

**Docente:**

Jhonny Edgard VALVERDE PARDAVE

**FACULTAD DE INGENIERIA**

**Carrera:**

Ingeniería de Sistemas e Informática

## **Curso Integrador I: Sistemas Software**

**Avance de Proyecto Final 3**

**Semana 12**

**Integrantes:**

MEJÍA COLLAZOS, Roberto Agustín

PECHO SANTOS, Manuel Angel

SOTOMAYOR BETETA, Daniel

VELASQUEZ YSUIZA, Miguel Angel

Lima, 14 de junio de 2025

## Contenido

CAPITULO I .....	3
Presentación del caso propuesto.....	3
Modelo lean Canvas del caso propuesto.....	4
Diagrama de Canvas.....	1
1. El proyecto del caso propuesto. ....	1
1.1 Project charter del proyecto.....	1
1.2 Desglose de la estructura de tareas WBS.....	1
1.3 Diagrama de Gantt .....	1
Requerimientos.....	2
1.4 Requerimientos funcionales.....	2
1.5 Requerimientos no funcionales.....	2
1.6 Otros – Requisitos de transición.....	3
Bibliografía .....	3
CAPITULO II .....	4
2. BPMN – Diagramas de proceso de negocio.....	4
2.1 Diagrama BPMN (As Is):.....	4
2.2 Diagrama BPMN (To Be):.....	5
Diseño de Base de Datos.....	6
2.3 Diagrama de casos de uso del sistema.....	6
2.4 Refinamiento de los RF y RNF.....	7
2.5 Modelo lógico de la Base de Datos .....	8
2.6 Modelo Físico de la Base de Datos.....	1
2.7 Diccionario de datos (presentar un avance).....	5
2.8 Scripts del proceso de construcción de la BBDD. (presentar un avance).....	7
2.9 Copias de respaldo (pendiente) .....	10
CAPITULO III .....	15
3.1 Presentación de la arquitectura de su Proyecto.....	36
3.2 La estructura del Proyecto (Debe apreciarse la arquitectura en su aplicación) .....	37
3.3 Detalles de las herramientas de desarrollo y sistemas de gestión de bases de datos. ....	43
3.4 Porcentaje de avance del Proyecto. ....	45
3.5 Carga al repositorio GitHub .....	45
3.6 Bibliografía .....	45

# Avance de Proyecto Final 3

## CAPITULO I

### Presentación del caso propuesto.

#### **IMPLEMENTACIÓN DE UN SITIO WEB (ECOMMERS PARA LA VENTA DE PRODUCTOS TECNOLÓGICOS Y/O DE CÓMPUTO).**

El presente proyecto tiene como propósito desarrollar una aplicación web de comercio electrónico para la empresa UTP Technology, con el fin de facilitar la venta de equipos tecnológicos en línea, como laptops, smartphones, periféricos y accesorios. Esta plataforma busca cubrir la necesidad actual de la empresa de contar con un sistema moderno, eficiente y seguro que le permita competir en el entorno digital, mejorar su gestión interna y brindar una experiencia satisfactoria a sus clientes.

El sistema será construido utilizando el framework Spring Boot, una herramienta poderosa basada en Java que permite el desarrollo ágil de aplicaciones web robustas. Además, se implementarán tecnologías complementarias como Spring Security, para garantizar la autenticación y autorización de usuarios, y Spring Data JPA, para una gestión eficiente de la base de datos. Estas herramientas permitirán construir una aplicación escalable, modular y segura, alineada con las mejores prácticas del desarrollo web actual.

La plataforma ofrecerá funcionalidades clave como registro e inicio de sesión de usuarios con diferentes roles, visualización de productos organizados por categorías, incorporación de artículos a un carrito de compras y la posibilidad de generar pedidos en línea. Por otro lado, los administradores contarán con un panel privado que les permitirá gestionar el inventario, editar productos, controlar categorías, monitorear pedidos y administrar usuarios.

Este proyecto se desarrolla en el marco del curso Desarrollo Web Integrado, y representa la integración práctica de conocimientos adquiridos en diversas áreas del desarrollo de software, como backend, bases de datos, seguridad y arquitectura MVC. Al finalizar su implementación, la solución permitirá a UTP Technology automatizar su proceso de ventas, ampliar su alcance comercial y mejorar la relación con sus clientes, posicionándose como una empresa moderna y competitiva en el rubro tecnológico.

## Modelo lean Canvas del caso propuesto.

### ■ Socios claves

Proveedores de productos tecnológicos (mayoristas)  
Plataformas de pago (MercadoPago, PayPal)  
Empresas de logística (Olva Courier, Shalom)  
Servicios de hosting y dominio web  
Alianzas con influencers o técnicos de tecnología

### ■ Actividades claves

Gestión del inventario  
Actualización de la tienda online  
Marketing digital (SEO, redes sociales, e-mailing)  
Atención al cliente  
Logística y entrega

### ■ Propuesta de valor

Acceso rápido y confiable a productos tecnológicos actuales  
Compra segura y asesoría técnica personalizada  
Buen precio y promociones exclusivas para estudiantes  
Envío rápido y garantizado

### ■ Relaciones con el cliente

Soporte técnico en línea  
Atención personalizada vía chat y WhatsApp  
Programas de fidelización y descuentos  
Seguimiento postventa y encuestas de satisfacción

### ■ Segmentos de clientes

Estudiantes universitarios  
Profesionales de tecnología  
Usuarios interesados en actualizar sus equipos  
Empresas pequeñas que requieren tecnología accesible

**■ Recursos clave**

- Plataforma e-commerce (sitio web y app)
- Stock de productos tecnológicos
- Equipo humano (atención, técnicos, marketing)
- Base de datos de clientes
- Acuerdos con proveedores confiables

**■ Canales**

- Página web oficial (UTPTechnology.pe)
- Redes sociales (Instagram, Facebook, TikTok)
- Marketplace (MercadoLibre)
- Publicidad en Google Ads y redes
- Email marketing

**■ Estructura de costes**

- Compra de productos tecnológicos
- Costos de logística y distribución
- Publicidad y marketing digital
- Mantenimiento de la web y soporte técnico
- Comisiones por métodos de pago

**■ Flujos de ingresos**

- Venta directa de productos
- Servicios de instalación y mantenimiento
- Paquetes tecnológicos personalizados
- Accesorios y gadgets complementarios
- Comisiones por afiliados

## Diagrama de Canvas.

# Modelo de negocio Canvas de UTPTECHNOLOGY



### Socios claves

¿Quiénes son nuestros socios claves?

- Proveedores de productos tecnológicos (mayoristas)
- Plataformas de pago (MercadoPago, PayPal)
- Empresas de logística (Olva Courier, Shalom)
- Servicios de hosting y dominio web
- Alianzas con influencers o técnicos de tecnología

### Actividades claves

¿Qué actividades requieren nuestra propuesta de valor?

- Gestión del inventario.
- Actualización de la tienda online.
- Marketing digital (SEO, redes sociales, e-mailing)
- Atención al cliente
- Logística y entrega

### Recursos claves

¿Qué recursos claves son requeridos por nuestra propuesta de valor?

- Plataforma e-commerce (sitio web).
- Stock de productos tecnológicos
- Equipo humano (atención, técnicos, marketing).
- Base de datos de clientes
- Acuerdos con proveedores confiables

### Propuesta de valor

¿Qué valor entregamos a nuestros clientes?

- Acceso rápido y confiable a productos tecnológicos actuales
- Compra segura y asesoría técnica personalizada
- Buen precio y promociones exclusivas para estudiantes
- Envío rápido y garantizado

### Relaciones con el cliente

¿Qué tipo de relación espera cada uno de nuestros segmentos de clientes?

- Soporte técnico en línea
- Atención personalizada vía chat y WhatsApp
- Programas de fidelización y descuentos
- Seguimiento postventa y encuestas de satisfacción

### Canales

¿A través de qué canales quieren los segmentos de clientes que lleguemos a ellos?

- Página web oficial (UTPTechnology.pe)
- Redes sociales (Instagram, Facebook, TikTok)
- Marketplace (MercadoLibre, OLX)
- Publicidad en Google Ads y redes
- Email marketing

### Segmentos de clientes

¿Para quién estamos creando valor?

- Estudiantes universitarios
- Profesionales de tecnología
- Usuarios interesados en actualizar sus equipos
- Empresas pequeñas que requieren tecnología accesible

### Estructura de costes

¿Cuáles son los costes más importantes inherentes a nuestro modelo de negocio?

- Compra de productos tecnológicos
- Costos de logística y distribución
- Publicidad y marketing digital
- Mantenimiento de la web y soporte técnico
- Comisiones por métodos de pago

### Flujos de ingresos

¿Por qué valor están realmente dispuestos a pagar nuestros clientes?

- Venta online de productos
- Servicios de instalación y mantenimiento
- Paquetes tecnológicos personalizados
- Accesorios y gadgets complementarios
- Comisiones por afiliados

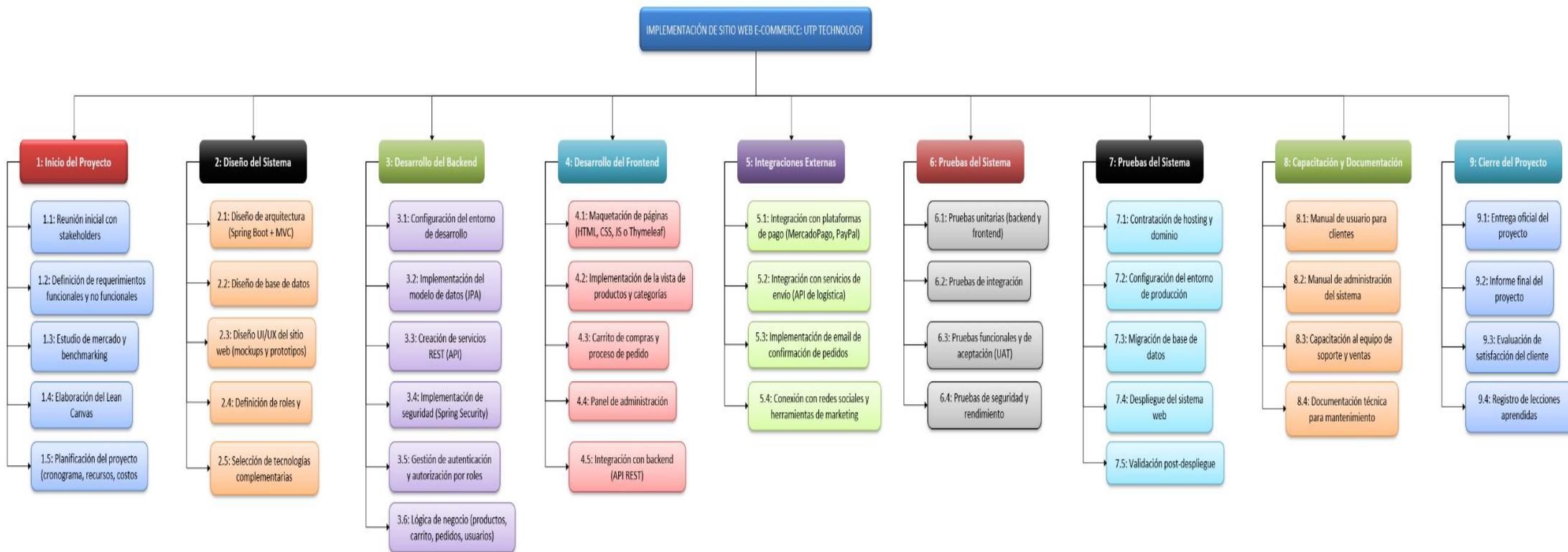
## 1. El proyecto del caso propuesto.

### 1.1 Project charter del proyecto.

Project Charter							
Título	UTP TECHNOLOGY		Jefe de Proyecto	MIGUEL VELASQUEZ YSUIZA			
Fecha inicio	14/04/2025	Fecha fin	25/07/2025	Sponsor	ECOMMERS		
Necesidad del negocio							
<p>El propósito de este proyecto es optimizar y expandir el modelo de negocio de UTPTECHNOLOGY para mejorar su presencia online, aumentar los canales de venta y optimizar las relaciones con los clientes. La necesidad del negocio se enfoca en incrementar las ventas de productos tecnológicos y mejorar los sistemas de logística y entrega, al mismo tiempo que se mantiene un alto nivel de servicio al cliente y satisfacción. UTPTECHNOLOGY busca ofrecer soluciones tecnológicas accesibles y actualizadas a estudiantes, profesionales y pequeñas empresas a través de una plataforma de comercio electrónico robusta..</p>							
Alcance		Entregables					
<p>El proyecto incluye la creación de una tienda en línea para productos tecnológicos, con una interfaz fácil de usar y diseñada para facilitar las compras. La plataforma se integrará con métodos de pago seguros como MercadoPago y PayPal.</p>		<p>Entregar una tienda en línea completamente operativa con integración de métodos de pago como MercadoPago y PayPal. La plataforma incluirá un sistema de gestión de inventarios que permitirá monitorear y actualizar el stock de productos en tiempo real, asegurando que la información de disponibilidad esté siempre actualizada.</p>					
<p>Se desarrollará una estrategia de marketing digital que abarque SEO (optimización para motores de búsqueda) para mejorar el posicionamiento en Google y atraer tráfico orgánico. Se crearán campañas en redes sociales (Instagram, Facebook, TikTok)</p>		<p>Desarrollar y ejecutar campañas de SEO para mejorar el posicionamiento de la tienda en motores de búsqueda. Se implementarán campañas en redes sociales (Instagram, Facebook, TikTok) con contenido atractivo y anuncios pagos para aumentar la visibilidad de la marca. Además, se llevará a cabo una estrategia de email marketing para captar y retener clientes</p>					
<p>Para asegurar que los productos lleguen de manera oportuna, se establecerán acuerdos con empresas logísticas como Olva Courier y Shalom. Esto garantizará que las entregas sean rápidas y confiables. Además, se implementará un sistema de seguimiento de pedidos para que los clientes puedan monitorear el estado de sus compras en tiempo real, asegurando así una experiencia de compra más transparente y satisfactoria</p>		<p>Firmar acuerdos con empresas de logística como Olva Courier y Shalom para asegurar la entrega puntual y segura de los productos. Además, se implementará un sistema de seguimiento de pedidos en tiempo real para que los clientes puedan consultar el estado de su pedido desde la compra hasta la entrega final..</p>					
<p>El proyecto incluirá un sistema de atención al cliente mediante chat en vivo en la plataforma de comercio electrónico y soporte a través de WhatsApp, brindando respuestas rápidas y asistencia personalizada. Además, se implementará un programa de fidelización, ofreciendo descuentos y promociones a clientes recurrentes para fomentar la lealtad a la marca.</p>		<p>Implementar un sistema de atención al cliente que incluya soporte a través de chat en vivo y WhatsApp. El equipo de atención estará disponible para responder dudas sobre productos, procesos de compra, y soluciones postventa.</p>					
Riesgos y problemas		Suposiciones y dependencias					
<p>El proyecto enfrenta riesgos técnicos como fallas en la integración con plataformas de pago (MercadoPago, PayPal), errores en el sistema de inventarios o en el seguimiento de pedidos, así como problemas de compatibilidad en la interfaz. También existen riesgos de seguridad relacionados con ataques informáticos o filtración de datos, lo que afectaría la confianza del cliente. En el aspecto logístico, pueden presentarse retrasos o pérdidas en los envíos debido a fallas por parte de Olva Courier o Shalom</p>		<p>Se asume que los usuarios contarán con acceso a Internet y dispositivos compatibles para realizar sus compras, y que las plataformas de pago como MercadoPago y PayPal estarán operativas sin restricciones. También se da por hecho que las empresas logísticas cumplirán con los tiempos de entrega establecidos, que los proveedores actualizarán el stock en tiempo real, y que el equipo contará con los recursos necesarios para implementar el proyecto. Entre las principales dependencias se encuentran la correcta integración con las pasarelas de pago, la disponibilidad y cumplimiento de las empresas de reparto (Olva Courier y Shalom)</p>					
Costos							
20,000.00 Veinte mil con 00/100							
Cronograma							

Hitos y Actividades		Responsable	Fecha Prog.	Fecha Real
PLANIFICACIÓN Y DISEÑO DEL SISTEMA		ROBERTO	14/04/2025	7/05/2025
IMPLEMENTACIÓN DE FUNCIONALIDADES BÁSICAS		DANIEL	14/05/2025	12/06/2025
GENERACIÓN DE REPORTES Y CÁLCULOS ESTADÍSTICOS		MIGUEL	19/06/2025	14/07/2025
PRUEBAS Y PRESENTACIÓN FINAL		MANUEL	17/07/2025	25/07/2025
Equipo		Comité de Aprobación		
Jefe de Proyecto	MIGUEL VELASQUEZ YSUIZA	Sponsor	ECOMMERS	
Supervisor	MEJÍA COLLAZOS, Roberto Agustín	Gerente	JUAN DIAZ	
Contador	PECHO SANTOS, Manuel Angel	Jefe	PEDRO MARTINEZ	

## 1.2 Desglose de la estructura de tareas WBS



## 1.3 Diagrama de Gantt

### IMPLEMENTACIÓN DE SITIO WEB E-COMMERCE

UTP TECHNOLOGY  
MANUEL ANGEL PECHO SANTOS

Fecha de inicio del proyecto: 14/04/2025

Incremento de desplazamiento: 0

Descripción del hito	Categoría	Asignado a	Progreso	Inicio	Número de días
----------------------	-----------	------------	----------	--------	----------------

#### ANÁLISIS Y PLANIFICACIÓN

Análisis de requisitos	Objetivo	Miguel	20%	14/04/2025	5
Definición de Tecnología y Plataforma	Hito	Roberto	15%	19/04/2025	3
Ánalisis de la Competencia	Riesgo bajo	Daniel	10%	22/04/2025	4
Elaboración de Plan de Trabajo	Según lo previsto	Manuel	25%	26/04/2025	7
Aprobación del Plan de Trabajo por parte del Comité	Hito	Manuel	30%	3/05/2025	4

#### DESARROLLO DEL SITIO WEB

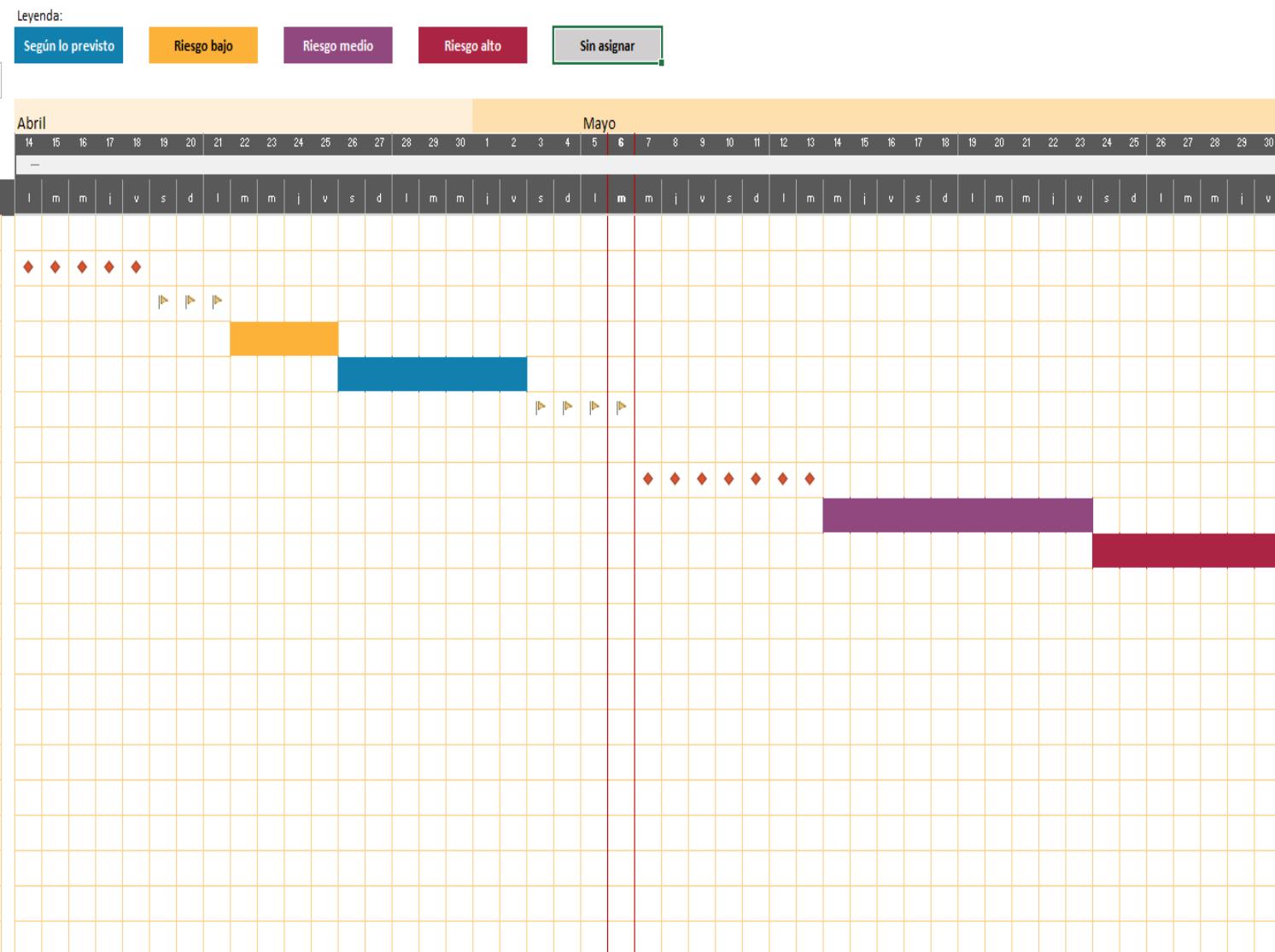
Diseño de la Estructura del Sitio	Objetivo	Daniel	20%	7/05/2025	7
Desarrollo del Frontend	Riesgo medio	Manuel	30%	14/05/2025	10
Desarrollo del Backend con Spring Boot e integración de base de Datos MySQL	Riesgo alto	Roberto	30%	24/05/2025	15
Pruebas de Funcionalidad y Usabilidad	Según lo previsto	Miguel	10%	8/06/2025	6
Pruebas de Seguridad	Riesgo medio	Manuel	10%	14/06/2025	6

#### OPTIMIZACIÓN Y PRUEBAS DE

Optimización de Velocidad y Seguridad	Riesgo bajo	Miguel	25%	20/06/2025	5
Pruebas de Usabilidad	Según lo previsto	Roberto	20%	25/06/2025	5
Pruebas de Carga y Estrés	Riesgo medio	Daniel	20%	30/06/2025	7
Ajustes según Retroalimentación de Usuarios	Objetivo	Manuel	15%	7/07/2025	7
Pruebas Finales y Aprobación	Hito	Manuel	20%	14/07/2025	5

#### DESPLIEGUE Y MANTENIMIENTO

Despliegue en el Servidor de Producción	Hito	Daniel	15%	19/07/2025	3
Monitoreo Post-Lanzamiento	Riesgo bajo	Manuel	25%	22/07/2025	4



## Requerimientos

### 1.4 Requerimientos funcionales

ID	Requerimiento Funcional	Descripción
RF01	Registro de usuarios	Permite a los usuarios registrarse ingresando sus datos personales y rol.
RF02	Inicio de sesión	Permite a los usuarios iniciar sesión con credenciales válidas.
RF03	Gestión de roles	Define permisos de acceso para clientes y administradores.
RF04	Visualización de productos	Muestra productos clasificados en categorías.
RF05	Búsqueda y filtrado	Permite buscar productos por nombre, categoría o rango de precio.
RF06	Carrito de compras	Permite agregar, quitar y modificar productos en el carrito.
RF07	Generación de pedidos	Permite confirmar un pedido y generar la orden de compra.
RF08	Métodos de pago	Integra pasarelas como MercadoPago o PayPal para procesar pagos.
RF09	Gestión de productos (Admin)	Permite al administrador agregar, editar y eliminar productos.
RF10	Gestión de pedidos (Admin)	Permite al administrador gestionar y monitorear pedidos.
RF11	Gestión de usuarios (Admin)	Permite al administrador ver y modificar la información de los usuarios.
RF12	Historial de compras	Permite a los usuarios ver sus compras anteriores.
RF13	Soporte y contacto	Permite comunicarse con soporte por chat, WhatsApp o correo
RF14	Notificaciones	Envía correos automáticos sobre el estado de pedidos, registro, etc.
RF15	Evaluación postventa	Permite a los usuarios calificar productos y dejar comentarios.

### 1.5 Requerimientos no funcionales

ID	Requerimiento No Funcional	Descripción
RNF01	Seguridad	Utiliza Spring Security para autenticar y proteger el acceso.
RNF02	Escalabilidad	Permite añadir nuevos productos y usuarios sin afectar el rendimiento.
RNF03	Disponibilidad	El sistema estará disponible al menos el 99% del tiempo.
RNF04	Rendimiento	Las páginas deben cargar en un tiempo inferior a 3 segundos.
RNF05	Usabilidad	Interfaz intuitiva y adaptada a dispositivos móviles.

RNF06	Compatibilidad	Funciona en navegadores modernos como Chrome, Firefox, Edge.
RNF07	Mantenibilidad	Código estructurado y documentado siguiendo el patrón MVC.
RNF08	Backup	Se realizan respaldos diarios de la base de datos.
RNF09	Protección de datos	La información sensible se almacena de forma encriptada.

### 1.6 Otros – Requisitos de transición.

ID	Requerimiento Funcional	Descripción
RT01	Capacitación al personal	Se capacitará al equipo de UTP Technology en el uso del sistema.
RT02	Migración de datos	Se trasladarán datos previos del inventario al nuevo sistema.
RT03	Pruebas del sistema	Se realizarán pruebas funcionales y de seguridad antes del despliegue.
RT04	Plan piloto	Se implementará una versión de prueba con un grupo limitado de usuarios.
RT05	Implementación progresiva	El sistema se activará por etapas para asegurar estabilidad.
RT06	Documentación técnica y de usuario	Se entregará manual técnico y guía de usuario final.

## Bibliografía

Pivotal Software, Inc. (n.d.). *Spring Boot Reference Documentation*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Pivotal Software, Inc. (n.d.). *Spring Security Reference*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-security/reference/index.html>

Pivotal Software, Inc. (n.d.). *Spring Data JPA Reference Documentation*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

OWASP Foundation. (2021). *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. Recuperado de <https://owasp.org/www-project-top-ten/>

Thymeleaf. (n.d.). *Thymeleaf Documentation*. Recuperado el 14 de mayo de 2025, de <https://www.thymeleaf.org/documentation.html>

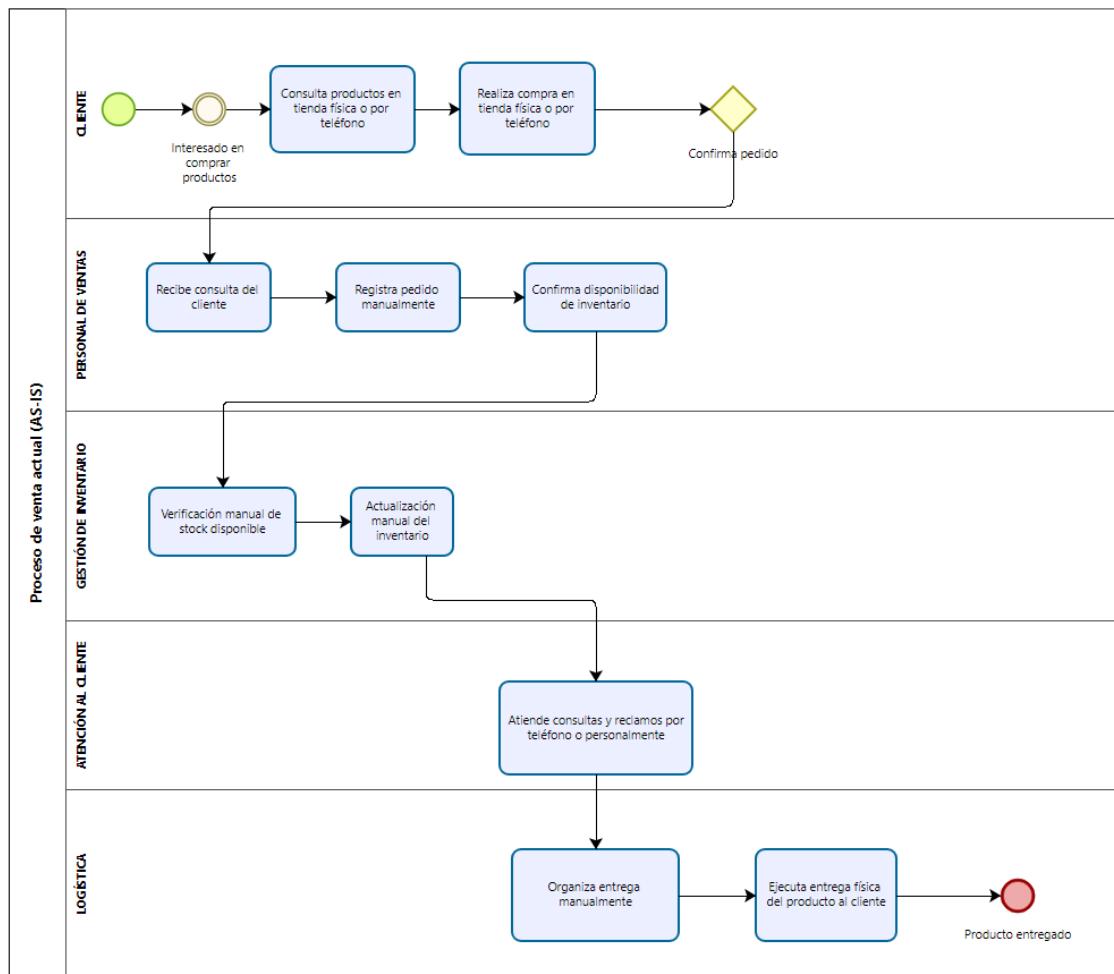
## CAPITULO II

### 2. BPMN – Diagramas de proceso de negocio.

Descripción del proceso o procesos de acuerdo con el formato del caso:

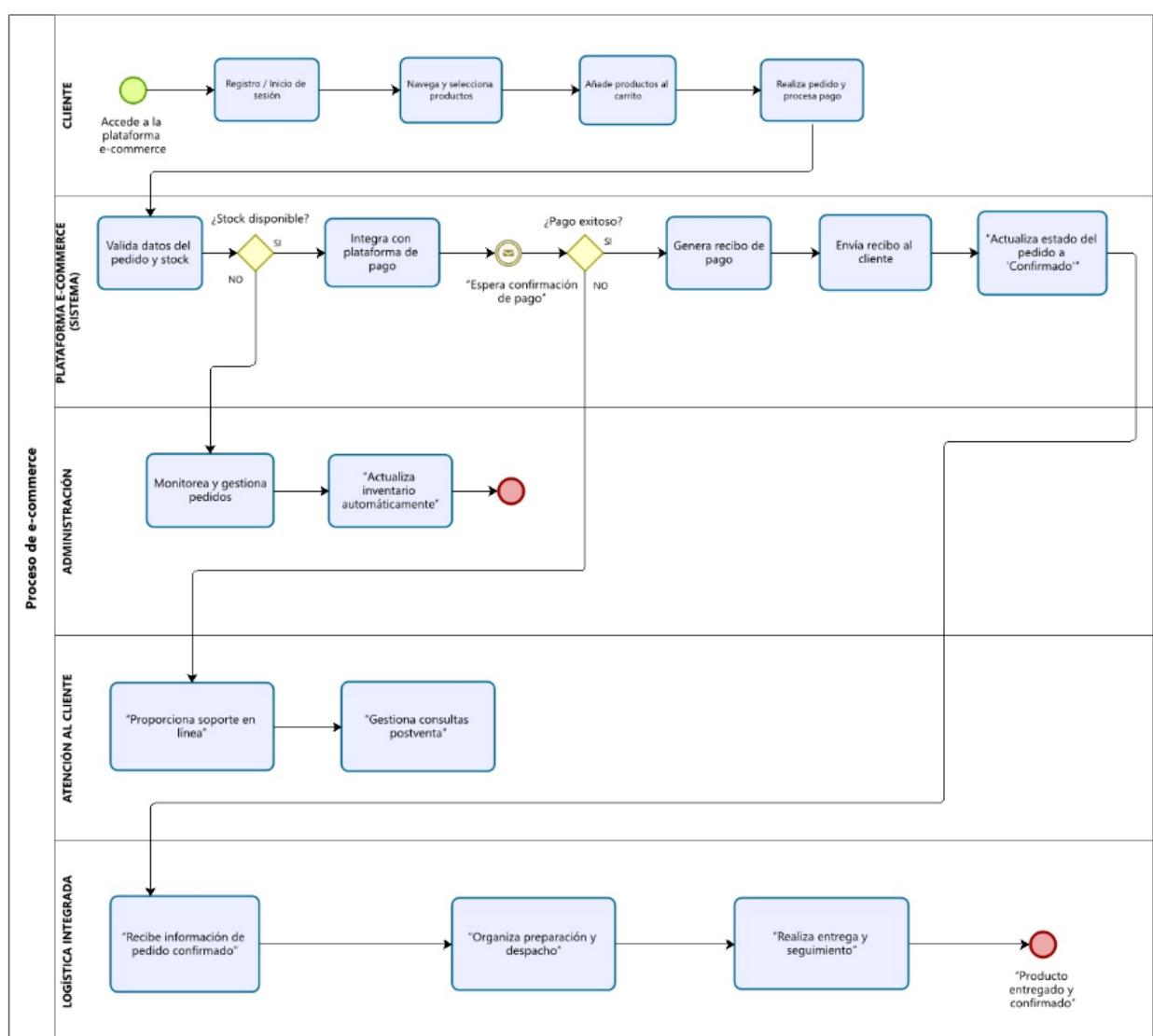
#### 2.1 Diagrama BPMN (As Is):

Muestra los procesos y/o actividades tales como se vienen realizando (As Is): Sin la intervención del equipo informático.



## 2.2 Diagrama BPMN (To Be):

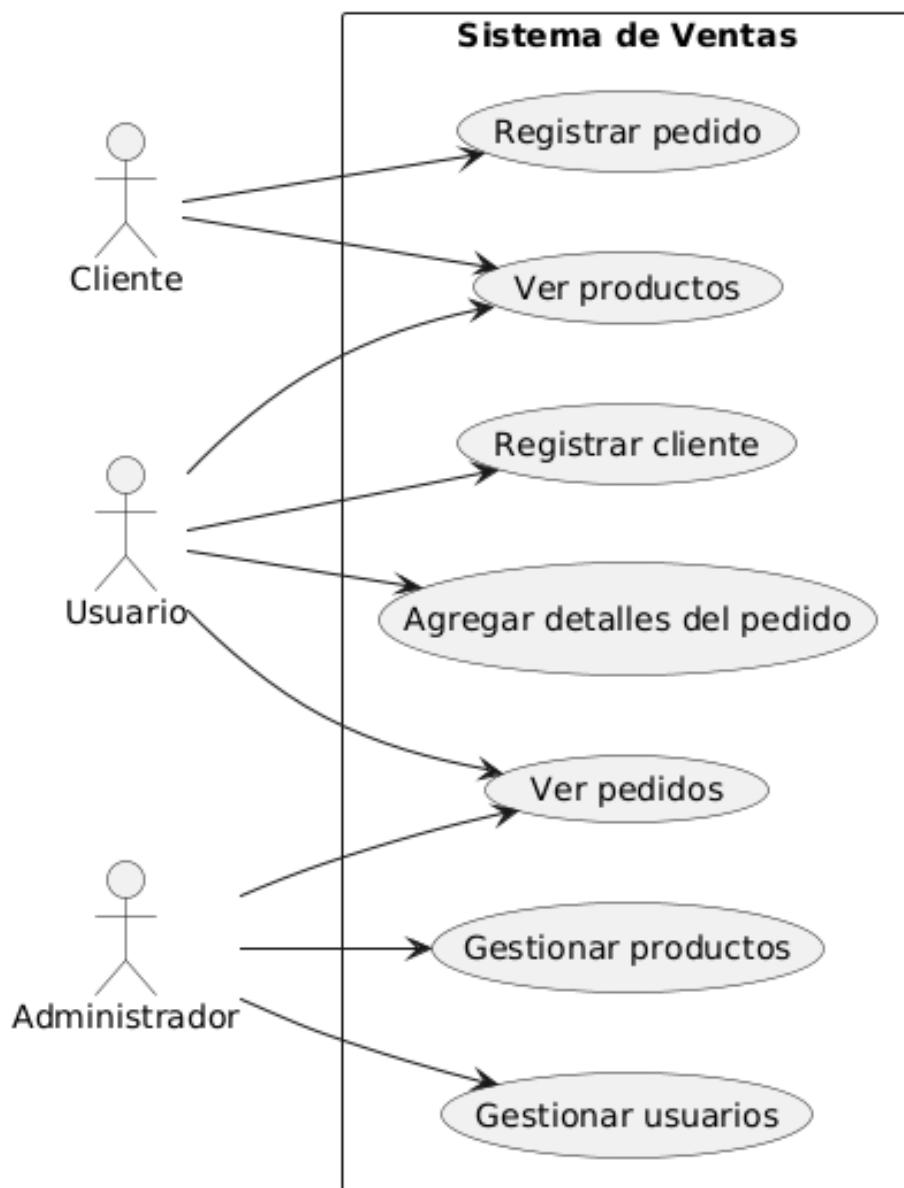
Muestra los procesos y/o actividades con la solución integrada (To be): Con la intervención del equipo informático.



## Diseño de Base de Datos

### 2.3 Diagrama de casos de uso del sistema.

Identificación de procesos (CU: Casos de Uso) del sistema, se pone en contexto el sistema y sus límites. (Un escenario ó caso de uso puede considerarse como el equivalente a un formulario de interacción de su aplicación).



## 2.4 Refinamiento de los RF y RNF

En base a los procesos del sistema (CU)identificados en el diagrama de casos de uso.

- Requerimientos Funcionales (RF)

Los requerimientos funcionales describen lo que el sistema **debe hacer**:

ID	Requerimiento Funcional
RF01	El sistema debe permitir a los usuarios registrar productos con nombre, imagen, precio y stock.
RF02	El sistema debe permitir registrar clientes con datos personales.
RF03	El sistema debe permitir a un usuario registrar un pedido asociado a un cliente.
RF04	El sistema debe permitir ingresar múltiples productos en un mismo pedido.
RF05	El sistema debe permitir consultar el historial de pedidos por cliente.
RF06	El sistema debe permitir actualizar el estado de un pedido (pendiente, en proceso, completado).
RF07	El sistema debe permitir la autenticación de usuarios mediante correo y clave.
RF08	El sistema debe validar el stock disponible antes de confirmar un pedido.

- Requerimientos No Funcionales (RNF)

Los requerimientos no funcionales describen cómo debe comportarse el sistema o las condiciones de su funcionamiento:

ID	Requerimiento No Funcional
RNF01	El sistema debe estar disponible las 24 horas del día.
RNF02	La interfaz del sistema debe ser intuitiva y fácil de usar.
RNF03	El tiempo de respuesta al registrar o consultar pedidos no debe superar los 3 segundos.
RNF04	Los datos sensibles (como la clave del usuario) deben almacenarse en formato cifrado.
RNF05	El sistema debe permitir escalar a una mayor cantidad de usuarios y productos.
RNF06	El sistema debe cumplir con normas básicas de seguridad web.
RNF07	Las imágenes de los productos deben cargarse en menos de 2 segundos con buena conexión.

## 2.5 Modelo lógico de la Base de Datos

Descripción de las entidades identificadas: nombre, descripción, atributos, llave primaria, llave foránea (ver página 21 de la guía ejemplo).

### Modelo Conceptual (entidades)

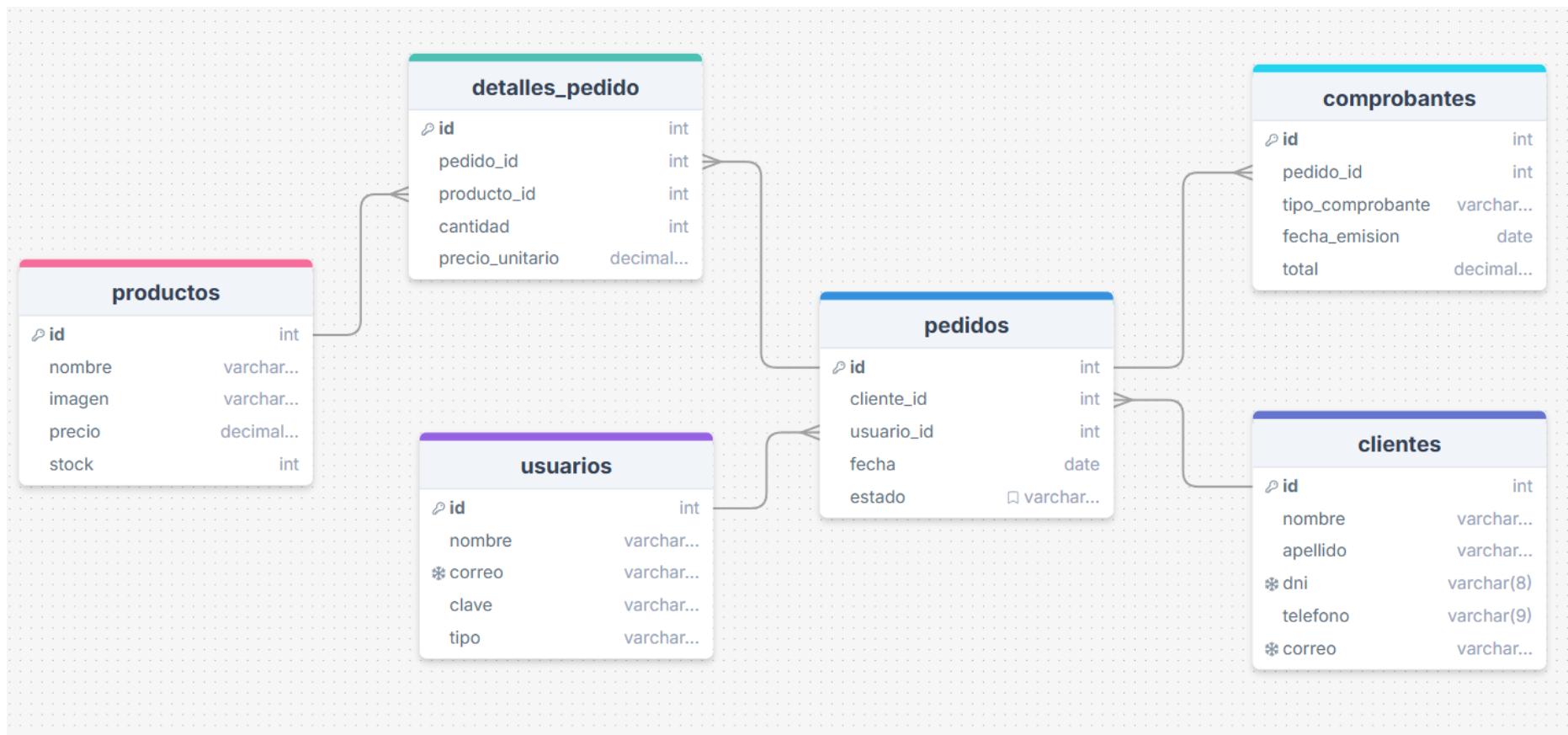
Nombre	Descripción	Atributos
Productos	Representa los artículos tecnológicos disponibles en la tienda.	id, nombre, imagen, precio, stock
Clientes	Representa a las personas que realizan pedidos en la tienda.	id, nombre, apellido, dni, teléfono, correo
Usuarios	Representa a los administradores o personal que gestionan el sistema.	id, nombre, correo, clave, tipo
Pedidos	Representa una orden realizada por un cliente y gestionada por un usuario.	id, cliente_id, usuario_id, fecha, estado
Detalles_Pedido	Representa cada producto incluido en un pedido.	id, pedido_id, producto_id, cantidad, precio_unitario
Comprobantes	Documento emitido tras una compra, como boleta o factura.	id pedido_id tipo_comprobante fecha_emision total

### Modelo Conceptual (relaciones)

Nombre de la Relación	Descripción
Clientes - Pedidos	Un cliente puede realizar uno o varios pedidos. Cada pedido pertenece a un único cliente.
Usuarios - Pedidos	Un usuario (administrador) registra uno o varios pedidos. Cada pedido es gestionado por un solo usuario.
Pedidos – Detalles_Pedido	Un pedido puede tener uno o varios detalles. Cada detalle pertenece a un solo pedido.

Nombre de la Relación	Descripción
Productos – Detalles_Pedido	Un producto puede estar en uno o varios detalles de pedido. Cada detalle se refiere a un solo producto.
Pedidos – Comprobantes	Cada pedido puede tener un comprobante asociado. Cada comprobante pertenece a un solo pedido.

## Esquema del modelo.



## Diseño lógico.

Relación	Atributos
productos	id (PK), nombre, imagen, precio, stock
clientes	id (PK), nombre, apellido, dni, telefono, correo
usuarios	id (PK), nombre, correo, clave, tipo
pedidos	id (PK), cliente_id (FK), usuario_id (FK), fecha, estado
detalles_pedido	id (PK), pedido_id (FK), producto_id (FK), cantidad, precio_unitario
comprobante	id (PK), pedido_id (FK → pedidos.id), tipo, monto_total, fecha_emision

## 2.6 Modelo Físico de la Base de Datos.

Indicar el SGBD, el fabricante, la versión y distribución.

Elemento	Detalle
SGBD	MySQL
Fabricante	Oracle Corporation
Versión	8.0.33
Distribución	MySQL Community Edition

Sustente porqué eligió el SGBD.

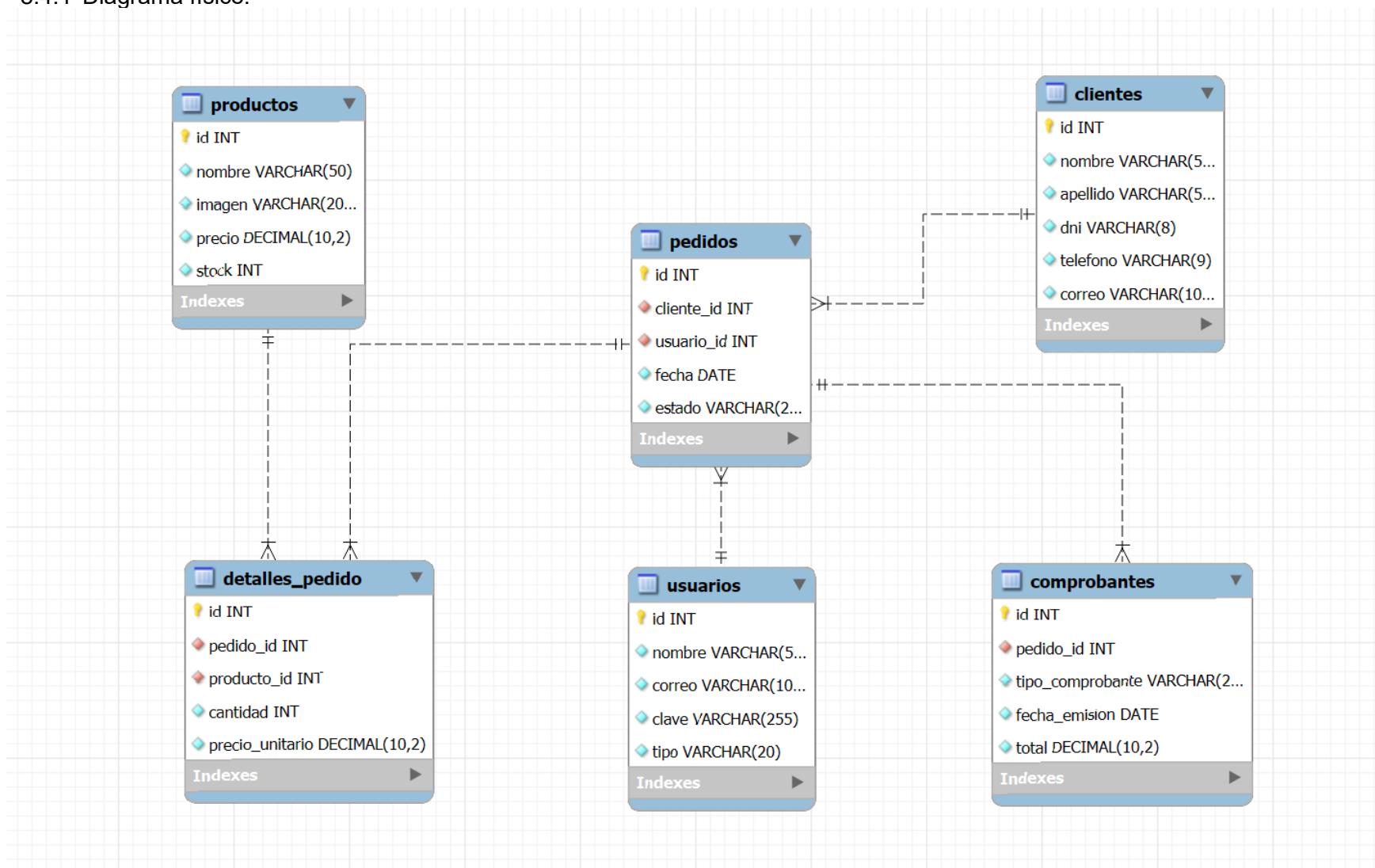
**Motivos por los que se eligió MySQL:**

- Es un sistema de gestión de base de datos relacional (RDBMS) de código abierto ampliamente utilizado.
- Compatible con múltiples plataformas y lenguajes de programación.
- Soporta integridad referencial, transacciones y procedimientos almacenados.
- Alto rendimiento para aplicaciones web.
- Amplia comunidad y documentación.
- Interfaz gráfica disponible con MySQL Workbench o herramientas como phpMyAdmin.

Detalle de cada tabla (ver página 24 de la guía ejemplo)

Tabla	Descripción
usuarios	Almacena los datos de los trabajadores o encargados del sistema.
clientes	Contiene la información de los compradores o usuarios finales.
productos	Almacena los productos tecnológicos disponibles.
pedidos	Representa cada compra realizada por un cliente.
detalles_pedido	Relaciona productos con pedidos, detallando cantidades y precios.
comprobante	Representa el documento generado como comprobante por un pedido realizado

## 8.4.1 Diagrama físico.



8.4.2 En caso de utilizar archivos planos (txt, csv, unl, xlsx) muestre el proceso de normalización.

☞ Supuesto archivo plano original: `ventas.csv`

```
Producto,Precio,Stock,Cliente,CorreoCliente,Usuario,CorreoUsuario
Audífonos,30.00,10,Juan Pérez,juan@gmail.com,Máximo Moreno,max@utp.edu
Laptop,3200.00,10,José Tello,jose@gmail.com,Máximo Moreno,max@utp.edu
```

Este archivo representa datos de ventas, pero está **desnormalizado** (todos los datos en una sola tabla).

⚙ Proceso de Normalización en MySQL

◆ **Paso 1: Crear base de datos**

```
CREATE DATABASE IF NOT EXISTS bd_utpTechnology;
USE bd_utpTechnology;
```

◊ **Paso 2: Crear tablas normalizadas**

```
-- Crear tabla productos
CREATE TABLE productos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    imagen VARCHAR(200) NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    stock INT NOT NULL
);

-- Crear tabla usuarios
CREATE TABLE usuarios (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    correo VARCHAR(100) UNIQUE NOT NULL,
    clave VARCHAR(255) NOT NULL,
    tipo VARCHAR(20) NOT NULL
);

-- Crear tabla clientes
CREATE TABLE clientes (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
```

```
dni VARCHAR(8) UNIQUE NOT NULL,  
telefono VARCHAR(9) NOT NULL,  
correo VARCHAR(100) UNIQUE NOT NULL  
);  
  
-- Crear tabla pedidos  
CREATE TABLE pedidos (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    cliente_id INT NOT NULL,  
    usuario_id INT NOT NULL,  
    fecha DATE NOT NULL,  
    estado VARCHAR(20) NOT NULL DEFAULT 'Pendiente',  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id),  
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id)  
);  
  
-- Crear tabla detalles de pedido  
CREATE TABLE detalles_pedido (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    pedido_id INT NOT NULL,  
    producto_id INT NOT NULL,  
    cantidad INT NOT NULL,  
    precio_unitario DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (pedido_id) REFERENCES pedidos(id),  
    FOREIGN KEY (producto_id) REFERENCES productos(id)  
);  
-- Crear tabla comprobantes  
CREATE TABLE comprobantes (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    pedido_id INT NOT NULL,  
    tipo_comprobante VARCHAR(20) NOT NULL,  
    fecha_emision DATE NOT NULL,  
    total DECIMAL(10,2) NOT NULL,  
    FOREIGN KEY (pedido_id) REFERENCES pedidos(id)  
);
```

**◆ Paso 3: Insertar los datos normalizados desde el CSV**

Simulación de inserción con los datos del ejemplo:

```
-- Insertar 10 productos
INSERT INTO productos (nombre, imagen, precio, stock) VALUES
('Audífonos Bluetooth',
'https://images.pexels.com/photos/16688488/pexels-photo-16688488.jpeg', 35.00, 10),
('Disco Duro 1TB',
'https://images.pexels.com/photos/20076003/pexels-photo-20076003.jpeg', 220.00, 8),
('Cámara Deportiva',
'https://images.pexels.com/photos/11002709/pexels-photo-11002709.jpeg', 150.00, 5),
('Auriculares Gaming',
'https://images.pexels.com/photos/3921803/pexels-photo-3921803.jpeg', 60.00, 12),
('Smartwatch',
'https://images.pexels.com/photos/1034653/pexels-photo-1034653.jpeg', 180.00, 9),
('Tarjeta de Video GTX 1660',
'https://images.pexels.com/photos/4581903/pexels-photo-4581903.jpeg', 350.00, 4),
('Celular Android',
'https://images.pexels.com/photos/10343717/pexels-photo-10343717.jpeg', 780.00, 6),
('Laptop i5',
'https://images.pexels.com/photos/577210/pexels-photo-577210.jpeg', 3200.00, 7),
('Teclado Mecánico',
'https://images.pexels.com/photos/267389/pexels-photo-267389.jpeg', 90.00, 10),
('Mouse Inalámbrico',
'https://images.pexels.com/photos/392018/pexels-photo-392018.jpeg', 45.00, 15);

-- Insertar 5 usuarios
INSERT INTO usuarios (nombre, correo, clave, tipo) VALUES
```

```
('Administrador General', 'admin@tienda.com', 'admin123',
'admin'),
('Vendedor 1', 'vendedor1@tienda.com', 'clave1',
'vendedor'),
('Vendedor 2', 'vendedor2@tienda.com', 'clave2',
'vendedor'),
('Soporte Técnico', 'soporte@tienda.com', 'soporte123',
'soporte'),
('Encargado Almacén', 'almacen@tienda.com', 'almacen123',
'almacen');

-- Insertar 5 clientes
INSERT INTO clientes (nombre, apellido, dni, telefono,
correo) VALUES
('Juan', 'Pérez', '78945612', '987654321',
'juan.perez@example.com'),
('María', 'Rodríguez', '12345678', '999888777',
'maria.rodriguez@example.com'),
('Luis', 'Gonzales', '87451236', '987321654',
'luis.gonzales@example.com'),
('Ana', 'Fernández', '45612378', '912345678',
'ana.fernandez@example.com'),
('Carlos', 'Ramírez', '74125896', '900800700',
'carlos.ramirez@example.com');

-- Insertar 5 pedidos
INSERT INTO pedidos (cliente_id, usuario_id, fecha, estado)
VALUES
(1, 1, '2024-05-01', 'Pendiente'),
(2, 2, '2024-05-02', 'En proceso'),
(3, 3, '2024-05-03', 'Completado'),
(4, 4, '2024-05-04', 'Pendiente'),
(5, 5, '2024-05-05', 'Cancelado');

-- Insertar 5 detalles de pedido
INSERT INTO detalles_pedido (pedido_id, producto_id,
cantidad, precio_unitario) VALUES
(1, 1, 2, 35.00),
```

```
(1, 2, 1, 220.00),
(2, 3, 1, 150.00),
(3, 4, 1, 60.00),
(4, 5, 2, 180.00);
```

-- Insertar 5 comprobantes

```
INSERT INTO comprobantes (pedido_id, tipo_comprobante,
fecha_emision, total) VALUES
(1, 'Boleta', '2024-05-01', 290.00),
(2, 'Factura', '2024-05-02', 150.00),
(3, 'Boleta', '2024-05-03', 60.00),
(4, 'Factura', '2024-05-04', 360.00),
(5, 'Boleta', '2024-05-05', 0.00);
```

## 2.7 Diccionario de datos (presentar un avance)

Tabla o Entidad	Campo	Tipo de Dato	Descripción
productos	id	INT, PK, AUTO_INCREMENT	Identificador único del producto.
	nombre	VARCHAR(50)	Nombre del producto.
	imagen	VARCHAR(200)	URL de la imagen del producto.
	precio	DECIMAL(10,2)	Precio del producto.
	stock	INT	Cantidad disponible en inventario.
usuarios	id	INT, PK, AUTO_INCREMENT	Identificador único del usuario.
	nombre	VARCHAR(50)	Nombre completo del usuario.
	correo	VARCHAR(100), UNIQUE	Correo electrónico del usuario.
	clave	VARCHAR(255)	Contraseña del usuario.
	tipo	VARCHAR(20)	Rol del usuario (admin, vendedor, soporte, etc.).

Tabla o Entidad	Campo	Tipo de Dato	Descripción
clientes	id	INT, PK, AUTO_INCREMENT	Identificador único del cliente.
	nombre	VARCHAR(50)	Nombre del cliente.
	apellido	VARCHAR(50)	Apellido del cliente.
	dni	VARCHAR(8), UNIQUE	Documento Nacional de Identidad.
	telefono	VARCHAR(9)	Número de teléfono del cliente.
	correo	VARCHAR(100), UNIQUE	Correo electrónico del cliente.
pedidos	id	INT, PK, AUTO_INCREMENT	Identificador único del pedido.
	cliente_id	INT, FK	Identificador del cliente que hizo el pedido.
	usuario_id	INT, FK	Identificador del usuario que procesó el pedido.
	fecha	DATE	Fecha de realización del pedido.
	estado	VARCHAR(20)	Estado actual del pedido (Pendiente, Completado, Cancelado, etc.).
detalles_pedido	id	INT, PK, AUTO_INCREMENT	Identificador único del detalle.
	pedido_id	INT, FK	Identificador del pedido asociado.
	producto_id	INT, FK	Identificador del producto incluido.
	cantidad	INT	Cantidad solicitada del producto.
	precio_unitario	DECIMAL(10,2)	Precio del producto al momento del pedido.
comprobante	id	INT, PK, AUTO_INCREMENT	Identificador único del comprobante.
	pedido_id	INT, FK	Pedido asociado al comprobante.
	tipo	VARCHAR(30)	Tipo de comprobante (boleta, factura, etc.).
	monto_total	DECIMAL(10,2)	Monto total del comprobante.

Tabla o Entidad	Campo	Tipo de Dato	Descripción
	fecha_emision	DATE	Fecha en que se emite el comprobante.
	numero_comprobante	VARCHAR(20)	Número único del comprobante.

## 2.8 Scripts del proceso de construcción de la BBDD. (presentar un avance)

```
3 -- Crear la base de datos y usarla
4 CREATE DATABASE IF NOT EXISTS utp_bd;
5 USE utp_bd;
6
7 -- Crear tabla rol
8 CREATE TABLE rol (
9     id_rol INT PRIMARY KEY AUTO_INCREMENT,
10    nombre_rol VARCHAR(50) NOT NULL,
11    descripcion VARCHAR(100)
12 );
13
14 -- Crear tabla productos
15 CREATE TABLE productos (
16    id INT PRIMARY KEY AUTO_INCREMENT,
17    nombre VARCHAR(50) NOT NULL,
18    imagen VARCHAR(200) NOT NULL,
19    precio DECIMAL(10,2) NOT NULL,
20    stock INT NOT NULL
21 );
22
23 -- Crear tabla usuarios (modificada para usar id_rol)
24 CREATE TABLE usuarios (
25    id INT PRIMARY KEY AUTO_INCREMENT,
26    nombre VARCHAR(50) NOT NULL,
27    correo VARCHAR(100) UNIQUE NOT NULL,
28    clave VARCHAR(255) NOT NULL,
29    id_rol INT NOT NULL,
30    FOREIGN KEY (id_rol) REFERENCES rol(id_rol)
31 );
32
33 -- Crear tabla clientes
34 CREATE TABLE clientes (
35    id INT PRIMARY KEY AUTO_INCREMENT,
36    nombre VARCHAR(50) NOT NULL,
37    apellido VARCHAR(50) NOT NULL,
38    dni VARCHAR(8) UNIQUE NOT NULL,
39    telefono VARCHAR(9) NOT NULL,
40    correo VARCHAR(100) UNIQUE NOT NULL,
41    usuario_id INT NOT NULL,
```

```
42 FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
43 );
44
45 -- ALTER TABLE clientes add column usuario_id int;
46 -- ALTER TABLE clientes ADD FOREIGN KEY (usuario_id) REFERENCES
47 usuarios(id);
48
49 -- Crear tabla pedidos
50 CREATE TABLE pedidos (
51   id INT PRIMARY KEY AUTO_INCREMENT,
52   cliente_id INT NOT NULL,
53   usuario_id INT NOT NULL,
54   fecha DATE NOT NULL,
55   estado VARCHAR(20) NOT NULL DEFAULT 'Pendiente',
56   FOREIGN KEY (cliente_id) REFERENCES clientes(id),
57   FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
58 );
59
60 -- Crear tabla detalles de pedido
61 CREATE TABLE detalles_pedido (
62   id INT PRIMARY KEY AUTO_INCREMENT,
63   pedido_id INT NOT NULL,
64   producto_id INT NOT NULL,
65   cantidad INT NOT NULL,
66   precio_unitario DECIMAL(10,2) NOT NULL,
67   FOREIGN KEY (pedido_id) REFERENCES pedidos(id),
68   FOREIGN KEY (producto_id) REFERENCES productos(id)
69 );
70
71 -- Crear tabla comprobantes
72 CREATE TABLE comprobantes (
73   id INT PRIMARY KEY AUTO_INCREMENT,
74   pedido_id INT NOT NULL,
75   tipo_comprobante VARCHAR(20) NOT NULL,
76   fecha_emision DATE NOT NULL,
77   total DECIMAL(10,2) NOT NULL,
78   FOREIGN KEY (pedido_id) REFERENCES pedidos(id)
79 );
80
81 -- Insertar roles
82 INSERT INTO rol (nombre_rol, descripcion) VALUES
83 ('Administrador', 'Acceso total al sistema'),
84 ('Vendedor', 'Gestión de ventas'),
85 ('Cliente', 'Clientes, acceso a pedidos, carrito, etc');
86
87 -- Insertar 10 productos
88 INSERT INTO productos (nombre, imagen, precio, stock) VALUES
```

```
88 ('Audífonos Bluetooth',
     'https://images.pexels.com/photos/16688488/pexels-photo-16688488.jpeg',
     35.00, 10),
89 ('Disco Duro 1TB', 'https://images.pexels.com/photos/20076003/pexels-
     photo-20076003.jpeg', 220.00, 8),
90 ('Cámara Deportiva', 'https://images.pexels.com/photos/11002709/pexels-
     photo-11002709.jpeg', 150.00, 5),
91 ('Auriculares Gaming',
     'https://images.pexels.com/photos/3921803/pexels-photo-3921803.jpeg',
     60.00, 12),
92 ('Smartwatch', 'https://images.pexels.com/photos/1034653/pexels-photo-
     1034653.jpeg', 180.00, 9),
93 ('Tarjeta de Video GTX 1660',
     'https://images.pexels.com/photos/4581903/pexels-photo-4581903.jpeg',
     350.00, 4),
94 ('Celular Android', 'https://images.pexels.com/photos/10343717/pexels-
     photo-10343717.jpeg', 780.00, 6),
95 ('Laptop i5', 'https://images.pexels.com/photos/577210/pexels-photo-
     577210.jpeg', 3200.00, 7),
96 ('Teclado Mecánico', 'https://images.pexels.com/photos/267389/pexels-
     photo-267389.jpeg', 90.00, 10),
97 ('Mouse Inalámbrico', 'https://images.pexels.com/photos/392018/pexels-
     photo-392018.jpeg', 45.00, 15);
98
99 -- Insertar 5 usuarios (ahora con id_rol)
100INSERT INTO usuarios (nombre, correo, clave, id_rol) VALUES
101('Administrador General', 'admin@tienda.com', 'admin123', 1), --
    Administrador
102('Vendedor 1', 'vendedor1@tienda.com', 'clave1', 2), -- Vendedor
103('Vendedor 2', 'vendedor2@tienda.com', 'clave2', 2), -- Vendedor
104('Cliente', 'soporte@tienda.com', 'soporte123', 3), -- Soporte
105('Cliente2', 'almacen@tienda.com', 'almacen123', 3); -- Almacén
106
107-- Insertar 5 clientes
108INSERT INTO clientes (nombre, apellido, dni, telefono, correo,
    usuario_id) VALUES
109('Juan', 'Pérez', '78945612', '987654321', 'juan.perez@example.com',
    4),
110('María', 'Rodríguez', '12345678', '999888777',
    'maria.rodriguez@example.com', 5);
111
112-- Insertar 5 pedidos
113INSERT INTO pedidos (cliente_id, usuario_id, fecha, estado) VALUES
114(1, 1, '2024-05-01', 'Pendiente'),
115(2, 2, '2024-05-02', 'En proceso'),
116(1, 3, '2024-05-03', 'Completado'),
117(2, 1, '2024-05-04', 'Pendiente'),
118(1, 2, '2024-05-05', 'Cancelado');
```

```
119
120-- Insertar 5 detalles de pedido
121INSERT INTO detalles_pedido (pedido_id, producto_id, cantidad,
    precio_unitario) VALUES
122(1, 1, 2, 35.00),
123(1, 2, 1, 220.00),
124(2, 3, 1, 150.00),
125(3, 4, 1, 60.00),
126(4, 5, 2, 180.00);
127
128-- Insertar 5 comprobantes
129INSERT INTO comprobantes (pedido_id, tipo_comprobante, fecha_emision,
    total) VALUES
130(11, 'Boleta', '2024-05-01', 290.00),
131(12, 'Factura', '2024-05-02', 150.00),
132(13, 'Boleta', '2024-05-03', 60.00),
133(14, 'Factura', '2024-05-04', 360.00),
134(15, 'Boleta', '2024-05-05', 0.00);
135
```

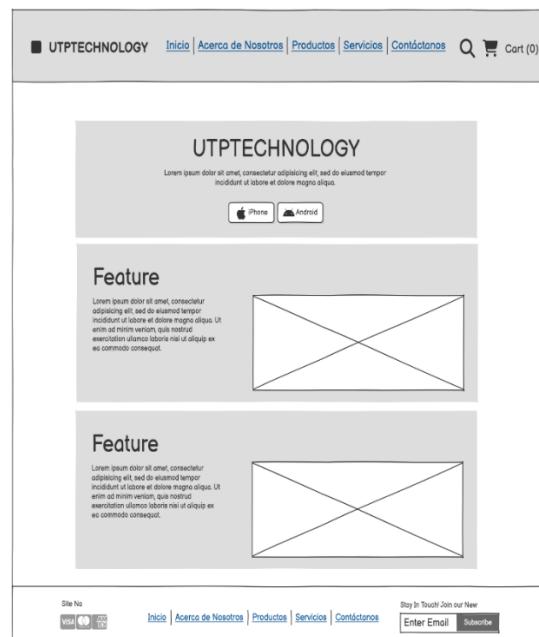
## 2.9 Copias de respaldo (pendiente)

## UI / UX:

Prototipos de bajo nivel y/o maquetado del producto a construir

Wireframe

Index.html



Acerca.html



Productos.html

■ UTPTECHNOLOGY [Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#) Cart (0)

## Nuestros Productos

Our North American Field Guides provide tips for identifying birds in all of the regions of the United States and Canada. Download any of our guides for free now!

### Feature Message

Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00

Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00      Product Title  
Attributes \$00.00

Site No Stay In Touch! Join our New [Enter Email](#) [Subscribe](#)

Login.html

■ UTPTECHNOLOGY [Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#) Cart (0)

## ■ UTPTECHNOLOGY

### Check your inbox

Enter the verification code we just sent to  
[tomcook@example.com](mailto:tomcook@example.com).

[Continue](#) [Resend email](#)

Site No Stay In Touch! Join our New [Enter Email](#) [Subscribe](#)

## Servicios.html

■ UTPTECHNOLOGY [Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#)   Cart (0)

## Servicios

Full Name  
Role at C

### Social Proof Heading

Site No

[Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#) Stay In Touch! Join our New [Enter Email](#) [Subscribe](#)

## Contactos.html

■ UTPTECHNOLOGY [Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#)   Cart (0)

## Contáctanos

Full Name  
Role at C

### Contact

ADDR  
456 Main  
Street  
Anywhere,

CONT  
[singers@avengers.com](mailto:singers@avengers.com)  
555 123-6543

[Learn Mo](#)

[Email a friend](#)

To: (email address)

From: (email address)

Note:

[Send](#)

Site No

[Inicio](#) | [Acerca de Nosotros](#) | [Productos](#) | [Servicios](#) | [Contáctanos](#) Stay In Touch! Join our New [Enter Email](#) [Subscribe](#)

Muestre escenarios de interacción.

Escenario N°	Nombre del Escenario	Actor Principal	Descripción del Escenario
1	Registro de cliente	Cliente	El usuario se registra en el sitio web completando un formulario con sus datos personales.
2	Inicio de sesión	Cliente / Usuario	El usuario ingresa sus credenciales para acceder al sistema.
3	Navegación de productos	Cliente	El cliente navega por las categorías y visualiza productos en el catálogo.
4	Agregar productos al carrito	Cliente	El cliente selecciona productos para comprarlos y los añade al carrito.
5	Realizar pedido	Cliente	El cliente confirma el carrito, selecciona la dirección y método de pago, y realiza el pedido.
6	Gestión de productos	Usuario (Admin)	El administrador agrega, edita o elimina productos desde el panel de administración.
7	Gestión de clientes	Usuario (Admin)	El usuario administrador puede ver y gestionar los clientes registrados.
8	Consulta de pedidos	Cliente / Usuario	El cliente puede consultar sus pedidos realizados, y el administrador puede revisarlos.
9	Envío de confirmación por correo	Sistema	Luego de completar un pedido, el sistema envía una confirmación al cliente.
10	Actualización del stock	Sistema / Usuario	Después de una compra, el sistema descuenta automáticamente el stock del producto.

Indicar la herramienta que utiliza.

Herramienta	Uso principal
Draw.io	Para diagramar los casos de uso y flujos de interacción. Gratis y en línea.
Lucidchart	Ideal para modelar procesos, diagramas de flujo y escenarios colaborativos.
Balsamiq	Para crear wireframes y prototipos rápidos de las interfaces del sistema web.

## CAPITULO III

### 3.1 Presentación de la arquitectura de su Proyecto.

El proyecto **utp-technology** está desarrollado siguiendo la arquitectura **MVC (Modelo - Vista - Controlador)** combinada con principios de **arquitectura en capas** y seguridad basada en **JWT (JSON Web Token)**. Esta organización permite separar responsabilidades, facilitando el mantenimiento y la escalabilidad del sistema.

#### Descripción de la arquitectura:

- **Capa de Configuración:** Define configuraciones globales de la aplicación, como la documentación Swagger y la seguridad (paquetes config y security).
- **Capa de Controladores (controller):** Contiene las clases que manejan las solicitudes HTTP, reciben peticiones del cliente, procesan datos de entrada y devuelven respuestas apropiadas. Cada controlador se encarga de un dominio específico (usuarios, productos, pedidos, clientes, autenticación).
- **Capa de DTO y Respuestas (dto y response):** Define los objetos de transferencia de datos (DTOs) para estructurar y validar la información intercambiada entre cliente y servidor, así como las clases para formatear las respuestas API.
- **Capa de Modelo (model):** Contiene las entidades de dominio que representan las tablas de la base de datos, mapeadas mediante JPA/Hibernate.
- **Capa de Repositorios (repository):** Proporciona interfaces para acceder y manipular los datos persistentes, siguiendo el patrón de repositorio de Spring Data JPA.
- **Capa de Servicios (services y impl):** Implementa la lógica de negocio de la aplicación, separando las reglas y procesos del resto de componentes. Las interfaces definen los contratos y las clases impl implementan dicha lógica.
- **Capa de Seguridad (security):** Gestiona la autenticación y autorización de usuarios usando JWT. Incluye filtros, servicios para validar tokens y clases para integrar la seguridad con Spring Security.
- **Pruebas (test):** Incluye pruebas unitarias básicas para validar la correcta configuración y ejecución de la aplicación.

### 3.2 La estructura del Proyecto (Debe apreciarse la arquitectura en su aplicación)

#### Estructura backend

```
utp-technology-backend/
├── utp-technology-backend/
│   ├── .editorconfig
│   ├── .gitattributes
│   ├── .gitignore
│   ├── mvnw
│   ├── mvnw.cmd
│   ├── pom.xml
│   ├── .devcontainer/
│   │   └── devcontainer.json
│   ├── .mvn/
│   │   └── wrapper/
│   │       └── maven-wrapper.properties
│   └── src/
│       ├── main/
│       │   ├── java/
│       │   │   └── com/
│       │   │       └── utp/
│       │   │           └── technology/
│       │   │               ├── UtpTechnologyApplication.java
│       │   │               └── config/
│       │   │                   └── SwaggerConfig.java
│       │   └── http/
│       │       ├── controller/
│       │       │   ├── AuthController.java
│       │       │   ├── ClienteController.java
│       │       │   ├── PedidoController.java
│       │       │   ├── ProductController.java
│       │       └── UsuarioController.java
│       └── dto/
│           ├── auth/
│           │   ├── LoginRequestDTO.java
│           │   ├── LoginResponseDto.java
│           │   └── SignInRequestDTO.java
│           ├── cliente/
│           │   └── ListClienteDto.java
│           └── pedido/
│               ├── ListDetallePedidoDto.java
│               ├── ListPedidoAdmin.java
│               ├── ListPedidoCliente.java
│               └── StorePedidoDetalleDto.java
```

```
 | | | | └── StorePedidoDto.java  
 | | | └── product/  
 | | | | └── ListProductoDto.java  
 | | | | └── StoreProductDTO.java  
 | | | └── user/  
 | | | | └── UserListDto.java  
 | | | | └── UserLoginDto.java  
 | | | └── response/  
 | | | | └── ApiPageResponse.java  
 | | | └── ApiResponse.java  
 | | └── model/  
 | | | └── Cliente.java  
 | | | └── Comprobante.java  
 | | | └── DetallePedido.java  
 | | | └── Pedido.java  
 | | | └── Producto.java  
 | | | └── Rol.java  
 | | | └── Usuario.java  
 | | └── repository/  
 | | | └── ClienteRepository.java  
 | | | └── PedidoDetalleRepository.java  
 | | | └── PedidoRepository.java  
 | | | └── ProductoRepository.java  
 | | | └── RolRepository.java  
 | | | └── UsuarioRepository.java  
 | | └── security/  
 | | | └── AuthService.java  
 | | | └── CustomUserDetailsService.java  
 | | | └── JWTAuthenticationFilter.java  
 | | | └── JWTService.java  
 | | | └── JwtUsuario.java  
 | | | └── SecurityConfig.java  
 | | └── services/  
 | | | └── ClienteService.java  
 | | | └── MercadoPagoService.java  
 | | | └── PedidoService.java  
 | | | └── ProductoService.java  
 | | | └── UsuarioService.java  
 | | └── impl/  
 | | | └── ClienteServiceImpl.java  
 | | | └── MercadoPagoServiceImpl.java  
 | | | └── PedidoServiceImpl.java  
 | | | └── ProductoServiceImpl.java  
 | | | └── UsuarioServiceImpl.java  
 | └── resources/  
 | └── application.properties
```

```
|   |   |   └─ META-INF/  
|   |   |       └─ additional-spring-configuration-metadata.json  
|   |   └─ test/  
|   |       └─ java/  
|   |           └─ com/  
|   |               └─ utp/  
|   |                   └─ technology/  
|   |                           └─ UtpTechnologyApplicationTests.java
```

**Explicación de cada parte****Archivos de configuración raíz:**

- .editorconfig, .gitattributes, .gitignore → Configuración para mantener coherencia en el formato de archivos, atributos de Git y exclusiones.
- mvnw, mvnw.cmd, .mvn/ → Wrapper de Maven para garantizar la versión y ejecución del proyecto sin instalación manual.
- pom.xml → Archivo principal de Maven, define dependencias, plugins y configuración del proyecto.

**devcontainer/**

- devcontainer.json → Configura el contenedor de desarrollo para usar un entorno reproducible.

**src/main/java/com/utp/jwt/**

- JwtApplication.java → Clase principal que arranca la aplicación Spring Boot.
- config/SwaggerConfig.java → Configura Swagger para documentar y probar la API.
- http/controller/ → Controladores REST para manejar las solicitudes:
  - AuthController: Autenticación y registro.
  - ClienteController: CRUD de clientes.
  - PedidoController: CRUD de pedidos.
  - ProductController: CRUD de productos.
  - UsuarioController: CRUD de usuarios.
- http/dto/ → Objetos de Transferencia de Datos (DTOs) organizados por contexto (auth, cliente, pedido, product, user). Se usan para solicitudes/respuestas.
- http/response/ → Clases para respuestas genéricas de la API.

- model/ → Entidades del dominio: Cliente, Pedido, Producto, Usuario, Rol, etc.
- repository/ → Interfaces que extienden JpaRepository para interactuar con la base de datos.
- security/ → Implementa autenticación y autorización JWT: filtros, servicios, configuración de seguridad.
- services/ → Interfaces de servicios y sus implementaciones (impl/): lógica de negocio de clientes, pedidos, productos, usuarios y pago (MercadoPago).

**src/main/resources/**

- application.properties → Configuración de propiedades (puerto, base de datos, JWT, etc.).
- META-INF/ → Metadatos adicionales de configuración.

**src/test/**

- JwtApplicationTests.java → Prueba básica de carga de contexto de la aplicación.

**Estructura Frontend**

```
— utp-angular-frontend-master
  — .editorconfig
  — .gitignore
  — .postcssrc.json
  — angular.json
  — package-lock.json
  — package.json
  — README.md
  — tsconfig.app.json
  — tsconfig.json
  — tsconfig.spec.json
  — .devcontainer
    — devcontainer.json
  — .vscode
    — extensions.json
    — launch.json
    — tasks.json
  — public
    — favicon.ico
    — images
      — logo.jpg
      — logo.png
  — src
```

```
custom-theme.scss
index.html
main.server.ts
main.ts
server.ts
styles.css
theme.scss
app
  app.config.server.ts
  app.config.ts
  app.css
  app.html
  app.routes.server.ts
  app.routes.ts
  app.spec.ts
  app.ts
  guards
    admin-guard.spec.ts
    admin-guard.ts
  interceptors
    token-expired-interceptor.spec.ts
    token-expired-interceptor.ts
    token-interceptor.spec.ts
    token-interceptor.ts
  intl
    spanish-paginator-intl.ts
models
  api.ts
  auth.model.ts
  carrito-compras.ts
  cliente.ts
  pedido.ts
  producto.ts
  respuesta.ts
  usuario.ts
services
  auth.service.spec.ts
  auth.service.ts
  carrito-compras.service.spec.ts
  carrito-compras.service.ts
  cliente.service.spec.ts
  cliente.service.ts
  pedido.service.spec.ts
  pedido.service.ts
  producto.service.spec.ts
  producto.service.ts
  usuario.service.spec.ts
  usuario.service.ts
views
auth
  login
```

```
login.component.css
login.component.html
login.component.spec.ts
login.component.ts
carrito-compras
  carrito-compras.component.css
  carrito-compras.component.html
  carrito-compras.component.spec.ts
  carrito-compras.component.ts
cliente
  cliente.component.css
  cliente.component.html
  cliente.component.spec.ts
  cliente.component.ts
pedido
  pedido.component.css
  pedido.component.html
  pedido.component.spec.ts
  pedido.component.ts
producto
  producto.component.css
  producto.component.html
  producto.component.spec.ts
  producto.component.ts
usuario
  usuario.component.css
  usuario.component.html
  usuario.component.spec.ts
  usuario.component.ts
```

### **Resumen de la estructura**

#### **Archivos de configuración general:**

Contiene archivos como angular.json, package.json, y configuraciones de compilación y dependencias de Angular.

#### **Carpeta .vscode y .devcontainer:**

Definen configuraciones del entorno de desarrollo, extensiones recomendadas y contenedor para desarrollo remoto.

#### **public:**

Recursos públicos como íconos y logotipos que se usan en la interfaz.

#### **src:**

Contiene toda la lógica de la aplicación:

- **Archivos base:** main.ts, index.html, styles.css.

- **app:** Núcleo de la aplicación Angular.
  - **guards:** Guardias de ruta para controlar acceso (por ejemplo, admin).
  - **interceptors:** Interceptores de HTTP para gestionar tokens y expiración.
  - **intl:** Configuraciones de internacionalización.
  - **models:** Modelos de datos para usuarios, pedidos, productos, etc.
  - **services:** Servicios para manejar la lógica de negocio y llamadas al backend JWT.
  - **views:** Componentes de interfaz para login, carrito, cliente, pedido, producto y usuario.

### 3.3 Detalles de las herramientas de desarrollo y sistemas de gestión de bases de datos.

Para el desarrollo del proyecto se utilizaron herramientas modernas y ampliamente adoptadas en la industria del desarrollo de software, garantizando eficiencia, seguridad y escalabilidad. A continuación, se detallan:

#### ***Herramientas de desarrollo***

- **Frontend:**
  - **Framework:** Angular 20  
Se utilizó Angular para construir la interfaz de usuario dinámica y responsive, facilitando la gestión de rutas, componentes y servicios HTTP.
  - **Lenguaje:** TypeScript 5.8  
Proporciona tipado estático y características avanzadas de programación orientada a objetos, mejorando la mantenibilidad.
  - **Entorno de desarrollo:** Visual Studio Code 1.101.0  
Editor ligero y flexible, con extensiones para desarrollo en Angular y control de versiones Git.
  - **Gestor de dependencias:** npm (Node Package Manager) 22  
Para instalar y administrar librerías y paquetes necesarios para Angular.
- **Backend:**
  - **Framework:** Spring Boot 3.4.6  
Permite crear microservicios robustos y seguros de forma rápida, con integración de seguridad JWT.

- **Lenguaje:** Java 24

Lenguaje de propósito general, orientado a objetos y con gran soporte para aplicaciones empresariales.
- **Entorno de desarrollo:** IntelliJ IDEA o Eclipse

IDEs potentes para el desarrollo en Java y configuración de proyectos Maven.
- **Gestor de dependencias:** Maven

Facilita la gestión de librerías y la construcción del proyecto backend.
- **Control de versiones:**
  - **Git**

Se utilizó Git para el control de versiones, permitiendo el trabajo colaborativo y el seguimiento de cambios.
  - **GitHub (opcional):**

Para almacenar y compartir el repositorio de forma remota.

### ***Sistemas de gestión de bases de datos (SGBD)***

- **Base de datos:** MySQL

Se implementó MySQL como sistema de gestión de bases de datos relacional, permitiendo almacenar la información estructurada del sistema (usuarios, productos, clientes, pedidos).  
Se integra con JPA/Hibernate en el backend para el mapeo objeto-relacional (ORM), facilitando la persistencia de datos.

### ***Otros recursos complementarios***

- **Swagger:**

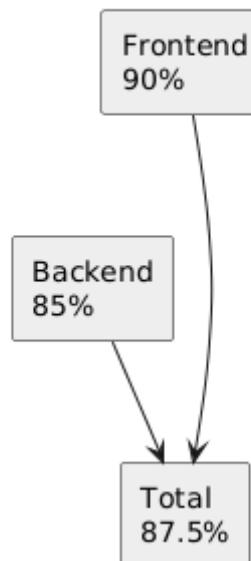
Se configuró Swagger para documentar y probar la API REST del backend de forma interactiva.
- **Postman (opcional):**

Para realizar pruebas manuales de los endpoints de la API.
- **Docker (opcional):**

Puede utilizarse para desplegar contenedores del backend, base de datos y frontend de forma aislada.

### 3.4 Porcentaje de avance del Proyecto.

**Avance del Proyecto**



### 3.5 Carga al repositorio GitHub

**Enlace al repositorio de GitHub:**

[https://github.com/robermejia/G03\\_Informe\\_03\\_Integrador\\_I](https://github.com/robermejia/G03_Informe_03_Integrador_I)

### 3.6 Bibliografía

Pivotal Software, Inc. (n.d.). *Spring Boot Reference Documentation*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

Pivotal Software, Inc. (n.d.). *Spring Security Reference*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-security/reference/index.html>

Pivotal Software, Inc. (n.d.). *Spring Data JPA Reference Documentation*. Recuperado el 14 de mayo de 2025, de <https://docs.spring.io/spring-data/jpa/docs/current/reference/html/>

OWASP Foundation. (2021). *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. Recuperado de <https://owasp.org/www-project-top-ten/>

Thymeleaf. (n.d.). *Thymeleaf Documentation*. Recuperado el 14 de mayo de 2025, de <https://www.thymeleaf.org/documentation.html>