



**Universidad  
Tecnológica  
del Perú**

**Facultad de Ingeniería**

**Carrera Profesional de Ingeniería de Sistemas e Informática**

**Asignatura**

**TALLER DE PROGRAMACIÓN WEB**

**Título del proyecto**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA  
GESTIÓN DE VENTAS Y PROMOCIÓN DE PRODUCTOS DE LA  
PANADERÍA AMASARTE, ICA 2025**

**Estudiantes:**

1. Pérez Llamas, Pedro Luis
2. Quispe Huayhua, Juan Carlos
3. Quispe Ticllasuca, Edy Jutner

**Ica – Perú**

**2025**

## ÍNDICE

INTRODUCCIÓN .....	3
CAPÍTULO 1 – ASPECTOS GENERALES .....	3
1.1. Descripción de la Organización.....	3
1.2. Misión y Visión.....	3
1.3. Arquitectura Tecnológica Actual.....	4
CAPÍTULO 2 – MARCO TEÓRICO .....	4
2.1. tecnologías Web .....	4
2.2 Arquitecturas Web.....	6
CAPÍTULO 3 – GESTIÓN DEL PROYECTO.....	8
3.1. Objetivos .....	8
3.2. Descripción del problema: .....	8
3.3. Alcance del proyecto .....	9
3.4. Metodología de desarrollo .....	10
3.5. Planificación y cronograma .....	10
CAPÍTULO 4 – IMPLEMENTACIÓN DE ESTRATEGIA .....	12
4.1. Lista de Requerimientos.....	12
4.2. Prototipo Preliminar .....	13
4.3 Prototipo Funcional .....	17
4.4. Desarrollo del sitio web.....	20
4.5. Desarrollo de la BD .....	22
4.6. Pruebas y Validaciones para la Página Web de una Panadería .....	23
4.7. Optimización y mejoras.....	25
CAPITULO 5 - CIERRE Y DOCUMENTACION .....	26
5.1. Versión Final de producto .....	26
5.2 Manuales del usuario.....	31

## **INTRODUCCIÓN**

Este proyecto busca crear un sistema web que ayude a la Panadería Dulce, ubicada en Ica, a organizar mejor sus ventas y a dar a conocer sus productos artesanales. La idea es que, con esta herramienta tecnológica, puedan registrar las ventas de forma más sencilla y mostrar sus productos de manera atractiva para que más clientes conozcan y compren lo que ofrecen. Así, la panadería podrá crecer y ofrecer un mejor servicio sin complicarse con registros manuales.

La plataforma también permitirá que los clientes hagan pedidos en línea, mejorando la atención y ampliando su alcance. Así, la panadería podrá crecer, ofrecer un mejor servicio y mantener viva su tradición artesanal, adaptándose a las nuevas tecnologías para potenciar su negocio

## **CAPÍTULO 1 – ASPECTOS GENERALES**

### **1.1. Descripción de la Organización**

La Panadería Dulce es un negocio familiar que se dedica a hacer y vender panes y otros productos artesanales, hechos con recetas tradicionales y mucho cariño. Tienen un local pequeño en Ica donde atienden a sus clientes todos los días con productos frescos y de calidad. Actualmente, manejan sus ventas con registros manuales y buscan modernizarse para mejorar la forma en que organizan su trabajo y llegan a más personas.

Actualmente, manejan sus ventas con registros manuales, usando hojas de papel y cálculos en la cabeza, lo que puede causar errores y dificulta la organización. Con el crecimiento del negocio, necesitan una forma más ordenada y moderna de administrar sus ventas, productos y atención a los clientes. La implementación de un sistema digital les ayudará a facilitar estas tareas, mejorar su servicio y seguir creciendo en el mercado local

### **1.2. Misión y Visión**

#### **Misión:**

Ofrecer a nuestros clientes productos de panadería artesanal frescos y deliciosos, manteniendo la calidad y el buen trato, mientras usamos herramientas tecnológicas que nos ayuden a ser más eficientes y a crecer.

#### **Visión:**

Ser una panadería reconocida en Ica por la calidad de nuestros productos y por utilizar la tecnología para mejorar la atención y llegar a más personas que disfruten de lo que hacemos.

### **1.3. Arquitectura Tecnológica Actual**

#### **Hardware:**

Actualmente, la panadería cuenta con algunas computadoras básicas donde llevan el control de ventas, pero no tienen equipos especializados ni sistemas digitales que integren toda la información.

#### **Software:**

Usan métodos sencillos como hojas de cálculo o registros manuales para apuntar las ventas y controlar los productos, sin una plataforma digital que facilite todo esto.

#### **Comunicaciones:**

Tienen conexión a Internet vía Wi-Fi, pero esta sólo se usa para cosas básicas, como ponerse en contacto con proveedores o buscar información, y no tienen aún un sistema en línea para gestionar la panadería o mostrar sus productos a los clientes.

## **CAPÍTULO 2 – MARCO TEÓRICO**

### **2.1. tecnologías Web**

#### **HTML (HyperText Markup Language)**

HTML es el lenguaje de marcado estándar utilizado para crear la estructura básica de las páginas web. Permite estructurar el contenido mediante etiquetas que definen elementos como encabezados, párrafos, imágenes, enlaces y formularios. Según Duckett (2011), HTML no es un lenguaje de programación, sino un lenguaje de marcado que describe la estructura de las páginas web.

#### **CSS (Cascading Style Sheets)**

CSS es un lenguaje que se utiliza para describir la presentación visual de los documentos HTML. Permite separar el contenido de la estructura visual, definiendo estilos como

colores, fuentes y posiciones. El uso de CSS mejora la accesibilidad y facilita el mantenimiento del sitio (Freeman & Robson, 2014).

## **Base de Datos**

Una base de datos es un sistema organizado para almacenar, gestionar y recuperar información de forma eficiente. Las bases de datos relacionales utilizan SQL y almacenan información en tablas, mientras que las NoSQL manejan estructuras más flexibles. Coronel y Morris (2018) destacan que las bases de datos permiten manejar grandes volúmenes de información de forma segura y eficiente.

## **JavaScript**

JavaScript es un lenguaje de programación interpretado que permite añadir interactividad a las páginas web. Permite manipular el DOM, responder a eventos del usuario y realizar peticiones asincrónicas. De acuerdo con Flanagan (2020), JavaScript se ha convertido en un lenguaje fundamental tanto en el cliente como en el servidor web.

## **Frameworks**

Un framework es una estructura tecnológica que proporciona un conjunto de herramientas, componentes reutilizables y convenciones que facilitan la creación de aplicaciones. A diferencia de una biblioteca, un framework impone una estructura y un flujo de trabajo, definiendo la arquitectura general del sistema e invocando el código del desarrollador a través del principio de inversión de control (IoC) (Fowler, 2004).

Un framework puede incluir módulos para el manejo de bases de datos, interfaces de usuario, seguridad, controladores, plantillas, entre otros. Su objetivo principal es agilizar el desarrollo, fomentar la reutilización del código y promover buenas prácticas de programación, reduciendo errores comunes y el tiempo de implementación.

Los frameworks son clasificados según el tipo de aplicación para el que están diseñados. Algunos de los más comunes son:

- Frameworks web backend: permiten construir la lógica del servidor y conectar con bases de datos. Ejemplos: Spring Boot (Java), Django (Python), Laravel (PHP).

- Frameworks frontend: ayudan a desarrollar interfaces dinámicas e interactivas en el navegador. Ejemplos: React, Angular, Vue.js.
- Frameworks móviles: facilitan la creación de aplicaciones para dispositivos móviles. Ejemplos: Flutter, React Native.

Uno de los principales beneficios de utilizar frameworks en el desarrollo web es que permiten implementar arquitecturas reconocidas como MVC (Modelo–Vista–Controlador), lo cual mejora la organización y mantenibilidad del código. Además, muchos frameworks ofrecen integración con herramientas modernas como motores de plantillas, ORM (Object-Relational Mapping), y sistemas de autenticación.

## **2.2 Arquitecturas Web.**

### **Introducción a las Arquitecturas Web.**

Las arquitecturas web hacen referencia a la forma en que se estructuran los componentes de una aplicación web para que funcionen adecuadamente en entornos distribuidos. Estas arquitecturas permiten dividir responsabilidades, facilitar el mantenimiento, y escalar los sistemas web con mayor facilidad (Shklar & Rosen, 2009).

#### **Arquitectura Cliente-Servidor**

Es la arquitectura más básica y ampliamente utilizada. En este modelo, el cliente (generalmente un navegador) realiza solicitudes que son atendidas por un servidor. El servidor responde con recursos, como páginas HTML o datos. Esta arquitectura es eficaz para aplicaciones simples, pero puede ser limitada en entornos de alto tráfico (Pressman & Maxim, 2015).

#### **Arquitectura de Tres Capas (Three-Tier)**

Esta arquitectura divide la aplicación en tres capas principales: presentación (interfaz de usuario), lógica de negocio (procesamiento de datos) y acceso a datos (base de datos). Esta separación permite mayor escalabilidad, mantenimiento y reutilización del código (Bass, Clements & Kazman, 2012).

#### **Arquitecturas Basadas en Servicios y APIs REST**

Las arquitecturas basadas en servicios, como REST (Representational State Transfer), permiten construir aplicaciones modulares y desacopladas. Mediante el uso de APIs, los distintos componentes pueden comunicarse entre sí de forma eficiente, permitiendo el desarrollo de aplicaciones móviles y web a partir de un mismo backend (Fielding, 2000).

### **Arquitectura de Microservicios**

La arquitectura de microservicios fragmenta la aplicación en pequeños servicios independientes que se comunican entre sí mediante APIs. Cada microservicio se encarga de una funcionalidad específica y puede desplegarse de manera individual, lo que permite una escalabilidad y resiliencia superiores (Newman, 2015).

### **Arquitectura MVC (Modelo-Vista-Controlador)**

La arquitectura MVC es un patrón de diseño de software que promueve la separación de responsabilidades dentro de una aplicación, dividiéndola en tres componentes principales: Modelo, Vista y Controlador. Este enfoque permite una mayor organización del código, facilitando su mantenimiento, escalabilidad y reutilización (Gamma et al., 1995).

- **Modelo:** Representa los datos y la lógica del negocio. Se encarga de gestionar el acceso a la base de datos, las reglas de validación y los procesos relacionados con la información que maneja el sistema.
- **Vista:** Es la capa encargada de la presentación. Muestra los datos al usuario y define la interfaz gráfica o web. En el desarrollo web, esto incluye plantillas HTML que se renderizan dinámicamente.
- **Controlador:** Actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario, procesa la lógica necesaria (interactuando con el Modelo) y determina qué respuesta o Vista devolver.

Este patrón permite el desarrollo paralelo del frontend y backend, mejora la legibilidad del código y promueve una arquitectura orientada a componentes, altamente utilizada en aplicaciones web modernas.

## **CAPÍTULO 3 – GESTIÓN DEL PROYECTO**

### **3.1. Objetivos**

#### **Objetivo General**

- Desarrollar un sistema web para la Panadería Amasarte, que permita gestionar eficientemente las ventas y la promoción de sus productos, facilitando la administración comercial y mejorando la experiencia del cliente.

#### **Objetivos Específicos**

- Diseñar y construir una base de datos en SQL Server que almacene la información de productos, ventas y usuarios.
- Implementar un sistema web utilizando el patrón MVC con Spring Boot, que permita la gestión de ventas.
- Facilitar el registro y seguimiento de ventas de productos mediante una interfaz amigable y segura.
- Facilitar la promoción y difusión de productos artesanales mediante funcionalidades que permitan destacar productos, novedades o recomendaciones en la plataforma web.
- Implementar herramientas en el sistema para crear campañas simples de difusión de productos
- Asegurar que el sistema sea escalable, mantenible y con un diseño modular para futuras mejoras.

### **3.2. Descripción del problema:**

La Panadería AmasArte actualmente utiliza un sistema básico para registrar las ventas diarias, común en muchas micro y pequeñas empresas. Sin embargo, este sistema presenta limitaciones para apoyar el crecimiento y expansión del negocio, especialmente en la gestión integral de productos, promociones y atención al cliente. En un contexto donde las oportunidades digitales son cada vez más relevantes, surge la necesidad de implementar un sistema web integrado que complemente la plataforma existente, facilitando el proceso de compra de forma eficiente y sin saturar los recursos operativos. Esta solución tecnológica permitirá a la panadería mejorar la difusión de sus productos artesanales, captar nuevos clientes y aumentar la fidelización de los habituales, condiciones esenciales para ampliar su competitividad en el mercado local. Asimismo,



un sistema estructurado y escalable garantizará un crecimiento ordenado del negocio, atendiendo la demanda progresiva sin comprometer la calidad del servicio ni la gestión interna.

### 3.3. Alcance del proyecto

#### Funcionalidades incluidas:

- **Gestión de ventas:** Registro de las ventas diarias de la panadería complementando el sistema básico actual para mejorar el control y la trazabilidad.
- **Gestión de productos:** Administración del catálogo de productos artesanales, con opciones para agregar, modificar y eliminar productos y precios actualizados.
- **Difusión y promoción de productos:** Herramientas para destacar productos en la plataforma web, mostrando novedades, productos recomendados o artesanales destacados, con contenidos visuales fáciles de actualizar para atraer clientes.
- **Interfaz web amigable:** Diseño de un portal web accesible desde diversos dispositivos (computadoras, tablets, smartphones) que facilite la consulta del catálogo y la interacción con los usuarios.
- **Gestión progresiva de compras:** Funcionalidades iniciales para facilitar la reserva o compra de productos desde la web, con un flujo simple y ordenado, evitando congestionar o saturar la operación actual del negocio.
- **Reportes básicos:** Generación de reportes sumarios sobre ventas y productos difundidos para apoyar la toma de decisiones.
- **Base de datos integrada:** Implementación de la base de datos en SQL Server para centralizar y organizar toda la información del sistema.
- **Arquitectura tecnológica:** Desarrollo bajo el patrón MVC utilizando Spring Boot, garantizando un diseño modular, escalable y mantenible.

#### Limitaciones y restricciones:

- La experiencia y conocimientos técnicos previos del equipo pueden limitar la complejidad y alcance de ciertas funcionalidades.
- No se incluirá inicialmente la integración con sistemas de pago electrónico ni plataformas externas para venta en línea inmediata.
- La gestión de inventarios será básica.

- El sistema estará diseñado para uso interno y atención progresiva a clientes, sin configurarse aún como un e-commerce completo de venta directa en línea.
- No se implementarán funcionalidades avanzadas de marketing digital, pero sí herramientas básicas para la difusión interna desde la plataforma.
- Se desarrollará una versión monolítica sin microservicios u otras arquitecturas distribuidas para evitar complejidad innecesaria en esta etapa inicial.

### **3.4. Metodología de desarrollo**

Para el desarrollo del sistema web de gestión de ventas y promoción de productos de la Panadería AmasArte, se aplicará la metodología ágil SCRUM, la cual facilita la gestión eficiente y flexible del proyecto en entornos con requisitos susceptibles a cambios y evolución progresiva. SCRUM permitirá al equipo trabajar en ciclos iterativos, en los que se planifican, desarrollan y revisan funcionalidades específicas con retroalimentación constante.

Este enfoque favorece la organización del trabajo en tareas pequeñas y manejables, promueve la comunicación continua entre los miembros del equipo, y habilita una rápida adaptación frente a nuevas necesidades o correcciones detectadas durante el desarrollo.

En el aspecto técnico, el proyecto aprovechará la arquitectura basada en Spring Boot y usará JPA (Java Persistence API) para la capa de persistencia, lo que contribuye a agilizar la gestión de la base de datos SQL Server mediante un mapeo objeto-relacional estándar. Esto encaja con la filosofía ágil de reutilización de herramientas robustas y estables para acelerar el desarrollo y minimizar errores. Por esto, el uso de SCRUM permitirá además incorporar pruebas y ajustes iterativos en cada sprint, asegurando la calidad y funcionalidad del sistema de manera progresiva y sostenible

### **3.5. Planificación y cronograma**

En esta estructura de actividades se describen las tareas organizadas por fases necesarias para completar el proyecto. Estas actividades están diseñadas considerando el nivel académico del equipo y la disponibilidad de tiempo, enfocándose en el desarrollo de un sistema web sencillo pero funcional que permita gestionar las ventas y la difusión de productos artesanales de la Panadería Dulce. Cada fase del proyecto se centra en pasos alcanzables, asegurando que el equipo pueda avanzar de manera progresiva y efectiva hacia los objetivos establecidos.

## **Planificación**

### **Fase 1: Investigación y Análisis Inicial (6 días)**

- Reunión del equipo para definir el alcance del proyecto y sus objetivos.
- Revisión de herramientas disponibles (SQL Server, Spring Boot y Thymeleaf).
- Análisis del proceso actual del negocio.

### **Fase 2: Diseño del Sistema (3 días)**

- Prototipo básico de la interfaz web usando Figma (pantalla de registro y consulta).

### **Fase 3: Avance del Prototipo(6 días)**

- Preparación de la documentación de acuerdo con el modelo proporcionado.
- Preparación de los esquemas y diapositivas a tratar.

### **Fase 4: Desarrollo y Configuración (7 días)**

- Configuración del entorno (instalación de Spring Boot, SQL Server y JPA).
- Creación de la base de datos y conexiones básicas.
- Desarrollo de funcionalidades para el registro y consulta de clientes, así como la gestión de ventas y promoción de productos.

### **Fase 5: Pruebas y Ajustes (5 días)**

- Realización de pruebas básicas (guardar, consultar, actualizar).
- Ajustes según errores identificados en pruebas.

### **Fase 6: Documentación Final (6 días)**

- Preparación de la documentación de acuerdo con el esquema proporcionado.
- Preparación de los esquemas y diapositivas a tratar.

### **Fase 7: Presentación Final (1 día)**

- Sustentación y cierre del proyecto.

## **Cronograma**

Fase	Actividad	Duración	Fecha
<b>Fase 1</b>	Reunión Inicial.	1 día	04/04/2025
	Revisión de herramientas.	3 días	05/04/2025 06/04/2025 07/04/2025
	Análisis del proceso actual.	2 días	07/04/2025 08/04/2025
<b>Fase 2</b>	Prototipo básico de la interfaz web usando Figma.	3 días	09/04/2025 10/04/2025 11/04/2025
<b>Fase 3</b>	Preparación de la documentación.	3 días	14/04/2025 15/04/2025 16/04/2025
	Preparación de los esquemas y diapositivas a tratar.	3 días	18/04/2025 19/04/2025 20/04/2025
<b>Fase 4</b>	Configuración del entorno.	2 días	25/04/2025 26/04/2025
	Creación de la base de datos y conexiones básicas.	2 días	27/04/2025 28/04/2025
	Desarrollo de funcionalidades.	3 días	02/05/2025 03/05/2025 04/05/2025
<b>Fase 5</b>	Realización de pruebas básicas	2 días	09/05/2025 10/05/2025
	Ajustes según errores identificados en pruebas.	3 días	11/05/2025 12/05/2025 13/05/2025
<b>Fase 6</b>	Preparación de la documentación final.	3 días	07/07/2025 08/07/2025 09/07/2025
	Preparación de los esquemas y diapositivas a tratar.	3 días	10/07/2025 11/07/2025 12/07/2025
<b>Fase 7</b>	Sustentación y cierre del proyecto.	1 día	14/07/2025

## CAPÍTULO 4 – IMPLEMENTACIÓN DE ESTRATEGIA

### 4.1. Lista de Requerimientos

#### Requerimientos Funcionales

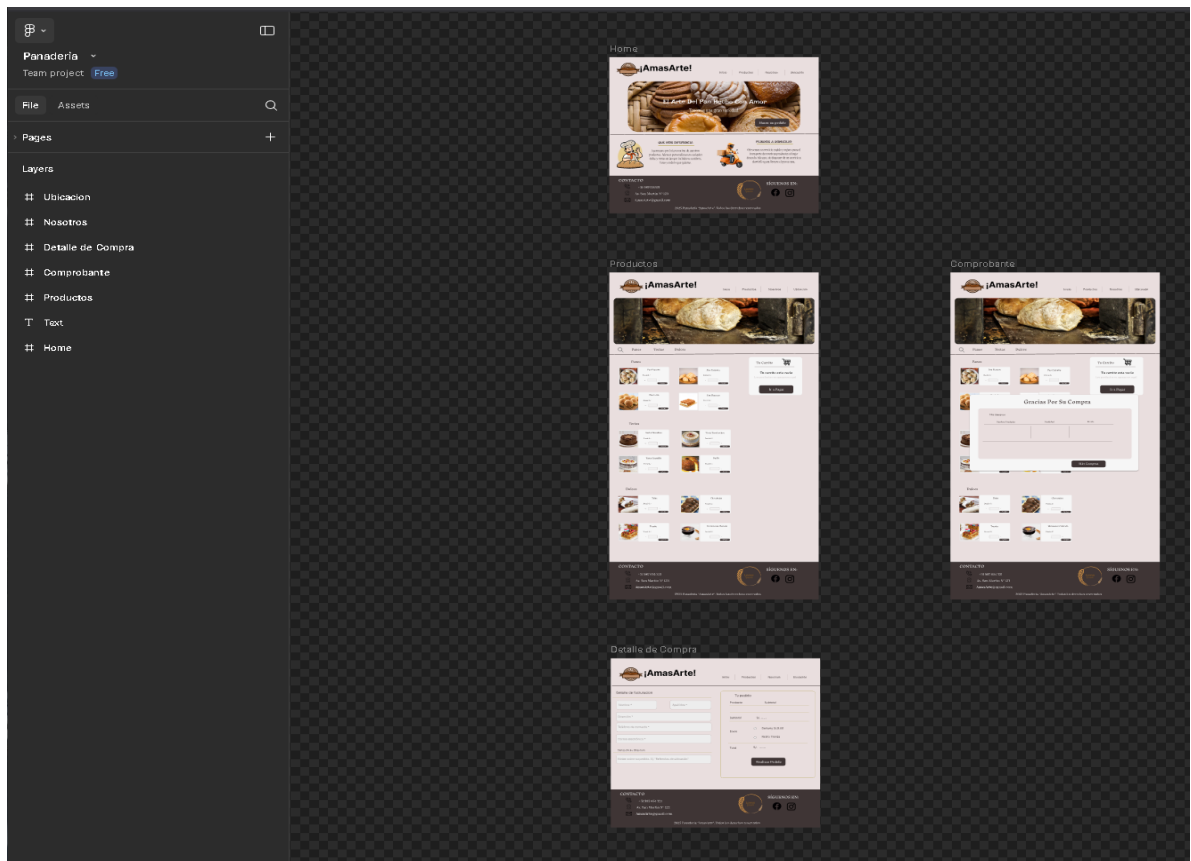
- Registrar y gestionar clientes para fidelización
- Gestión y difusión de productos artesanales
- Simplificación de la Visualización de datos
- Proveer una plataforma web
- Generar reportes básicos sobre ventas y productos difundidos para soporte a la toma de decisiones.
- Interfaz web responsiva, la plataforma debe ser accesible y usable desde dispositivos móviles y de escritorio.

#### **Requerimientos no Funcionales**

- Seguridad: Implementar autenticación segura y control de acceso para proteger datos sensibles de usuarios y ventas.
- Escalabilidad: El sistema debe estar diseñado para soportar un crecimiento gradual en usuarios y volumen de datos sin perder rendimiento.
- Automatización respaldo: Realizar respaldos automáticos periódicos de la base de datos para evitar pérdida de información.
- Usabilidad: Interfaz intuitiva y sencilla que facilite la navegación y operación para usuarios con conocimientos básicos.
- Compatibilidad: El sistema debe ser compatible con los principales navegadores web.
- Rendimiento: La aplicación debe responder a interacciones del usuario en menos de 3 segundos bajo carga normal.

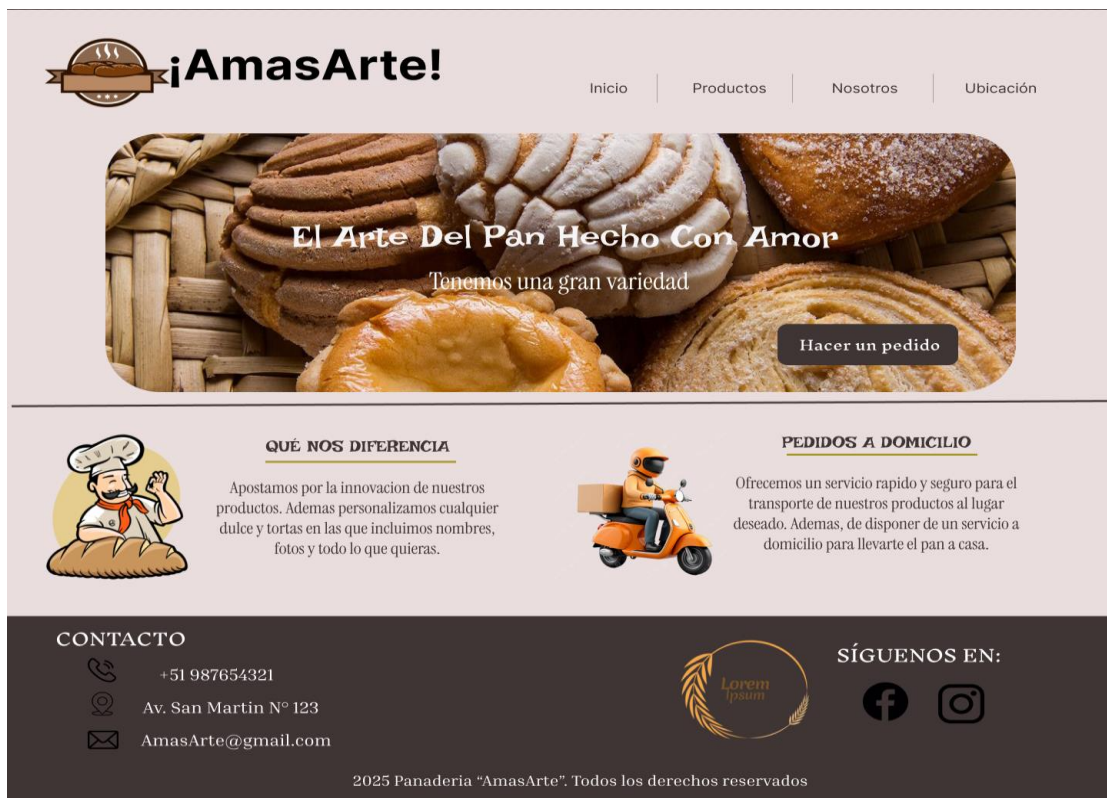
#### **4.2. Prototipo Preliminar**

En esta etapa, presentamos la maqueta de la página web diseñada para la panadería AmasArte, creada con Figma. El diseño busca transmitir calidez y cercanía. Por lo cual esta interfaz es amigable e intuitiva, permitiendo a los usuarios explorar el catálogo, conocer promociones y realizar pedidos.



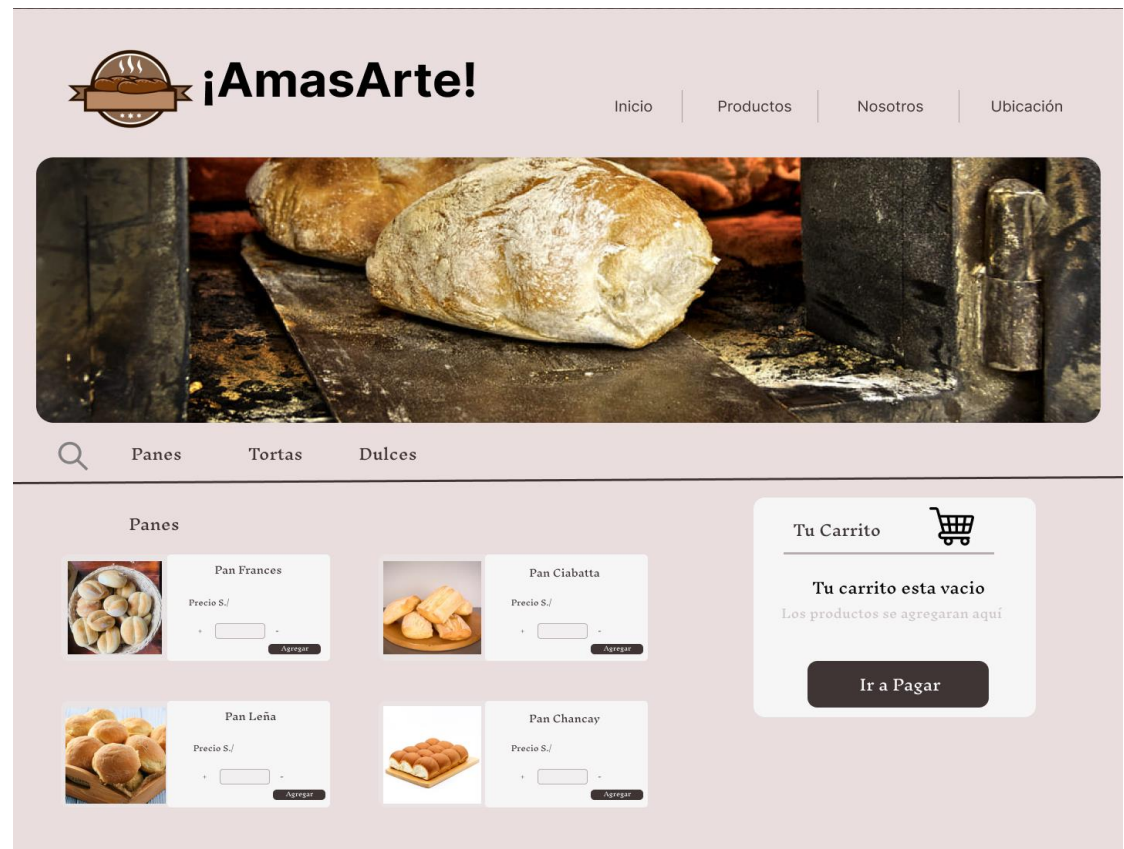
## Home

Es la Página Principal de nuestro sitio web, con esto esperamos cautivar al cliente mediante una presentación simple y efectiva.



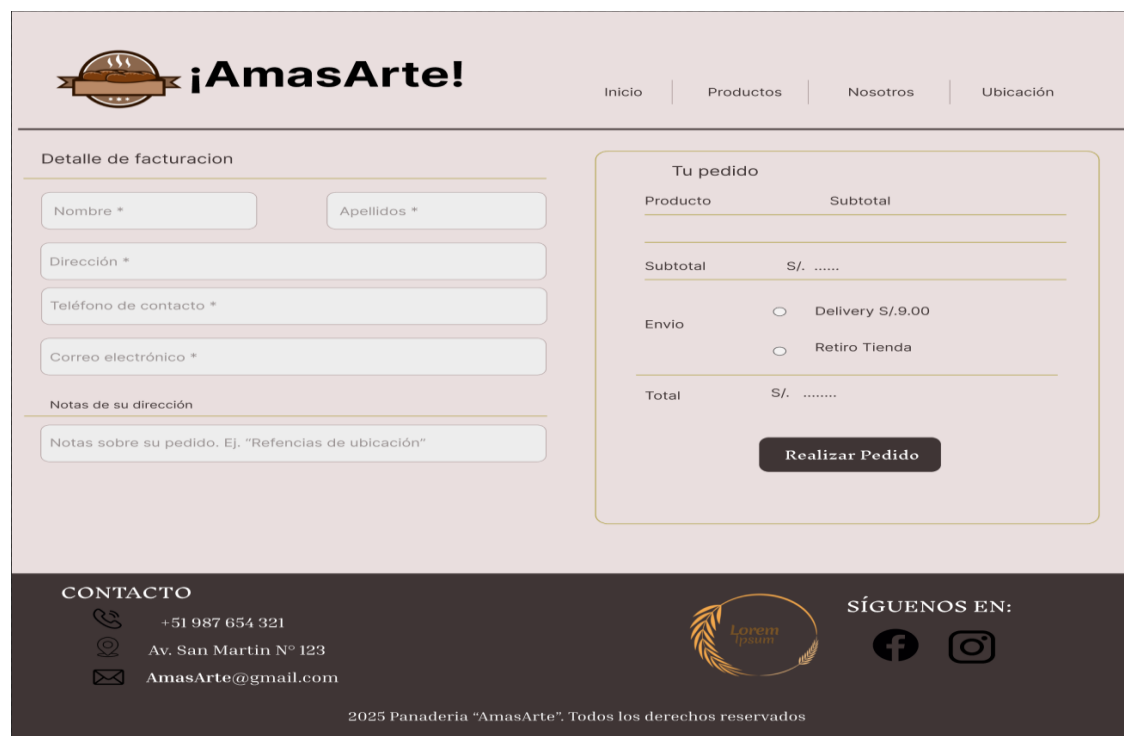
## Productos

En esta sección se enlistan todos los productos que comercializa nuestro negocio.




## Detalle de Compra

Detalla de manera organizada los datos del cliente y las preferencias de la compra. Para mejorar su experiencia.




## Ubicación

En esta sección podrás conocer la dirección exacta de nuestra panadería, visualizar nuestra ubicación en el mapa y acceder fácilmente a nuestros datos de contacto.




# ¡AmasArte!

[Inicio](#) | [Productos](#) | [Nosotros](#) | [Ubicación](#)




## UBICANOS

### Map





### ENCUENTRANOS:


Actualmente contamos con una única sede, desde la cual atendemos con dedicación a todos nuestros clientes, garantizando un servicio cercano y personalizado.


 Ave. San Martín N° 123

### CONTACTO



 +51 987 654 321

 Av. San Martín N° 123

 AmasArte@gmail.com




### SÍGUENOS EN:



2025 Panadería "AmasArte". Todos los derechos reservados


## Nosotros

En esta sección te contamos quiénes somos y que nos inspiran a ofrecer productos de calidad con un servicio cercano y apasionado.



# ¡AmasArte!


[Inicio](#) | [Productos](#) | [Nosotros](#) | [Ubicación](#)





## Nosotros


En nuestra panadería, creemos que el pan no solo alimenta, sino que une. Cada día horneamos con pasión, dedicación y los mejores ingredientes, para llevar a tu mesa ese sabor que reconforta y acompaña los momentos más especiales, con un toquecito de nuestra especialidad, el sabor y la calidad van de la mano.

### CONTACTO



 +51 987 654 321

 Av. San Martín N° 123

 AmasArte@gmail.com



### SÍGUENOS EN:



2025 Panadería "AmasArte". Todos los derechos reservados

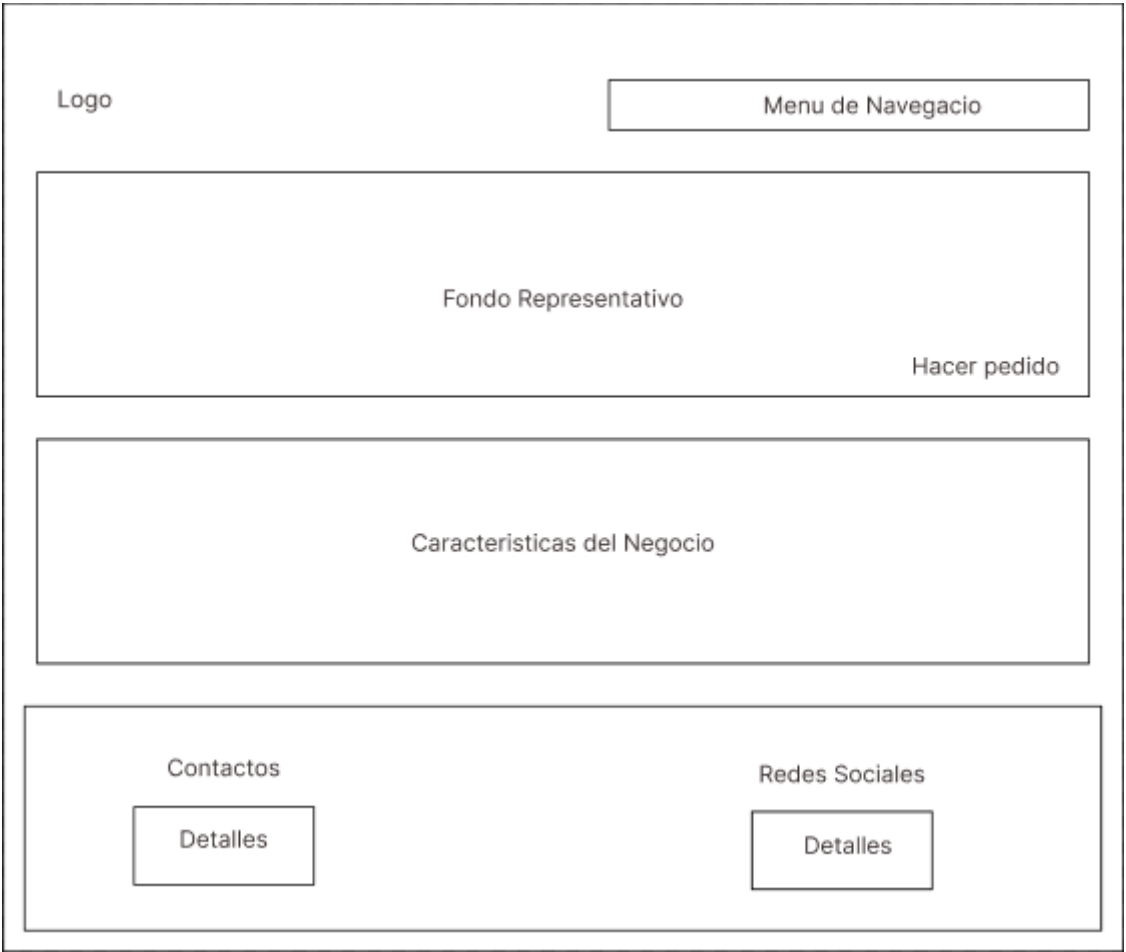


### 4.3 Prototipo Funcional

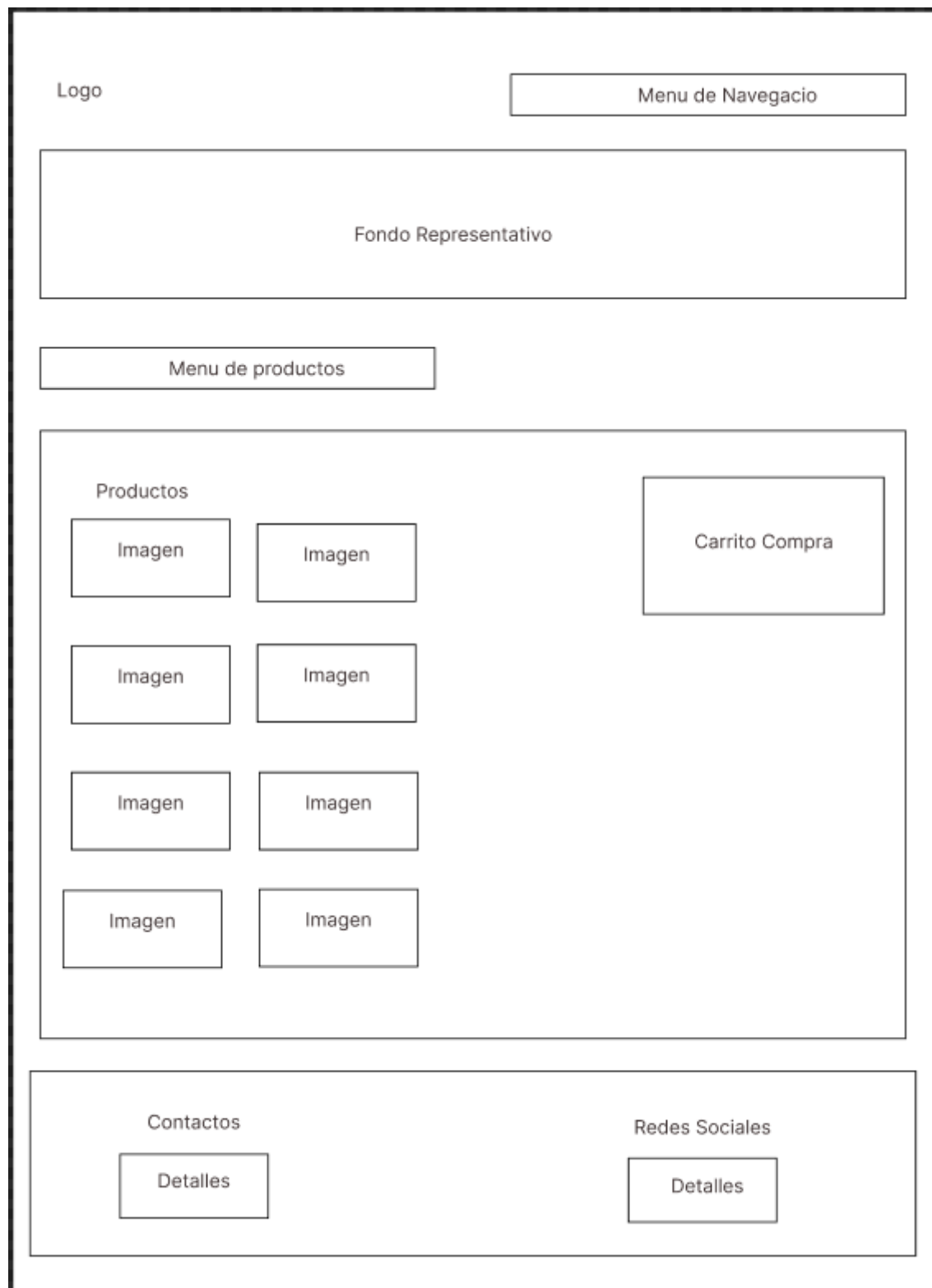
Antes de implementar la lógica del sistema, se ha realizado un prototipo funcional utilizando únicamente HTML y CSS. El objetivo de esta etapa es plasmar visualmente las ideas de diseño, estructura de contenido y navegación del sitio web. Estos prototipos permiten validar la distribución de los elementos y la experiencia de usuario sin necesidad de programación avanzada o conexión con bases de datos.

A continuación, se presentan los diferentes bocetos correspondientes a las secciones principales del sitio.

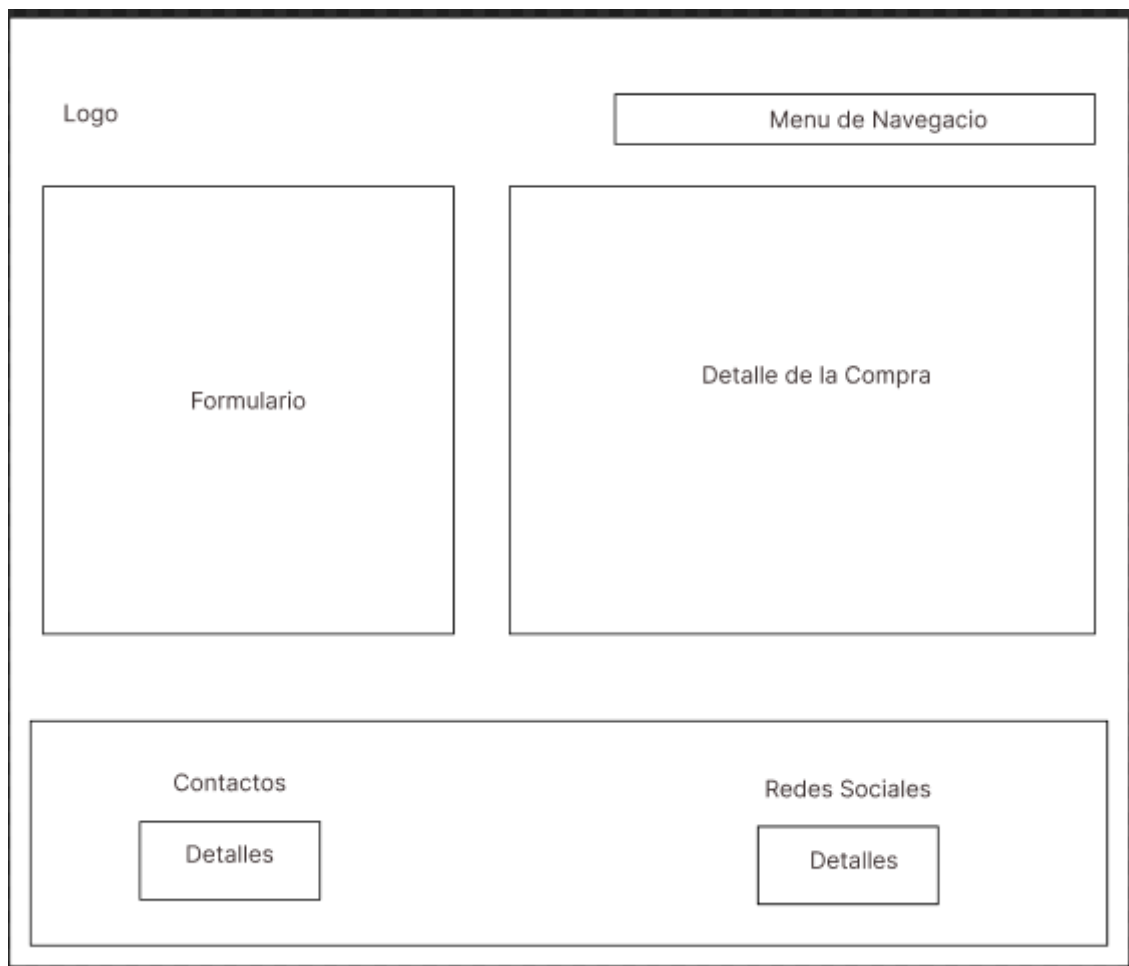
#### Inicio



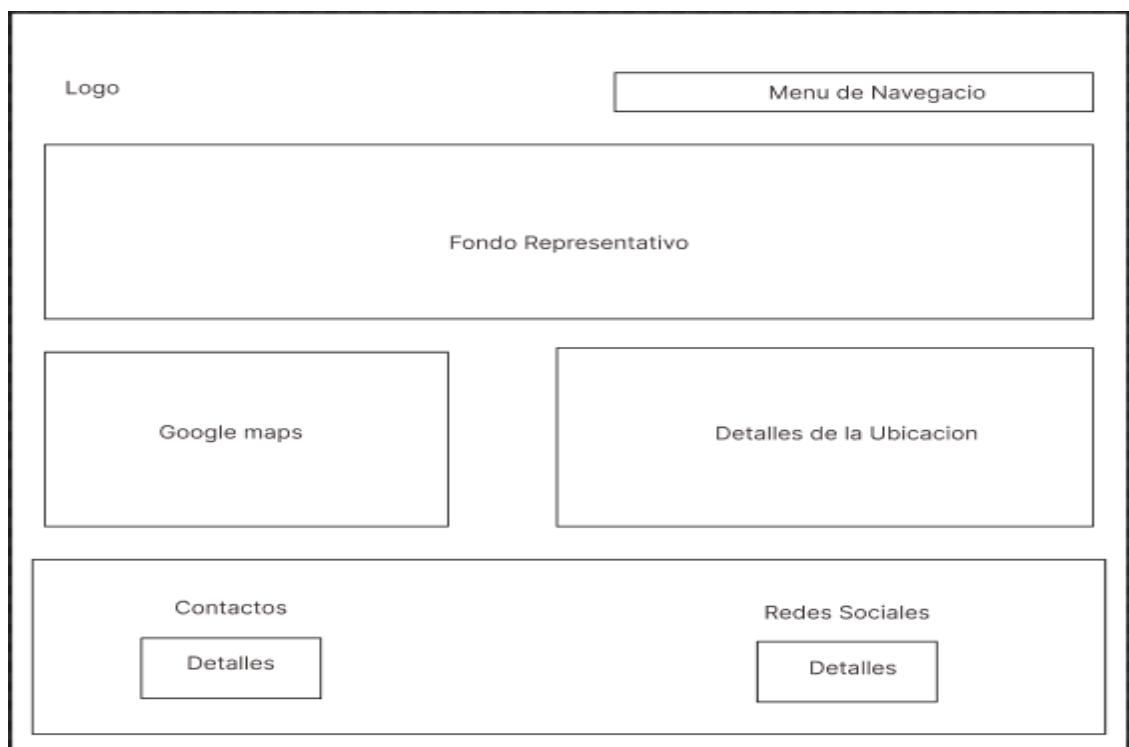
#### Productos



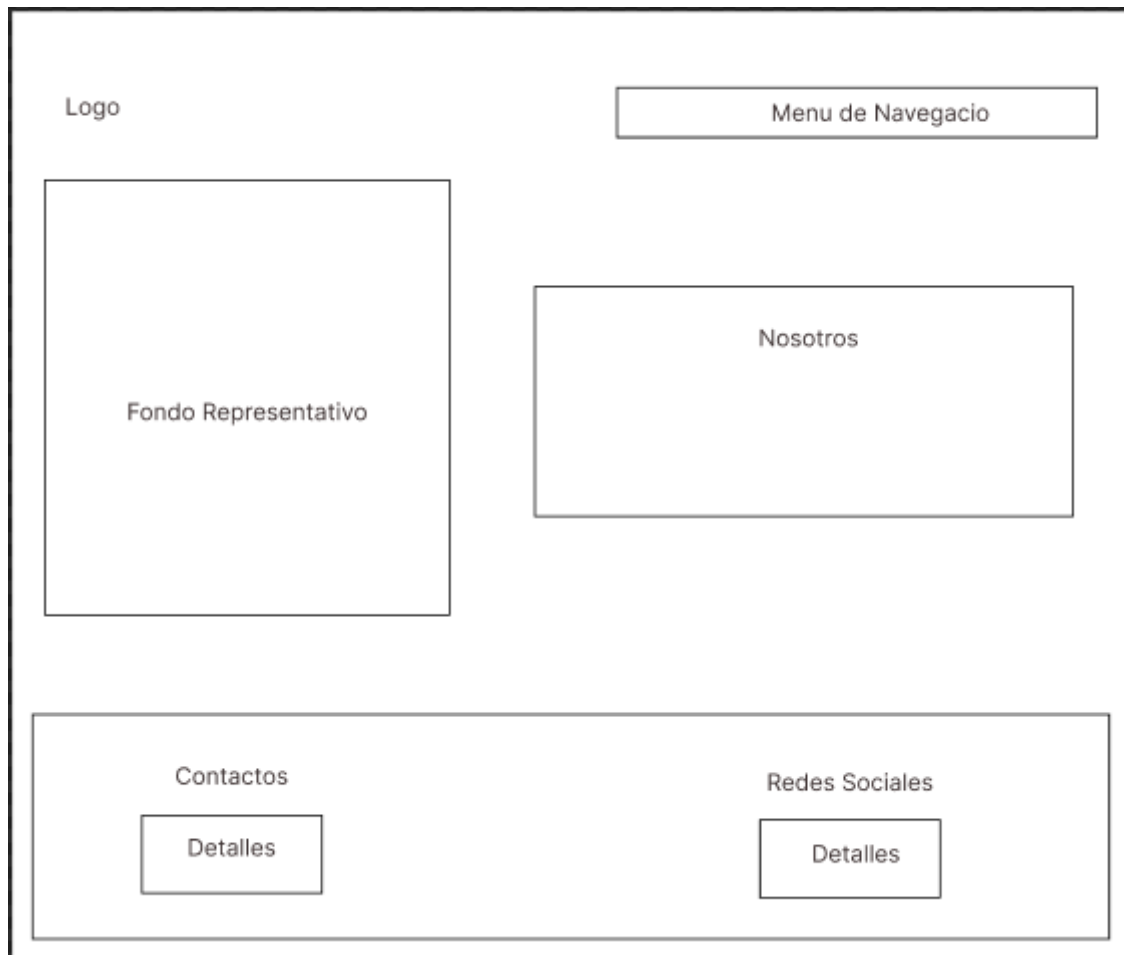
## Detalle Venta



## Ubicación



## Nosotros



#### 4.4. Desarrollo del sitio web

El sitio web de la panadería Amasarte fue desarrollado utilizando una arquitectura basada en capas, que separa las responsabilidades del sistema en frontend, backend y base de datos. Esta división facilita la escalabilidad, el mantenimiento y la organización del código.

##### *Lenguajes y tecnologías utilizadas:*

- HTML5 y CSS3: Para la estructura y diseño visual de las páginas.
- JavaScript: Para la gestión del carrito de compras, interacción dinámica en el navegador y envío de datos al servidor mediante fetch.
- Java: Lenguaje principal para el backend.
- Spring Boot: Framework utilizado para desarrollar la lógica del lado del servidor y exponer endpoints RESTful.
- Spring Data JPA: Para la persistencia de datos a través de entidades mapeadas.
- Spring Security: Para la autenticación de usuarios mediante correo electrónico y contraseña, y restricción de acceso a páginas administrativas.

- Thymeleaf: Motor de plantillas utilizado para renderizar páginas HTML con datos del
- backend.
- SQL Server: Motor de base de datos utilizado para almacenar productos, clientes, pedidos y detalles de pedidos.
- Lombok: Para reducir el código repetitivo (getters, setters, constructores).

### ***Estructura del proyecto***

El proyecto fue organizado en los siguientes paquetes:

- model: Contiene las clases de entidad como Producto, Cliente, Pedido y DetallePedido.
- repository: Interfaces JPA para acceder a la base de datos.
- controller: Controladores encargados de manejar las peticiones HTTP (@RestController o @Controller).
- service: Contiene la lógica de negocio (opcional en algunas funcionalidades).
- security: Clases relacionadas con la configuración de Spring Security y autenticación personalizada.

### ***Funcionalidades implementadas***

- Visualización de productos con opción de "Agregar al carrito".
- Gestión de carrito con almacenamiento temporal en localStorage y envío al backend.
- Registro y login de usuarios con validación de correo electrónico y contraseña.
- Autocompletado de datos del cliente al realizar pedidos si está logueado.
- Panel de administración para registrar productos, incluyendo la carga de imágenes.
- Restricción de acceso a páginas protegidas mediante roles (por ejemplo, solo el administrador puede acceder al panel de gestión de productos).

### ***Diseño y adaptabilidad***

El diseño fue desarrollado utilizando HTML y CSS personalizados, manteniendo la coherencia visual y una buena experiencia de usuario. Para asegurar la compatibilidad con dispositivos móviles, se aplicaron reglas de diseño responsivo, empleando flexbox, media queries y estructuras adaptables.

### ***Herramientas utilizadas***

- Visual Studio Code: IDE principal para el desarrollo frontend.

- IntelliJ IDEA / Spring Tool Suite: Para el desarrollo backend con Spring Boot.
- Postman: Para probar endpoints del backend.
- Git: Para el control de versiones del proyecto.

#### **4.5. Desarrollo de la BD**

##### Estructura de la Base de Datos

La base de datos está diseñada para manejar las entidades principales: Cliente, Producto, Pedido y DetallePedido. Cada una de estas entidades tiene una estructura con campos que se utilizan para almacenar la información relevante.

##### Entidades Principales:

La tabla Cliente guarda la información de los clientes que realizan pedidos.

##### Campos:

- id (PK): Identificador único del cliente.
- apellidos: Apellidos del cliente.
- correo: Correo electrónico del cliente (único).
- direccion: Dirección de envío del cliente.
- nombre: Nombre del cliente.
- password: Contraseña (almacenada de forma segura).
- telefono: Teléfono del cliente.

##### Producto:

La tabla Producto contiene los productos disponibles para la venta.

##### Campos:

- id (PK): Identificador único del producto.
- descripcion: Descripción detallada del producto.
- imagen\_url: URL de la imagen del producto.
- nombre: Nombre del producto.
- precio: Precio unitario del producto.

##### Pedido:

La tabla Pedido contiene la información de los pedidos realizados por los clientes.

##### Campos:

- id (PK): Identificador único del pedido.

- envío: Información sobre el tipo de envío.
- fecha: Fecha de creación del pedido.
- notas: Notas adicionales sobre el pedido.
- total: Total del pedido.
- cliente\_id (FK): ID del cliente que realizó el pedido (relación con la tabla Cliente).

DetallePedido:

La tabla DetallePedido almacena los detalles de los productos en cada pedido.

Campos:

- id (PK): Identificador único del detalle del pedido.
- cantidad: Cantidad de productos en el pedido.
- subtotal: Subtotal por cada producto en el pedido.
- pedido\_id (FK): ID del pedido al que pertenece este detalle (relación con la tabla Pedido).
- producto\_id (FK): ID del producto que se pidió (relación con la tabla Producto).

Relaciones entre Entidades

- Cliente -> Pedido: Un cliente puede realizar múltiples pedidos, lo que genera una relación de uno a muchos entre Cliente y Pedido. La clave foránea cliente\_id en la tabla Pedido se refiere al id de Cliente.
- Pedido -> DetallePedido: Un pedido puede tener múltiples productos, por lo que existe una relación de uno a muchos entre Pedido y DetallePedido. La clave foránea pedido\_id en la tabla DetallePedido se refiere al id de Pedido.
- Producto -> DetallePedido: Un producto puede aparecer en múltiples pedidos, por lo que hay una relación de uno a muchos entre Producto y DetallePedido. La clave foránea producto\_id en la tabla DetallePedido se refiere al id de Producto.

## 4.6. Pruebas y Validaciones para la Página Web de una Panadería

### 1. Pruebas Funcionales

- Navegación: Todos los menús, botones y enlaces internos fueron verificados. Se confirmó que conducen correctamente a sus respectivas secciones (Inicio, Productos, Acerca de, Contacto, etc.).
- Formularios: El formulario de contacto fue probado con distintos tipos de datos. Se validó el envío exitoso de mensajes, así como la verificación de campos obligatorios y formatos válidos.

- Pedidos en línea / carrito (si aplica): Se comprobó el correcto funcionamiento al agregar, modificar y eliminar productos. Las órdenes se generan correctamente.
- Pasarela de pago (si aplica): Se realizaron pruebas simuladas y reales. Los pagos se procesan con éxito y se genera confirmación al cliente.

## *2. Pruebas de Usabilidad*

- La navegación del sitio es fluida, intuitiva y clara para cualquier usuario.
- La estructura y jerarquía visual permiten una experiencia agradable.
- El diseño responsivo se adapta sin problemas a distintos tamaños de pantalla (escritorio, móvil y Tablet).

## *3. Pruebas de Rendimiento*

- El sitio carga en un promedio de 1.8 segundos, incluso con imágenes de alta calidad.
- Se realizaron pruebas de carga con tráfico simulado. La web respondió correctamente sin interrupciones ni caídas.
- La optimización de recursos (caché, compresión y carga diferida) está implementada.

## *4. Pruebas de Seguridad*

- La validación de entradas bloquea correctamente intentos de inyecciones SQL y XSS.
- Toda la comunicación del sitio está cifrada mediante HTTPS.
- Los datos de usuario y transacciones están protegidos con medidas actualizadas de seguridad.

## *5. Pruebas de Compatibilidad*

- El sitio fue probado con éxito en los navegadores: Chrome, Firefox, Safari, Microsoft Edge y Opera.
- Se verificó el funcionamiento correcto en sistemas operativos: Windows, macOS, iOS y Android.

## *6. Pruebas de Contenido*

- No se detectaron errores ortográficos ni gramaticales.
- Las descripciones, nombres y precios de los productos están correctamente actualizados y alineados con la oferta de la panadería.

## **Conclusión General de evaluación**



- La página web ha superado satisfactoriamente casi todas las pruebas funcionales, técnicas y de diseño. Se encuentra 70% lista para su lanzamiento y uso por parte del público, garantizando una experiencia estable, segura y amigable.

#### **4.7. Optimización y mejoras**

En este punto, se analizan diversas estrategias para mejorar el rendimiento, la accesibilidad y el SEO de nuestro proyecto web "Amasarte", un sistema de panadería.

*Optimización del Rendimiento:*

- **Uso de Caché:** Implementar caché para las respuestas más frecuentes, como las consultas de productos o pedidos, usando herramientas como `@Cacheable` de Spring. Esto reducirá el tiempo de respuesta y la carga sobre la base de datos.
- **Optimización de Consultas SQL:** Verificar las consultas SQL generadas por JPA para asegurarse de que no haya problemas de N+1 queries, utilizando `fetch = FetchType.LAZY` donde sea necesario y `JOIN FETCH` para evitar consultas innecesarias.
- **Compresión de Archivos Estáticos:** Asegurarse de que todos los archivos estáticos (CSS, JavaScript e imágenes) están comprimidos y minificados. Utilizar herramientas como Webpack para realizar estas tareas automáticamente.
- **CDN para Recursos Estáticos:** Considerar el uso de una Red de Distribución de Contenidos (CDN) para servir archivos estáticos (imágenes, hojas de estilo y scripts), lo que reducirá la latencia de carga para los usuarios fuera del servidor principal.

*Mejoras en la Accesibilidad:*

- **Etiquetas ARIA:** Asegurar que todos los elementos interactivos de la interfaz de usuario (botones, enlaces, formularios) tengan etiquetas ARIA apropiadas para mejorar la experiencia de usuarios con discapacidades. Por ejemplo, agregar `aria-label` a los botones de acción o `aria-live` a los elementos que se actualizan dinámicamente.
- **Contraste de Colores:** Garantizar que el contraste entre el texto y el fondo cumpla con los estándares WCAG 2.1 para facilitar la lectura de usuarios con dificultades visuales.

- Soporte para Navegación por Teclado: Asegurarse de que todos los elementos interactivos sean accesibles mediante teclado, permitiendo que los usuarios sin ratón puedan navegar fácilmente por el sitio.
- Texto Alternativo para Imágenes: Implementar descripciones claras y concisas en el atributo de todas las imágenes, para que los lectores de pantalla puedan proporcionar a los usuarios con discapacidades visuales una descripción de las imágenes.

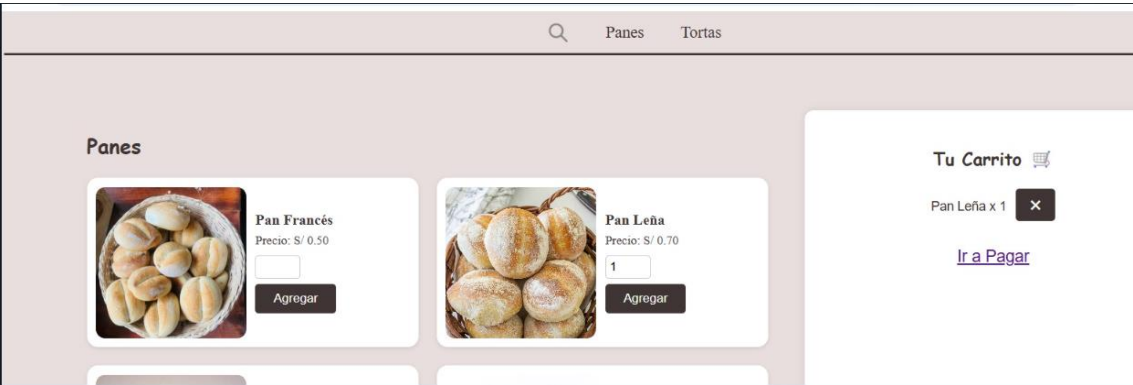
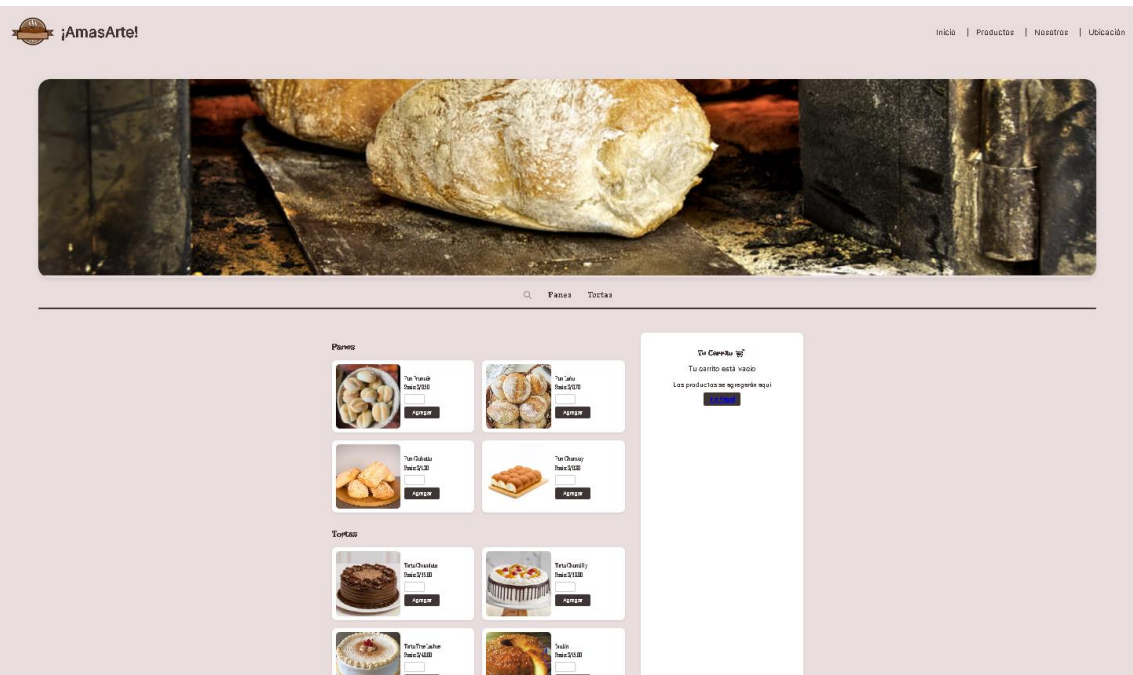
#### *Optimización para SEO:*

- Metadatos Adecuados: Asegurarse de que cada página tenga metadatos adecuados como títulos (<title>) y descripciones (<meta name="description">), los cuales deben ser únicos y relevantes para el contenido de la página.
- URLs Amigables: Las rutas de las páginas deben ser amigables para los motores de búsqueda, utilizando nombres descriptivos y evitando caracteres especiales o IDs complicados en las URLs.
- Uso de Etiquetas HTML Semánticas: Utilizar correctamente las etiquetas HTML semánticas (por ejemplo, <header>, <footer>, <article>, <section>, <main>) para estructurar el contenido de la página, lo que ayuda a los motores de búsqueda a entender mejor el contenido.
- Optimización de Imágenes: Asegurarse de que las imágenes están optimizadas tanto en tamaño como en formato. Usar formatos modernos como WebP puede reducir significativamente el tamaño de las imágenes sin pérdida de calidad.
- Mapa del Sitio (Sitemap): Crear y enviar un archivo sitemap.xml a los motores de búsqueda para facilitar el rastreo de las páginas del sitio.

## **CAPITULO 5 - CIERRE Y DOCUMENTACION**

### **5.1. Versión Final de producto**

Presentaremos la versión final de nuestro Proyecto web



## ¿Tienes una cuenta?

Para continuar con tu compra, por favor inicie sesion:

Iniciar Sesión

Cancelar

## Iniciar Sesión

Correo electrónico

Contraseña

Entrar

¿No tienes cuenta aún? [Regístrate aquí](#)

## Crear cuenta

Registrarse

¿Ya tienes cuenta? [Inicia sesión](#)

### Detalle de facturación

Notas de su dirección

### Tu pedido

Pan Leña (2 x S/ 0.70)	S/ 1.40
Pan Francés (1 x S/ 0.50)	S/ 0.50
Pan Ciabatta (1 x S/ 1.20)	S/ 1.20
Subtotal	S/ 3.10

#### Envío

- ☒ Delivery S/9.00  
☐ Retiro Tienda

Total S/ 12.10

Realizar Pedido

Cancelar Pedido

localhost:8080 dice  
¡Gracias por su compra! 🍕

### Detalle de facturación

Pedro Luis

calle san isidro

943523361

pedroluisperezllamas@gmail.com

Perez Llamas

Notas de su dirección

Todo muy rico

### Tu pedido

Pan Leña (2 x S/ 0.70) S/ 1.40

Pan Francés (1 x S/ 0.50) S/ 0.50

Pan Ciabatta (1 x S/ 1.20) S/ 1.20

Subtotal S/

#### Envío

☒ Delivery S/9.00
 ☐ Retiro Tienda

#### Total S/

Realizar Pedido

Cancelar Pedido

En nuestra BD SQL Server

```
use amasarte
```

```
select * from cliente;
```

```
select * from detalle_pedido;
```

```
select * from pedido;
```

```
select * from producto;
```

	id	apellidos	correo	direccion	nombre	password	telefono
1	1	perez	pedro@gmail.com	calle san isidro	pedro	pedro	943523361
2	2	pe	pe@gmail.com	d	pedro	pe	94
3	3	b	e@gmail.com	c	a	f	d
4	4	edy	edy@gmail.com	edy	edy	edy	1

	id	cantidad	subtotal	pedido_id	producto_id
1	16	5	3,5	8	2
2	17	9	4,5	8	1
3	18	1	35	8	5

	id	envio	fecha	notas	total	cliente_id
1	8	delivery	2025-06-30	Hola edy muy rico como fabianita	52	4

	id	descripcion	imagen_url	nombre	precio
1	1	Pan crocante tradicional	img/PanFrances.jpg	Pan Francés	0,5
2	2	Pan hecho al horno de leña	img/pan-leña.jpg	Pan Leña	0,7
3	3	Pan de estilo italiano	img/Pan-ciabatta.jpg	Pan Ciabatta	1,2
4	4	Pan dulce tradicional	img/PanChancay.jpg	Pan Chancay	0,8
5	5	Deliciosa torta de chocolate	img/Chocolate.jpeg	Torta Chocolate	35
6	6	Torta con crema chantilly	img/Chantilly.png	Torta Chantilly	38
7	7	Bizcocho con tres leches	img/torta-3-leches.j...	Torta Tres Lec...	40
8	8	Postre tradicional de pan	img/Budin.jpg	Budín	15

## 5.2 Manuales del usuario

### Manual de Usuario - Sistema Amasarte

#### Introducción

Bienvenido al sistema de gestión de panadería Amasarte, donde puedes comprar productos como panes y tortas de manera fácil y rápida. Este manual te guiará a través del proceso de registro, navegación de productos, compra y validación de pedidos.

#### Requisitos del Sistema

- Para utilizar el sistema, necesitarás:
- Un navegador web actualizado (Google Chrome, Firefox, Safari, etc.).
- Conexión a Internet.
- Una cuenta de usuario en el sistema (requiere registro previo).

### Acceso al Sistema

#### 1. Registro de Usuario

Si eres un nuevo usuario, debes registrarte para realizar compras:

1. Dirígete a la página principal de Amasarte.
2. Haz clic en "Crear Cuenta".
3. Completa los campos requeridos:
  - Nombre
  - Apellidos
  - Correo Electrónico
  - Teléfono
  - Dirección (para la entrega de productos)
  - Contraseña
4. Haz clic en "Registrar" para crear tu cuenta.

Recibirás un correo de confirmación y podrás acceder al sistema.

#### 2. Iniciar Sesión

Para acceder al sistema:

1. En la página principal, haz clic en "Iniciar Sesión".
2. Ingresa tu correo electrónico y contraseña.
3. Haz clic en "Acceder" para entrar al sistema.

### Uso del Sistema

#### 1. Navegación de Productos

Una vez que inicies sesión, verás los productos disponibles en la tienda, como panes y tortas.

1. Los productos se listan en la página con su nombre y precio.
2. Cuando decidas agregar un producto a tu pedido, selecciona la cantidad y haz clic en "Realizar Compra".

## **2. Realizar Pedido (Detalle de la Venta)**

Al hacer clic en "Realizar Compra", serás redirigido a la página de Detalle de Compra:

1. Si aún no has completado tus datos, el sistema te pedirá ingresar la siguiente información:

- Nombre
- Apellidos
- Dirección (para la entrega)
- Teléfono
- Correo electrónico

2. El sistema mostrará un resumen del pedido, que incluirá los productos seleccionados, la cantidad, el subtotal y el total a pagar.
3. Revisa tu pedido, y cuando todo esté correcto, haz clic en "Confirmar Pedido" para finalizar la compra.

## **3. Validación de la Compra**

Una vez que confirmes el pedido:

1. El sistema validará tu compra y te mostrará un mensaje de confirmación.
2. El mensaje incluirá los detalles de tu pedido, como los productos seleccionados y el número de pedido.

Si la compra fue exitosa, recibirás la confirmación del pedido.



## Bibliografias

Coronel, C., & Morris, S. (2018). \*Database systems: Design, implementation, & management\* (13th ed.). Cengage Learning.

Duckett, J. (2011). \*HTML and CSS: Design and build websites\*. Wiley.

Flanagan, D. (2020). \*JavaScript: The definitive guide\* (7th ed.). O'Reilly Media.

Freeman, E., & Robson, E. (2014). \*Head First HTML and CSS\* (2nd ed.). O'Reilly Media.

Bass, L., Clements, P., & Kazman, R. (2012). \*Software Architecture in Practice\* (3rd ed.). Addison-Wesley.

Fielding, R. T. (2000). \*Architectural Styles and the Design of Network-based Software Architectures\* (Doctoral dissertation, University of California, Irvine).

Newman, S. (2015). \*Building Microservices: Designing Fine-Grained Systems\*. O'Reilly Media.

Pressman, R. S., & Maxim, B. R. (2015). \*Software Engineering: A Practitioner's Approach\* (8th ed.). McGraw-Hill Education.

Shklar, L., & Rosen, R. (2009). \*Web Application Architecture: Principles, Protocols and Practices\* (2nd ed.). Wiley.