislación vigente en materia tributaria del comercio electrónico (e-commerce) en Colombia y la necesidad de un pronunciamiento por parte del legislador. *Academia & Derecho*, (15), 85-110.
- Jingyi, W. (2021). Aplicación de la cadena de suministro en el comercio electrónico.
- Jones, C. (2010). Éxitos y fracasos de los proyectos de comercio electrónico. Disponible en.
- Laguna, M. A. (2009). Desarrollo de Líneas de Productos: un Caso de Estudio en Comercio Electrónico. En *7th Latin American and Caribbean Conference for Engineering and Technology* (pp. 2-5).
- Maida, E. G., Pacienza, J., & Casiello, F. A. (2015). *Metodologías de desarrollo de software*.
- Malca, O. (2001). *Comercio electrónico*.

Manobanda Natif, A. G. (2025). Diseño y desarrollo de microfrontends y microservicios para varios módulos de una plataforma de e-commerce: módulos de carrito de compras, órdenes y pagos.

Moncayo Ronquillo, C. A. (2020). Diseño e implementación de una solución multiplataforma para incentivar el comercio electrónico entre residentes de una urbanización (Doctoral dissertation, Espol).

Ojeda, J. C., & Fuentes, M. D. C. G. (2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. *Universidades*, (52), 37-47.

Ortiz, A. F. A. (2022). Diseño, implementación y despliegue de Arquitectura de microservicios Cloud Native para un Ecommerce (Doctoral dissertation, Universidad de Antioquia).

Parra-Quero, G., Palma-Urdaneta, O., Torres-Samuel, M. E., & Durán-Garrido, F. (2020). Caracterización de buenas prácticas en la elicitación de requisitos de software referidas en el estándar ISO/IEC/IEEE 29148. *Publicaciones en Ciencias y Tecnología*, 14(2), 91-99.

Paz, J. A. M. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria*, 12(20), 163-176.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Quevedo-Avila, P., Zhindón-Mora, M. G., & Quevedo-Sacoto, A. S. (2020). Arquitectura de microservicios para compras en línea: caso de uso "ala orden". *Polo del Conocimiento*, 5(1), 151-162.

- Rengifo, Y. M. C., Chanchí, G. E. G., Oliveros, D. P., & Campo, W. Y. M. (2020). Aplicativo web para la ejecución de evaluaciones heurísticas de accesibilidad en portales web según la norma NTC 5854. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E26), 501-513.
- Rodríguez, G. D. L. C., Fernández, R. A. M., & Fernández, A. C. M. (2023). Seguridad de la información en el comercio electrónico basado en ISO 27001: Una revisión sistemática. *Innovación y software*, 4(1), 219-236.
- Rodríguez, G. S. (2003). El e-commerce a nivel internacional algunos casos. *Revista de Derecho*, (20), 12-29.
- Rodríguez Noa, D. D. (2022). Aplicación de gestión para planificación de la transportación de cargas y visualización de reportes de Comercio Electrónico en ENCOMIL (Bachelor's thesis, Universidad de las Ciencias Informáticas. Facultad 4).
- Romero Sánchez, R. (2015). Desarrollo de un portal ECommerce en. Net.
- Sachdeva, S. (2016). Scrum methodology. *Int. J. Eng. Comput. Sci*, 5(16792), 16792-16800.
- Sandoval Perez, L. D., & Rojas Pérez, J. J. (2024). Diseño de una arquitectura de microservicios con cloud computing para una aplicación de comercio electrónico.
- Sarmiento, E. D. P. P., & Galán, M. A. C. (2014). Los derechos del consumidor frente al comercio electrónico: un tema sin fronteras, análisis desde el derecho colombiano. *Iustitia*, (12), 279-297.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education Limited.

Taica Sanchez, R. Y. (2023). Análisis comparativo de la norma ISO/IEC 9126 y la norma ISO/IEC 25010 en una aplicación web con los procesos de alojamiento, alimentación, guía y transporte en una organización turística de Pacasmayo–La Libertad.

Tobar, E. G. (2015). Impacto del comercio electrónico en las grandes y medianas empresas de la ciudad de Sogamoso Boyacá, Colombia. *Cuadernos Latinoamericanos de Administración*, 11(20), 83-96.

Vasilica-Maria, M. (2022). Factores que afectan la intención de compra de los consumidores de moda en el comercio electrónico: un modelo teórico para América Latina. *Revista de Investigación Sigma*, 9(01), 84-96.

Velasco Navas, J. P. (2024). Construcción de una plataforma de comercio electrónico basado en microservicios para la gestión, entrega y rastreo de productos a domicilio (Master's thesis).

Villatoro Alas, R. A. (2015). Seguridad en las transacciones en línea de comercio electrónico.

Wang, X., Zhang, Y., Jia, S., & Liu, Y. (2023). Design and Implementation of a Microservices-based Medical Platform. *International Core Journal of Engineering*, 9(6), 143-151.

Zuluaga, L. M. E. Comercio electrónico en Colombia: un mercado pionero amenazado por los gigantes del e-commerce.