



2025

PROYECTO BASE DE DATOS SISTEMA E-COMMERCE VENTA DE ROPA

Autores:

**Manuel Rodríguez
Juan Cruz Vargas
Lucas González
Juan Cruz González
Sebastián Luna**

Introducción

En el contexto del comercio digital, las marcas de indumentaria necesitan soluciones modernas y eficientes para ofrecer una experiencia de compra personalizada. Este proyecto propone el desarrollo de un sistema de comercio electrónico para la venta de ropa de una sola marca, que permita a los usuarios navegar un catálogo, seleccionar talles y colores, gestionar su carrito y realizar pedidos en línea.

El backend, construido con Java 17, Spring Boot y MySQL, expone una API RESTful para manejar la lógica de negocio y la persistencia de datos, asegurando escalabilidad y seguridad. El frontend utiliza React, TypeScript y Bootstrap para ofrecer una interfaz dinámica y responsiva, comunicándose con el backend mediante llamadas a la API.

La base de datos ha sido diseñada y normalizada en 3FN, priorizando integridad referencial y rendimiento. Se incluyen módulos de administración, clientes, productos, stock y pedidos. El documento detalla las entidades clave y las historias de usuario. De esta forma, se busca construir un sistema robusto que respalde la operación y crecimiento del negocio.

Abstract

This document describes the architecture and development of both the backend and frontend for an e-commerce system specializing in the sale of clothing from a single brand. It details the structure of the relational database, its normalization up to the Third Normal Form (3NF), and the design of the main entities and relationships, prioritizing referential integrity and query performance. On the backend, a REST API is implemented using Java and Spring Boot, with Hibernate as the ORM and MySQL as the database engine. On the frontend, React, TypeScript, and Bootstrap are used to build a modern and responsive user interface. The document analyzes the user stories, with a particular focus on data persistence, security (Firebase Authentication), user authentication, and order management.

Keywords: Backend, Spring Boot, JPA, MySQL, E-commerce, Java, REST API, Frontend, React, TypeScript, Bootstrap, Relational Database, Security, MVC Architecture.

Introducción	1
Abstract	1
Objetivo	4
Alcance	4
Límites	4
Tecnología y Herramientas Utilizadas	5
Motor base de datos	5
Lenguajes y Frameworks	5
Backend (Lado del servidor)	5
Frontend (Lado del cliente)	5
Base de Datos	5
Requerimientos	5
Requerimientos Funcionales	5
Requerimientos No Funcionales	6
Tareas	6
Etapa 1: Inicio del Proyecto y Planificación	6
Etapa 2: Diseño de Base de Datos (Simplificada)	6
Etapa 4: Desarrollo de la API REST	7
Etapa 5: Desarrollo del Frontend (React + TypeScript + Bootstrap)	7
Etapa 6: Pruebas y Documentación	7
Diagrama Entidad-Relación	7
Diagrama de Clases	8
Análisis de Entidades Principales	10
1. Módulo de Administración	10
Entidad: Usuario	10
Entidad: Administrador	10
2. Módulo de Información Geográfica y Contacto	10
Entidad: Domicilio	10
Entidad: Localidad, Provincia y País	10
Entidad: Teléfono	11
3. Módulo de Gestión de Clientes	11
Entidad: Cliente	11
Entidad: Domicilio_cliente	11
4. Módulos de Productos y Categorización	11
Entidad: Producto	11
Entidad: Imagen_producto	11
Entidad: Categoria_producto	12
Entidad: Categoría	12
5. Módulo de Inventario y Stock	12
Entidad: Stock	12
Entidad: Color y Talle	12
6. Módulo de Precios Históricos	12

Entidad: Historico_precio_compra	12
Entidad: Historico_precio_venta	13
7. Módulo de Pedidos y Ventas	13
Entidad: Pedido	13
Entidad: Pedido_detalle	13
Análisis de Relaciones Clave	13
Integridad Referencial	13
Escalabilidad del Diseño	14
Optimización y Performance	14
Casos de Uso Soportados	14
Fortalezas del Diseño	14
Historias de Usuario	14
Cliente	14
Historia de Usuario #01 – Registro de un nuevo cliente	14
Historia de Usuario #02 – Inicio de sesión	15
Historia de Usuario #03 – Actualización de datos personales	16
Historia de Usuario #04 – Visualización del catálogo	17
Historia de Usuario #05 – Filtro de productos	18
Historia de Usuario #06 – Ver detalles del producto	18
Historia de Usuario #07 – Comprobar disponibilidad	19
Historia de Usuario #08 – Agregar productos al carrito	19
Historia de Usuario #09 – Realizar pedido	20
Historia de Usuario #10 – Historial de pedidos	20
Administrador	21
Historia de Usuario #11 – Inicio de sesión del administrador	21
Historia de Usuario #12 – Crear productos	22
Historia de Usuario #13 – Editar productos	23
Historia de Usuario #14 – Gestión de stock	23

Objetivo

El objetivo principal es diseñar e implementar una base de datos relacional robusta, normalizada y eficiente para un sistema E-Commerce de ropa, que permita:

- Registrar usuarios y sus datos personales
- Administrar productos con características propias de la indumentaria (talles, colores, stock)
- Clasificar productos por categorías (remeras, pantalones, abrigos, etc.)
- Permitir la gestión de carritos de compra
- Registrar pedidos y sus detalles
- Administrar múltiples direcciones de envío
- Gestionar métodos de pago básicos

Se busca garantizar integridad de los datos, buen rendimiento en las consultas, y una estructura escalable para soportar el crecimiento del catálogo y de los usuarios.

Alcance

El sistema incluirá:

- Registro e inicio de sesión de clientes, e inicio de sesión de administradores (el registro de los administradores lo realizarán los creadores del sistema)
- Gestión de roles: cliente, administrador
- Catálogo de productos con filtros por categoría, talle y color
- Gestión de stock por variante (talle + color). Y manejo de existencias.
- Administración de categorías de productos
- Carrito de compras Persistente con reservas del Pedido
- Registro de pedidos con estado (pendiente, enviado, entregado, cancelado).
- El Cliente puede agregar elementos al carrito sin estar logueado. Si desea comprar se le pedirá que se loguee/registre.
- Múltiples direcciones de envío por usuario
- Historial de Pedidos de Usuario
- Métodos de pago simulados (mercado pago)
- Panel de administración para gestionar productos, pedidos y usuarios

Desde el punto de vista técnico:

- Modelo entidad-relación (DER) completo
- Consultas SQL (stock bajo, historial de pedidos, etc.)
- Normalización de la base hasta 3FN
- Claves primarias, foráneas e integridad referencial correctamente definidas

Límites

Este proyecto no incluirá:

- Pagos reales, se realizará una simulación de pago con cuentas de prueba.
- Funcionalidades avanzadas de marketing (descuentos automáticos, recomendaciones por IA, cupones)
- Gestión logística con servicios externos (Correo Argentino, Andreani, etc.). No manejamos envíos.
- Soporte multilinguaje o multicurrency

- Desarrollo de interfaz gráfica completa
- Personalización de productos por el cliente (bordados, impresiones, etc.)

Tecnología y Herramientas Utilizadas

Motor base de datos

El sistema de gestión de bases de datos que se utilizará es MySQL, una solución confiable, eficiente y ampliamente adoptada en entornos productivos. Nos permitirá estructurar, almacenar y gestionar toda la información relevante del sistema, como usuarios, productos, pedidos y pagos.

Lenguajes y Frameworks

Backend (Lado del servidor)

Para la lógica de negocio y la gestión de los datos utilizaremos:

- Java como lenguaje principal de programación.
- Spring Boot, un framework que facilita el desarrollo rápido de aplicaciones web robustas y modulares.
- Spring Data JPA para la persistencia de datos, permitiendo trabajar con la base de datos a través de objetos y entidades de forma eficiente.
- Spring Web (REST API) para exponer servicios que serán consumidos desde el frontend.
- Hibernate como implementación por defecto de JPA, facilitando el mapeo objeto-relacional (ORM).
- Firebase Authentication como servicio para el registro y login de usuarios.
- Checkout Pro de Mercado Pago para el manejo de links de pago con cuentas de prueba.
- Cloudinary como servicio de carga de imágenes a la nube de los productos.

Frontend (Lado del cliente)

La interfaz visual y la interacción con el usuario se desarrollará con:

- React como biblioteca principal para construir interfaces de usuario interactivas y dinámicas.
- TypeScript para mejorar la calidad del código y reducir errores durante el desarrollo, gracias a su tipado estático.
- Frameworks de estilo (Bootstrap) para el diseño visual, brindando una apariencia moderna, adaptativa y atractiva.

Base de Datos

- MySQL será el motor encargado de almacenar toda la información del sistema, como catálogos de productos, datos de clientes, historial de pedidos, medios de pago, etc.
- Workbench o cualquier cliente de SQL puede ser usado como herramienta visual para modelar y consultar la base de datos.

Requerimientos

Requerimientos Funcionales

1. Registro y autenticación de usuarios
El sistema permitirá a los usuarios registrarse con sus datos personales y autenticarse mediante correo electrónico y contraseña.
2. Visualización y gestión de productos en la tienda online
Los usuarios podrán visualizar un catálogo de productos con filtros por categoría, talle y color. Los administradores podrán agregar, modificar y eliminar productos desde el panel de administración.
3. Sistema de carrito de compras
Cada usuario podrá agregar productos al carrito, visualizar el contenido del mismo y modificar cantidades o eliminar ítems antes de confirmar la compra.
4. Generación de pedidos y procesamiento de pagos
El sistema permitirá al usuario realizar un pedido, seleccionando dirección de envío y método de pago (simulado). Cada pedido incluirá sus productos, cantidades, estado y fecha.
5. Historial de compras por usuario
Los usuarios podrán consultar el historial de sus pedidos anteriores, con detalles como fecha, estado, productos comprados y total abonado.
6. Panel de administración para la gestión de productos y ventas
Los administradores tendrán acceso a un panel para gestionar productos, ver pedidos realizados por los clientes y modificar su estado.

Requerimientos No Funcionales

1. Normalización de la base de datos:
 - a. La base de datos deberá estar correctamente normalizada hasta la Tercera Forma Normal (3FN) para evitar redundancias y garantizar integridad.
2. Arquitectura cliente-servidor
 - a. El sistema debe desarrollarse bajo una arquitectura cliente-servidor, donde el frontend (React + TypeScript) se comunique con el backend (Java + Spring Boot) mediante APIs REST.

Tareas

Etapa 1: Inicio del Proyecto y Planificación

1. Definir alcance, objetivos, funcionalidades y límites del sistema
2. Analizar requerimientos funcionales y no funcionales
3. Identificar entidades principales (usuarios, productos, pedidos, etc.)
4. Elaborar reglas de negocio y glosario de términos

Etapa 2: Diseño de Base de Datos (Simplificada)

1. Diseñar el modelo entidad-relación (DER)
2. Aplicar normalización hasta la Tercera Forma Normal (3FN)
3. Establecer claves primarias, foráneas e integridad referencial
4. Validar el diseño con ejemplos o casos reales

5. Crear scripts SQL de definición de tablas (DDL)

Etapa 3: Configuración del Backend (Spring Boot + JPA)

1. Crear proyecto Spring Boot y conectarlo con MySQL
2. Configurar archivo application.properties para la conexión
3. Crear entidades Java con anotaciones JPA (Usuario, Producto, Pedido, etc.)
4. Crear repositorios (interfaces) con Spring Data JPA
5. Implementar servicios con lógica de negocio básica (alta, baja, modificación, búsqueda)

Etapa 4: Desarrollo de la API REST

1. Crear controladores REST para usuarios, productos, carritos y pedidos
2. Implementar endpoints para login, registro, listado de productos, creación de pedidos, etc.
3. Probar APIs con Postman o alguna herramienta similar
4. Manejar errores y validar datos en los endpoints

Etapa 5: Desarrollo del Frontend (React + TypeScript + Bootstrap)

1. Inicializar proyecto en React y estructurar carpetas
2. Diseñar componentes para Home, Catálogo, Detalle de Producto, Carrito, etc.
3. Consumir la API REST desde el frontend (Fetch)
4. Implementar navegación con React Router
5. Agregar validaciones de formularios (login, registro, compra)

Etapa 6: Pruebas y Documentación

1. Realizar pruebas funcionales del sistema completo
2. Documentar el modelo de datos y principales consultas SQL
3. Documentar la API REST

Diagrama Entidad-Relación

El siguiente diagrama representa una parte del modelo Entidad-Relación para el sistema E-Commerce de venta de ropa. En esta sección se modela la relación entre los clientes y los productos a través de la transacción Comprar, que refleja la posibilidad de que un cliente adquiera múltiples productos y que cada producto pueda ser comprado por varios clientes. Esta estructura es clave para registrar las compras realizadas y servirá como base para construir el historial de pedidos del sistema.

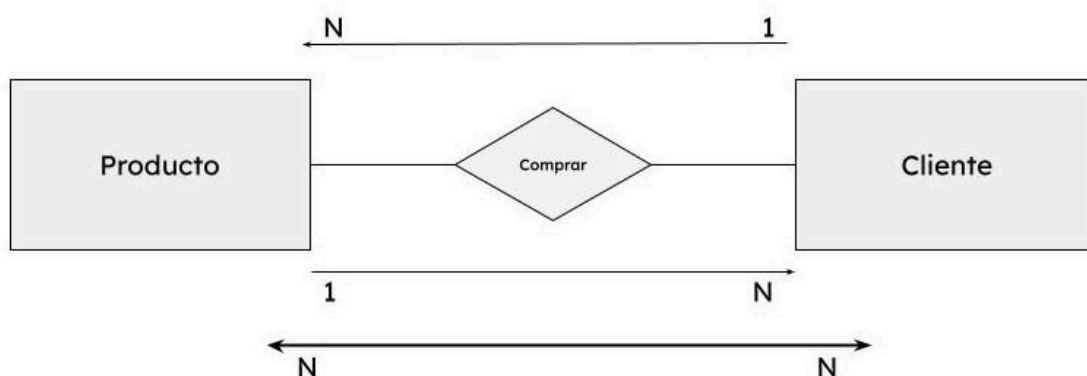
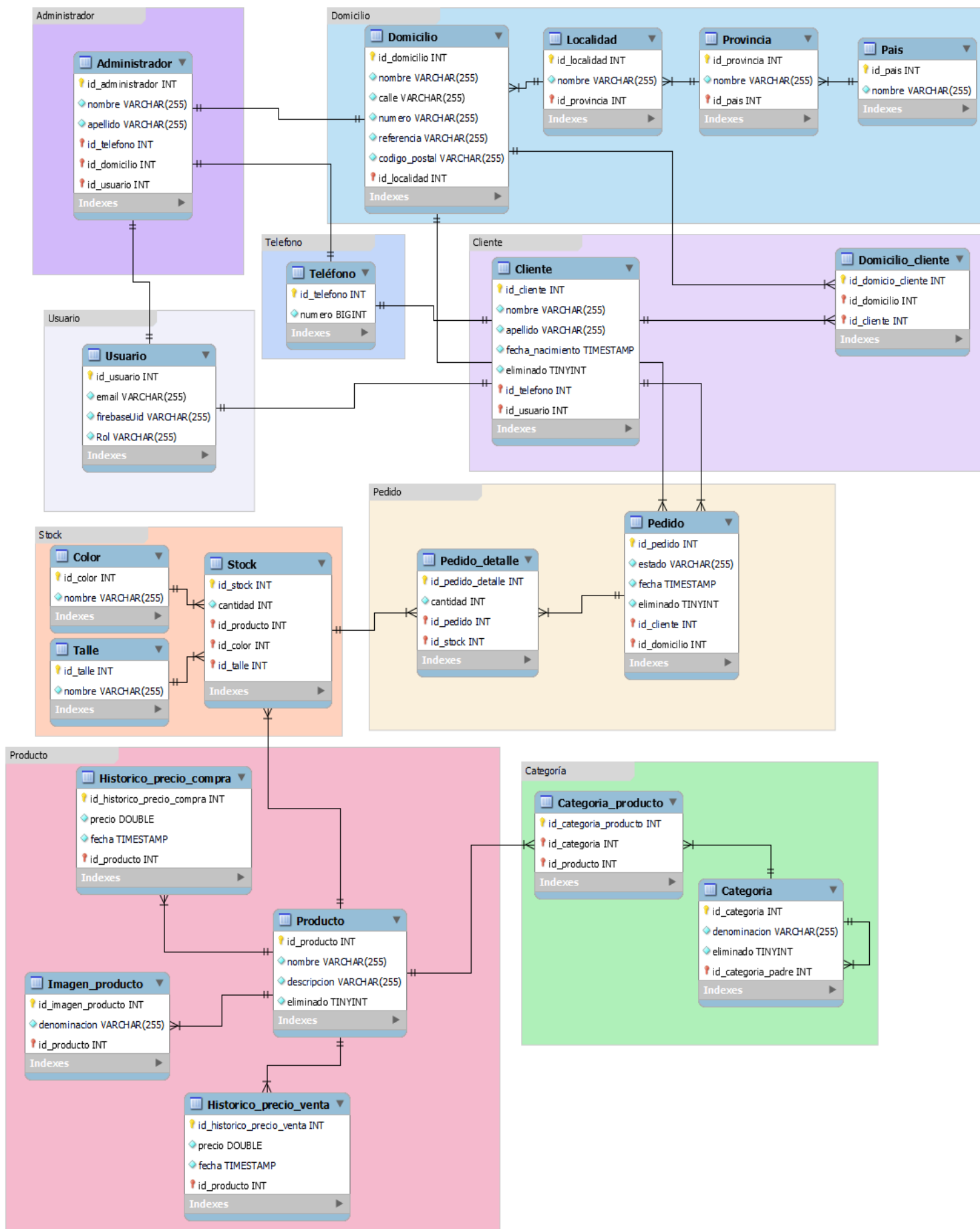


Diagrama de Clases

El siguiente Diagrama de Clases forma parte del diseño conceptual del sistema E-Commerce para la venta de ropa de una sola marca. Este modelo extendido permite representar no solo entidades y relaciones, sino también detalles como claves primarias, claves foráneas y el tipo de participación de cada entidad en las relaciones.

En particular, se muestra cómo los clientes interactúan con los productos para generar una compra, reflejando la multiplicidad de compras posibles en un entorno real, generando registro de múltiples pedidos. Este diagrama constituye la base sobre la cual se desarrollará el esquema relacional definitivo que alimentará la base de datos del sistema. Además, el modelo contempla la participación de un administrador, quienes gestionan el catálogo de productos, categorías y usuarios dentro del sistema.

Este diagrama ha sido construido respetando los principios de la Tercera Forma Normal (3NF), con el fin de eliminar redundancias, asegurar la integridad de los datos y optimizar el rendimiento en las consultas.



Análisis de Entidades Principales

1. Módulo de Administración

Entidad: Usuario

- Constituye el núcleo del sistema de autenticación y autorización
- Atributos principales:
 - **id_usuario** (INT, PK): Identificador único del usuario.
 - **email** (VARCHAR(255)): Credencial de acceso único.
 - **firebaseUid** (VARCHAR(255)): Credencial de seguridad.
 - **Rol** (VARCHAR(255)): Rol del usuario
- Relación: Se conecta con la entidad **Administrador** mediante clave foránea, estableciendo una relación 1:1 que separa las credenciales de acceso de los datos personales del administrador.
- También puede extenderse para roles diferenciados en el futuro (cliente o administrador).

Entidad: Administrador

- Almacena la información personal y profesional de los administradores del sistema
- Atributos clave:
 - **id_administrador** (INT, PK): Identificador único
 - **nombre** (VARCHAR(255)): Nombre completo del administrador
 - **apellido** (VARCHAR(255)): Apellido del administrador
 - **id_telefono** (INT, FK): Referencia a información de contacto
 - **id_domicilio** (INT, FK): Referencia a dirección física
 - **id_usuario** (INT, FK): Referencia a usuario

2. Módulo de Información Geográfica y Contacto

Entidad: Domicilio

- Gestiona las direcciones físicas del sistema
- Estructura jerárquica de ubicación:
 - **id_domicilio** (INT, PK): Identificador único
 - **calle** (VARCHAR(255)): Vía principal
 - **numero** (VARCHAR(255)): Numeración
 - **referencia** (VARCHAR(255)): Referencias adicionales
 - **codigo_postal** (VARCHAR(255)): Código postal
 - **id_localidad** (INT, FK): Conexión con localidad

Entidad: Localidad, Provincia y País

- **Localidad** conecta con **Provincia** (**id_provincia**, FK)
- **Provincia** conecta con **País** (**id_pais**, FK)

Entidad: Teléfono

- Gestiona información de contacto telefónico
- Atributos:
 - **id_telefono** (INT, PK): Identificador único
 - **numero** (BIGINT): Número de teléfono

3. Módulo de Gestión de Clientes

Entidad: Cliente

- Almacena información personal de los clientes
- Estructura similar a **Administrador** pero orientada al cliente final
- Incluye referencias a domicilio y teléfono para envíos y contacto
- **Atributos clave:**
 - **id_cliente** (INT, PK): Identificador único.
 - **nombre** (VARCHAR(255)): Apellido del cliente.
 - **apellido** (VARCHAR(255)): Nombre del cliente.
 - **fecha_nacimiento** (TIMESTAMP): Fecha de nacimiento del cliente.
 - **eliminado** (TINYINT): Control de borrado lógico.
 - **id_telefono** (INT, FK): Teléfono del cliente.
 - **id_usuario** (INT, FK): Usuario del cliente (con el que se Loguea en la aplicacion).

Entidad: Domicilio_cliente

- Relaciona domicilio con cliente
- Atributos:
 - **id_domicilio_cliente** (INT, PK): Identificador único
 - **id_domicilio** (INT, FK): Referencia al domicilio
 - **id_cliente** (INT, FK): Referencia al cliente

4. Módulos de Productos y Categorización

Entidad: Producto

- Núcleo del catálogo comercial
- Atributos fundamentales:
 - **id_producto** (INT, PK): Identificador único
 - **nombre** (VARCHAR(255)): Denominación comercial
 - **descripcion** (VARCHAR(255)): Descripción detallada
 - **eliminado** (TINYINT): Flag de borrado lógico

Entidad: Imagen_producto

- Gestiona recursos visuales de los productos
- Atributos:
 - **id_imagen_producto** (INT, PK): Identificador único

- **denominacion** (VARCHAR(255)): Nombre del archivo
 - **id_producto** (INT, FK): Vinculación con producto específico
- Relación: Un producto puede tener múltiples imágenes (1:N)

Entidad: Categoria_producto

- Sistema de clasificación de productos
- Permite organización jerárquica del catálogo
- Atributos:
 - **id_categoria_producto** (INT, PK): Identificador único
 - **id_categoria** (INT, FK): Referencia a la categoria
 - **id_producto** (INT, FK): Referencia al producto
- Relación: Un producto pertenece a una categoría, una categoría puede tener múltiples productos (1:N).

Entidad: Categoría

- Define las clasificaciones disponibles
- Estructura:
 - **id_categoria** (INT, PK): Identificador único
 - **denominacion** (VARCHAR(255)): Nombre de la categoría
 - **eliminado** (TINYINT): Control de borrado lógico
 - **id_categoria_padre** (INT, FK)

5. Módulo de Inventario y Stock

Entidad: Stock

- Control de inventario por producto
- Atributos críticos:
 - **id_stock** (INT, PK): Identificador único
 - **cantidad** (INT): Unidades disponibles
 - **id_producto** (INT, FK): Producto asociado
 - **id_color** (INT, FK): Variante de color
 - **id_talle** (INT, FK): Variante de talla

Entidad: Color y Talle

- Definen variantes de productos
- Permiten gestión de inventario granular por características específicas
- Relación: Cada registro de stock puede asociarse a combinaciones específicas de color y talle.
- **Color** y **Talle** poseen:
 - **nombre** (VARCHAR (255))

6. Módulo de Precios Históricos

Entidad: Historico_precio_compra

- Registra evolución de precios de adquisición
- Atributos temporales:
 - **id_historico_precio_compra** (INT, PK): Identificador único
 - **precio** (DOUBLE): Valor de compra
 - **fecha** (TIMESTAMP): Momento de registro
 - **id_producto** (INT, FK): Referencia al producto
- Propósito: Análisis de costos y rentabilidad histórica

Entidad: Historico_precio_venta

- Registra evolución de precios de venta
- Estructura similar a precios de compra
- Atributos:
 - **id_historico_precio_venta** (INT, PK): Identificador único
 - **precio** (DOUBLE): Valor de venta
 - **fecha** (TIMESTAMP): Momento de registro
 - **id_producto** (INT, FK): Referencia al producto
- Propósito: Análisis de estrategias de pricing y rentabilidad

7. Módulo de Pedidos y Ventas

Entidad: Pedido

- Gestiona órdenes de compra de clientes
- Atributos de control:
 - **id_pedido** (INT, PK): Identificador único
 - **estado** (VARCHAR(255)): Estado del pedido (pendiente, procesando, enviado, etc.)
 - **fecha** (TIMESTAMP): Momento de creación
 - **eliminado** (TINYINT): Control de borrado lógico
 - **id_cliente** (INT, FK): Cliente que realizó el pedido
 - **id_domicilio** (INT, FK): Domicilio de entrega

Entidad: Pedido_detalle

- Detalla productos específicos dentro de cada pedido
- Implementa relación muchos a muchos entre pedidos y productos
- Atributos:
 - **id_pedido_detalle** (INT, PK): Identificador único
 - **cantidad** (INT): Unidades solicitadas
 - **precio** (INT): precio unitario de las unidades solicitadas
 - **id_pedido** (INT, FK): Pedido al que pertenece
 - **id_stock** (INT, FK): Stock solicitado

Análisis de Relaciones Clave

Integridad Referencial

El diagrama implementa un sistema robusto de claves foráneas que garantiza:

- Consistencia geográfica: La jerarquía País → Provincia → Localidad → Domicilio
- Trazabilidad de productos: Desde categoría hasta stock específico
- Auditabilidad de precios: Historial completo de variaciones de precios
- Integridad de pedidos: Relación completa cliente → pedido → detalle → producto

Escalabilidad del Diseño

- Borrado lógico: Uso de campos **eliminado** para mantener integridad histórica
- Flexibilidad geográfica: Estructura que soporta expansión internacional
- Variantes de producto: Sistema de colores y talles extensible

Optimización y Performance

- Índices implícitos: Claves primarias automáticamente indexadas
- Normalización adecuada: Evita redundancia de datos
- Separación de concerns: Módulos independientes pero interconectados

Casos de Uso Soportados

1. Gestión de Catálogo: Creación, modificación y organización de productos con sus variantes
2. Procesamiento de Pedidos: Flujo completo desde selección hasta entrega
3. Control de Inventario: Seguimiento en tiempo real de stock por variante
4. Análisis de Precios: Histórico completo para análisis de rentabilidad
5. Gestión de Clientes: Perfil completo con historial de compras
6. Administración del Sistema: Control de acceso y gestión operativa

Fortalezas del Diseño

1. Estructura normalizada que evita redundancia
2. Flexibilidad para futuras extensiones
3. Separación clara entre datos operativos y de configuración
4. Sistema de auditoría mediante timestamps y flags de eliminación

Historias de Usuario

Cliente

Historia de Usuario #01 – Registro de un nuevo cliente

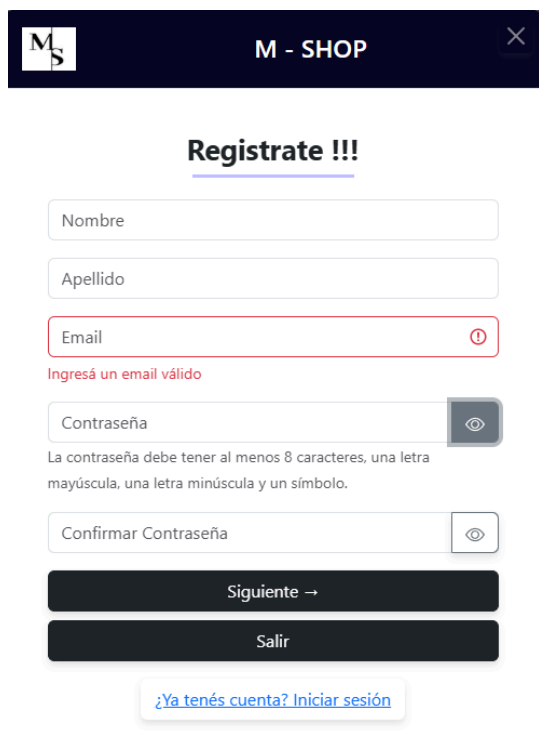
Como: Cliente

Quiero: Registrarme en la plataforma con mis datos personales

Para: Poder realizar compras

Criterios de aceptación:

- El formulario de registro debe incluir campos obligatorios como nombre, correo electrónico, contraseña y domicilio.
- El sistema debe validar que el correo electrónico no esté previamente registrado.



The screenshot shows a web interface for 'M - SHOP' with a registration form titled 'Regístrate !!!'. The form includes input fields for 'Nombre', 'Apellido', 'Email', 'Contraseña', and 'Confirmar Contraseña'. The 'Email' field has a red border and an error icon, with a message below it: 'Ingresá un email válido'. The 'Contraseña' field has a toggle icon and a message: 'La contraseña debe tener al menos 8 caracteres, una letra mayúscula, una letra minúscula y un símbolo.' Below the form are two buttons: 'Siguiente →' and 'Salir'. At the bottom, there is a link: '¿Ya tenés cuenta? Iniciar sesión'.

Img.1 Registro

Historia de Usuario #02 – Inicio de sesión

Como: Cliente

Quiero: Iniciar sesión con mi usuario y contraseña

Para: Acceder a mi cuenta

Criterios de aceptación:

- El cliente debe poder ingresar con su correo y contraseña previamente registrados.

- El sistema debe mostrar un mensaje de error si las credenciales son incorrectas.
- Se debe mantener la sesión activa hasta que el usuario cierre sesión manualmente o se desconecte por inactividad.



Img 2 Inicio Sesión

Historia de Usuario #03 – Actualización de datos personales

Como: Cliente

Quiero: Actualizar mis datos de contacto y domicilio

Para: Mantener mi información actualizada

Criterios de aceptación:

- El cliente debe poder modificar campos como dirección, teléfono y correo electrónico.
- El sistema debe validar el formato correcto de los datos.
- Los cambios deben reflejarse inmediatamente en su perfil.



Img. 3 Perfil

Cambiar Datos Personales

Nombre

Juan

Apellido

Hurtado

Fecha de nacimiento:

10/06/2001

261-123-4567

El número debe tener 10 dígitos, sin el 15 y con el código de área.

Cancelar

Guardar Cambios

Img. 4 Cambio datos personales

Historia de Usuario #04 – Visualización del catálogo

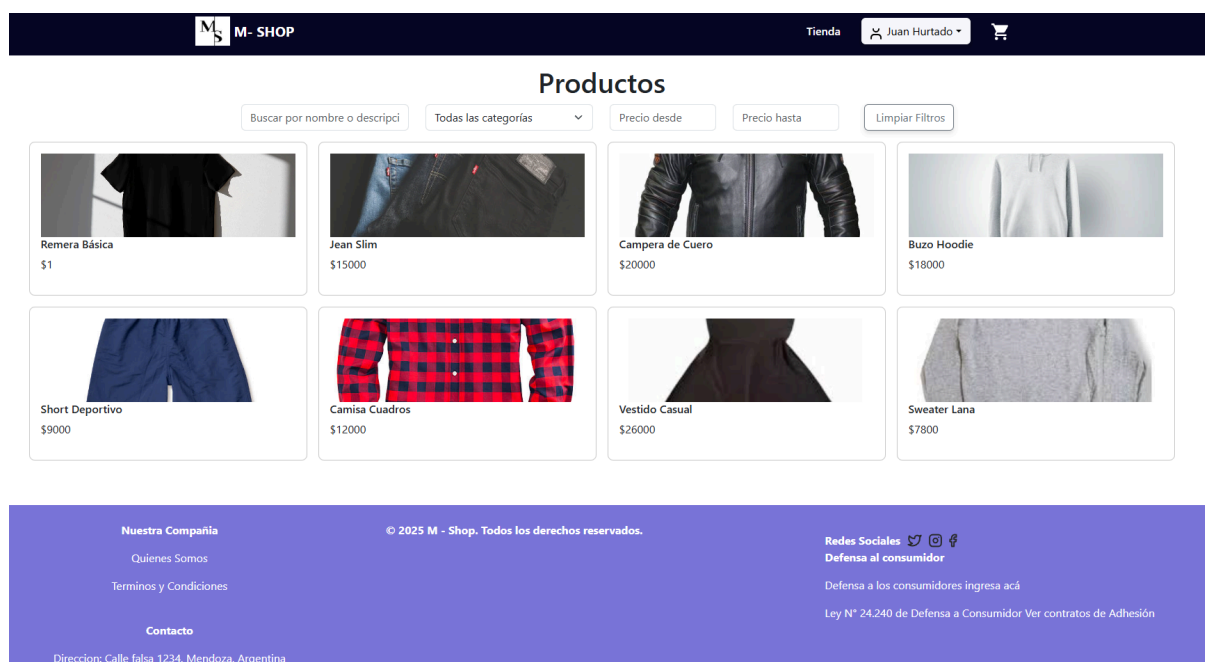
Como: Cliente

Quiero: Ver el catálogo de productos organizados por categorías

Para: Encontrar fácilmente lo que busco

Criterios de aceptación:

- El catálogo debe estar dividido por categorías
- El cliente debe poder navegar entre las categorías y ver los productos correspondientes.
- Los productos deben mostrar una imagen, nombre, precio.



Img. 5 Catálogo

Historia de Usuario #05 – Filtro de productos

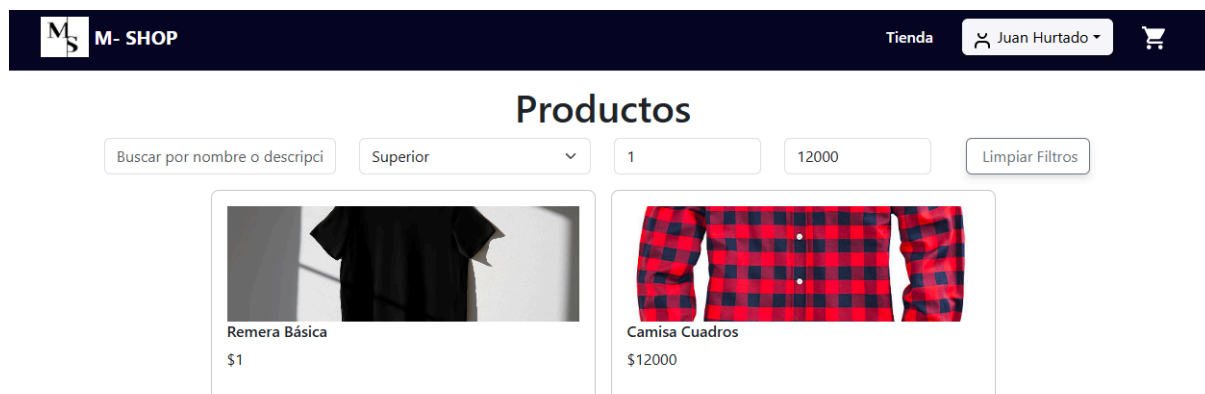
Como: Cliente

Quiero: Filtrar productos por tipo

Para: Encontrar exactamente lo que necesito

Criterios de aceptación:

- Los filtros deben estar visibles y ser fáciles de usar.
- El sistema debe actualizar los resultados en tiempo real según los filtros aplicados.



Img. 6 Filtro Catálogo

Historia de Usuario #06 – Ver detalles del producto

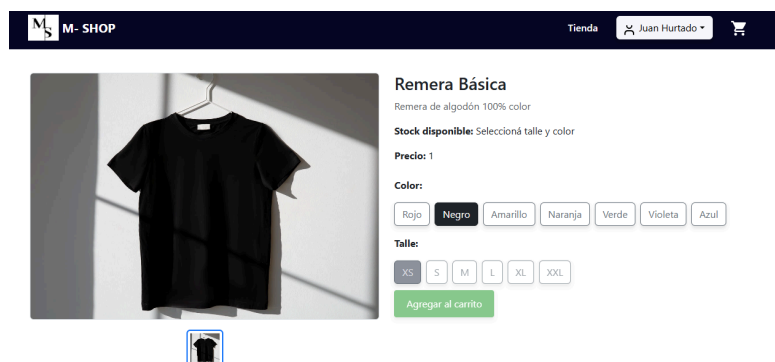
Como: Cliente

Quiero: Ver las imágenes y detalles de un producto

Para: Conocer sus características antes de comprarlo

Criterios de aceptación:

- Al seleccionar un producto, se debe abrir una vista con información detallada (material, descripción, precio, imágenes, disponibilidad).
- El cliente debe poder seleccionar combinaciones de color y talle disponibles.



Img. 7 Detalles Producto

Historia de Usuario #07 – Comprobar disponibilidad

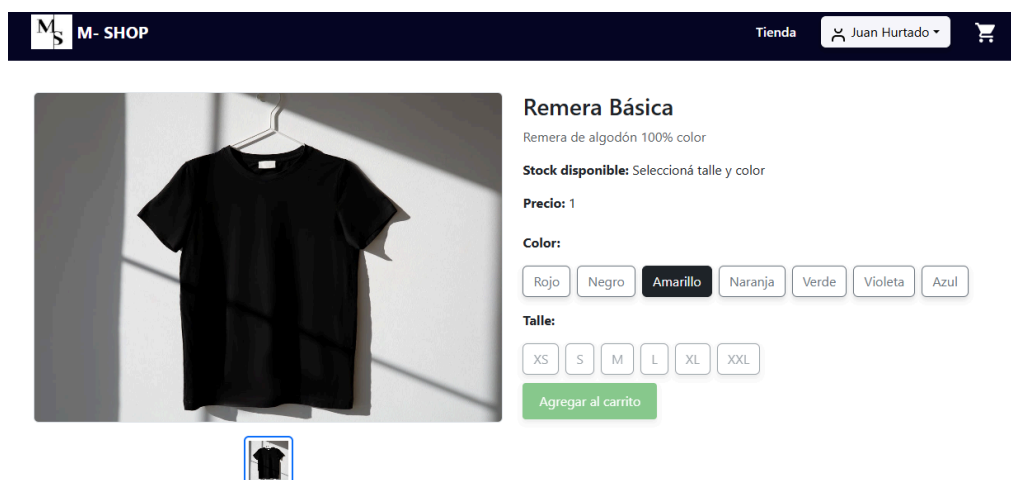
Como: Cliente

Quiero: Verificar la disponibilidad de stock por talla y color

Para: Saber si puedo agregar el producto al carrito

Criterios de aceptación:

- Debe mostrarse la disponibilidad por cada combinación de color y talla.
- Si una combinación no está disponible, deshabilita los botones.
- El sistema no debe permitir agregar combinaciones sin stock al carrito.



Img. 8 Disponibilidad

Historia de Usuario #08 – Agregar productos al carrito

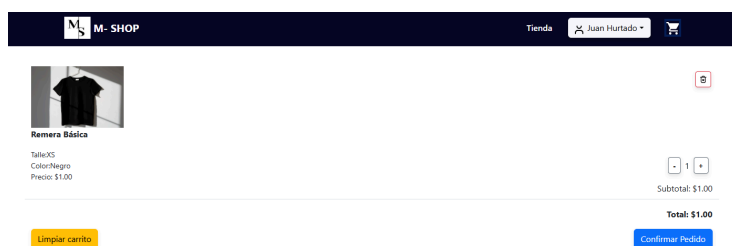
Como: Cliente

Quiero: Agregar productos al carrito de compras

Para: Realizar una compra múltiple

Criterios de aceptación:

- El cliente debe poder seleccionar talla y color antes de agregar un producto al carrito.
- El sistema debe mostrar un resumen del carrito.



Img. 9 Carrito

Historia de Usuario #09 – Realizar pedido

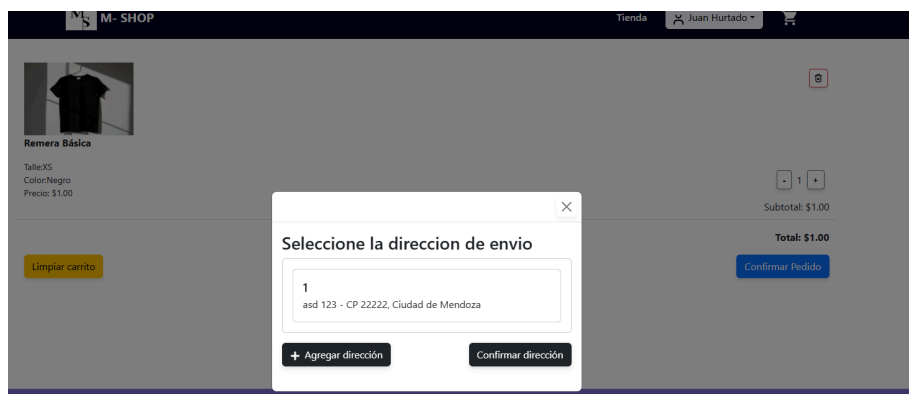
Como: Cliente

Quiero: Realizar un pedido seleccionando productos del carrito

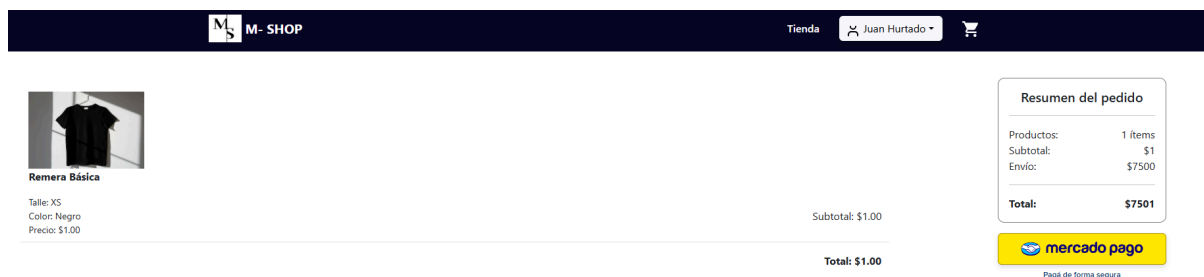
Para: Completar mi compra

Criterios de aceptación:

- El cliente debe poder confirmar dirección de envío y método de pago.
- El sistema debe validar la disponibilidad del stock al momento de confirmar la compra.
- Se debe generar un número de pedido y una confirmación de compra.



img. 10 Confirmación domicilio



Img. 11 Confirmación Pago

Historia de Usuario #10 – Historial de pedidos

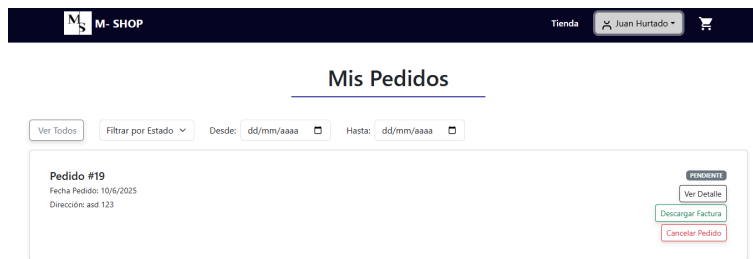
Como: Cliente

Quiero: Ver el historial de mis pedidos

Para: Hacer seguimiento de mis compras

Criterios de aceptación:

- El cliente debe poder acceder a una lista de todos sus pedidos anteriores.
- Cada pedido debe mostrar fecha, productos comprados, monto total y estado actual.
- Se debe poder visualizar los detalles individuales de cada pedido.
- Si el estado está en pendiente podrá cancelarlo caso contrario no.



img. 12 Pedidos

Administrador

Historia de Usuario #11 – Inicio de sesión del administrador

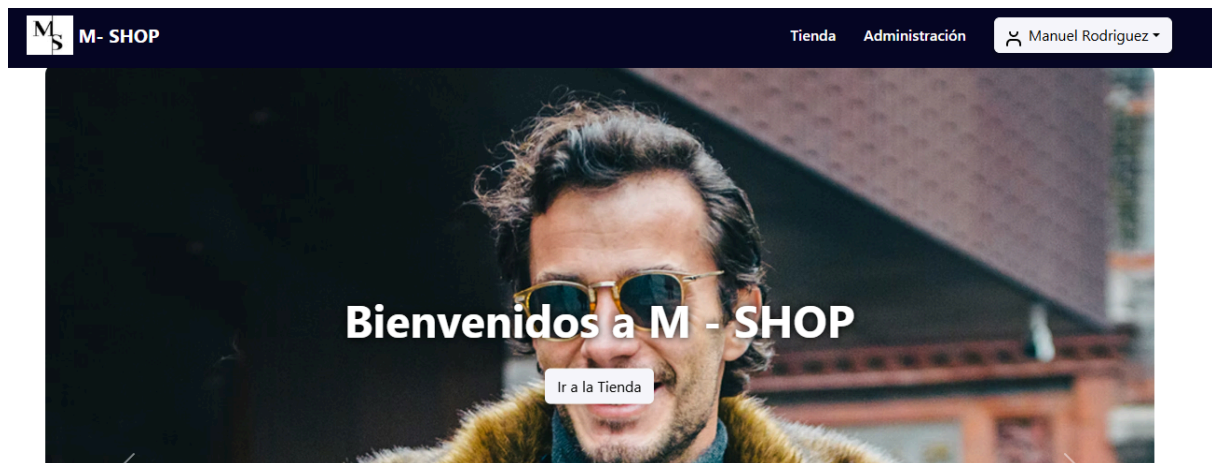
Como: Administrador

Quiero: Iniciar sesión con mis credenciales

Para: Acceder al panel de administración

Criterios de aceptación:

- El sistema debe validar las credenciales del administrador antes de permitir el acceso.
- Debe haber control de sesiones activas por seguridad.



Img. 13 Vista principal admin.



Img. 14 Panel Administrador

Historia de Usuario #12 – Crear productos

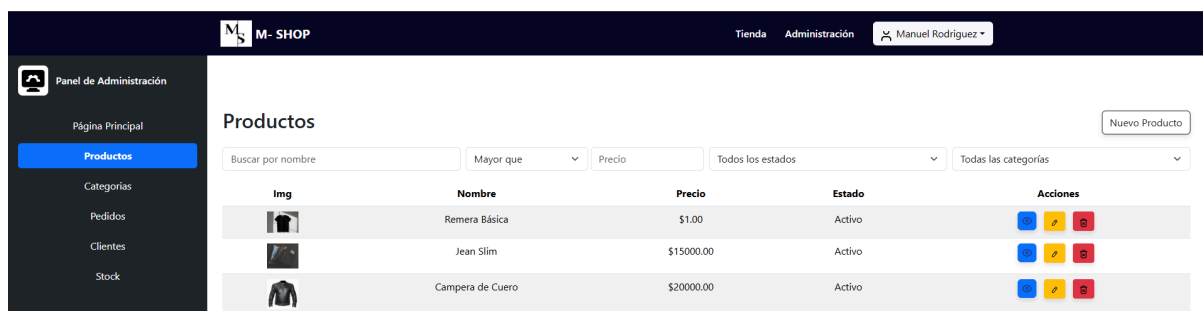
Como: Administrador

Quiero: Crear nuevos productos con sus características

Para: Ampliar el catálogo

Criterios de aceptación:

- El administrador debe poder cargar nombre, descripción, precio, talla
- Debe existir validación de campos obligatorios antes de guardar.
- El nuevo producto debe aparecer automáticamente en el catálogo público tras su creación.



Img. 15 Panel Productos

Crear Producto

Nombre:

Descripción:

Precio:

Categorías:

<input type="checkbox"/> Superior	<input type="checkbox"/> Inferior	<input type="checkbox"/> Abrigos
<input type="checkbox"/> Accesorios	<input type="checkbox"/> Camisas	<input type="checkbox"/> Buzos
<input type="checkbox"/> Pantalones	<input type="checkbox"/> Shorts	<input type="checkbox"/> Camperas
<input type="checkbox"/> Sacos	<input type="checkbox"/> Gorros	<input type="checkbox"/> Remeras

Imágenes:

Sin archivos seleccionados

Estado:

Img. 16 Modal Creación Producto

Historia de Usuario #13 – Editar productos

Como: Administrador

Quiero: Editar información de productos existentes

Para: Mantener el catálogo actualizado

Criterios de aceptación:

- El administrador debe poder buscar un producto y modificar sus datos.
- Los cambios deben reflejarse en tiempo real en el catálogo.
- El sistema debe mantener el historial de modificaciones.

Actualizar Producto

Nombre:


Descripción:

Precio:

Categorías:

<input checked="" type="checkbox"/> Superior	<input type="checkbox"/> Inferior	<input type="checkbox"/> Abrigos
<input type="checkbox"/> Accesorios	<input type="checkbox"/> Camisas	<input type="checkbox"/> Buzos
<input type="checkbox"/> Pantalones	<input type="checkbox"/> Shorts	<input type="checkbox"/> Camperas
<input type="checkbox"/> Sacos	<input type="checkbox"/> Gorros	<input checked="" type="checkbox"/> Remeras

Imágenes:
 Sin archivos seleccionados



Estado:

Img. 17 Edición

Historia de Usuario #14 – Gestión de stock

Como: Administrador

Quiero: Gestionar el stock de productos por talla y color

Para: Controlar el inventario

Criterios de aceptación:

- El administrador debe poder actualizar las cantidades disponibles de cada combinación de talla y color.
- Si el stock de una combinación se agota, debe reflejarse como “Sin stock” en el catálogo.

Gestión de Stock

Crear Stock

Producto	Color	Talle	Cantidad	Acciones	
Remera Básica	Rojo	S	112	<button>Agregar</button>	<button>Actualizar</button>
Remera Básica	Negro	XS	98	<button>Agregar</button>	<button>Actualizar</button>
Jean Slim	Rojo	XS	100	<button>Agregar</button>	<button>Actualizar</button>
Campera de Cuero	Negro	S	100	<button>Agregar</button>	<button>Actualizar</button>
Buzo Hoodie	Rojo	S	100	<button>Agregar</button>	<button>Actualizar</button>
Jean Slim	Rojo	L	100	<button>Agregar</button>	<button>Actualizar</button>
Campera de Cuero	Negro	M	99	<button>Agregar</button>	<button>Actualizar</button>
Short Deportivo	Rojo	M	80	<button>Agregar</button>	<button>Actualizar</button>
Camisa Cuadros	Azul	XXL	10	<button>Agregar</button>	<button>Actualizar</button>

Img. 18 Panel Stock

Agregar Stock

Producto
Remera Básica

Color
Rojo

Talle
S

Seleccione un talle
XS
S
M
L
XL
XXL

Cancelar Guardar

Img. 19 Edición stock

Historia de Usuario #15 – Gestión de imágenes

Como: Administrador

Quiero: Subir y gestionar imágenes de productos

Para: Mostrar los artículos correctamente

Criterios de aceptación:

- El administrador debe poder subir múltiples imágenes por producto.
- Debe existir un orden de visualización (por ejemplo, imagen principal).
- Las imágenes deben mostrarse en alta calidad y permitir su reemplazo o eliminación.

Imágenes:

Elegir archivos Sin archivos seleccionados



Estado:

Img. 20 Subida de imágenes

Historia de Usuario #16 – Gestión de categorías

Como: Administrador

Quiero: Crear y organizar categorías de productos

Para: Estructurar el catálogo

Criterios de aceptación:

- El administrador debe poder crear, renombrar o eliminar categorías.
- Los productos deben poder ser asignados a una o más categorías.
- Las categorías deben mostrarse organizadas y reflejarse en el catálogo público.

Categorías

Nueva Categoría

Superior	↑			
Remeras				
Buzos				
Abrigos	↑			
Sacos				
Camperas				
Camisas				

Img. 21 Categorías

Historia de Usuario #17 – Ver pedidos realizados

Como: Administrador

Quiero: Ver todos los pedidos realizados

Para: Gestionar las ventas

Criterios de aceptación:

- Debe existir un listado de pedidos con filtros por fecha, estado y cliente.
- Cada pedido debe mostrar sus detalles: productos, montos, dirección de envío y estado actual.
- El administrador podrá descargar la factura.

Gestión de Pedidos

Todos los estados
dd/mm/aaaa
dd/mm/aaaa
Limpiar Filtros

N° Pedido	Cliente	Domicilio	Fecha	Estado	Acciones			
# 19	Juan Hurtado	asd 123, Ciudad de Mendoza 1	10/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 9	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 10	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 11	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 12	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 13	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 14	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PENDIENTE	Ver	Factura	PENDIENTE	Actualizar
# 15	Juan Cruz Gonzalez	Bahia Blanca 3256, Ciudad de Mendoza casa	9/6/2025	PROCESANDO	Ver	Factura	PROCESANDO	Actualizar

Img. 22 Pedidos admin

Historia de Usuario #18 – Cambiar estado de pedidos

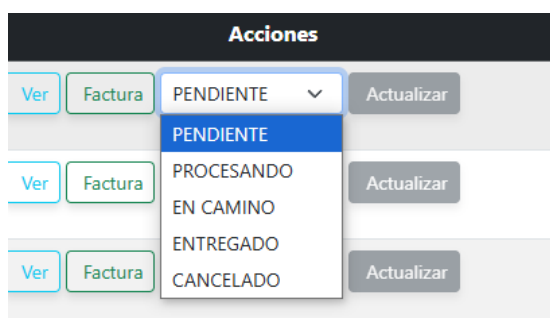
Como: Administrador

Quiero: Cambiar el estado de los pedidos

Para: Informar el progreso a los clientes

Criterios de aceptación:

- El sistema debe permitir cambiar estados como: “Pendiente”, “Procesado”, “En camino”, “Entregado”, “Cancelado”.
- El cambio de estado se debe notificar al cliente por correo electrónico (opcional).
- Debe registrarse la fecha y hora del cambio de estado.



Img. 23 Estados

Historia de Usuario #19 – Consultar histórico de precios

Como: Administrador

Quiero: Consultar el histórico de precios de productos

Para: Analizar tendencias de mercado

Criterios de aceptación:

- Cada producto debe tener un historial con fecha y valor de cada modificación de precio.

Fecha	Precio
2025-06-10T13:06:19	\$1
2025-06-10T13:05:25	\$9000
2025-06-10T13:02:31	\$10000

Img. 24 Historicos