



**Instituto Politécnico Nacional**  
Unidad Profesional Interdisciplinaria de Ingeniería, Campus Tlaxcala (UPIIT)

# **Proyecto: "Flor y Hojaldra"**

## **Reporte de Documentación - Versión 1.0**

**Alumno:** Edgar Emmanuel Águila Guzmán

Estefanny Sanluis Carcaño

Guadalupe Brised Checa Linarte

**Profesor:** Jesús Rojas Hernández

**Materia:** Tecnologías para el Desarrollo de Aplicaciones Web

**Carrera:** Ingeniería en Inteligencia Artificial

**Grupo:** 4IV1

14 de noviembre de 2025

# Índice

<b>1. Introducción al Proyecto</b>	<b>3</b>
1.1. Objetivos de la Versión 1.0 . . . . .	3
1.2. Stack Tecnológico . . . . .	3
<b>2. Análisis y Diseño</b>	<b>3</b>
2.1. Actores del Sistema . . . . .	3
2.2. Casos de Uso (CU) V1.0 . . . . .	4
<b>3. Diseño de la Base de Datos</b>	<b>4</b>
3.1. Archivo de Conexión (config/db_connect.php) . . . . .	5
<b>4. Estructura de Archivos del Proyecto</b>	<b>6</b>
<b>5. Descripción Detallada de Archivos PHP</b>	<b>8</b>
5.1. Módulo de Tienda (Frontend - Raíz) . . . . .	8
5.2. Módulo de Administración (Backend - /admin) . . . . .	8
5.2.1. Autenticación y Seguridad . . . . .	8
5.2.2. Gestión de Productos (CU-06) . . . . .	8
5.2.3. Gestión de Pedidos y Ventas (CU-02, 04, 05) . . . . .	9
5.2.4. Gestión Financiera y de Personal (CU-07, 08, 09) - (Solo Dueño) . . . . .	9
<b>6. Hitos y Corrección de Errores</b>	<b>10</b>
6.1. Errores Técnicos Corregidos . . . . .	10
6.2. Evolución Funcional (Mejoras V1.0) . . . . .	10
6.3. Mejoras de Interfaz (UI/UX) . . . . .	11
<b>7. Evidencias (Capturas de Pantalla)</b>	<b>12</b>
<b>8. Conclusión</b>	<b>19</b>

# 1. Introducción al Proyecto

Este documento detalla el desarrollo de la Versión 1.0 del sistema web "Flor y Hojaldra". El objetivo principal de esta versión es crear una plataforma funcional que permita la gestión de un negocio de venta de pan de muerto y flores, cubriendo tanto la interfaz pública para clientes como un panel de administración robusto para la gestión interna.

## 1.1. Objetivos de la Versión 1.0

- Implementar un catálogo de productos público.
- Permitir a los clientes levantar pedidos en línea (pago en tienda).
- Desarrollar un panel de administración seguro con roles (Dueño y Empleado).
- Gestionar el ciclo de vida completo de productos, empleados, gastos y pedidos.
- Proveer un Punto de Venta (POS) para ventas en mostrador.
- Generar reportes financieros básicos (Ingresos vs. Gastos).
- Asegurar un diseño responsivo tanto en la tienda pública como en el panel de administración.

## 1.2. Stack Tecnológico

El proyecto se desarrolló enteramente con tecnologías del lado del servidor y del cliente, utilizando un stack LAMP clásico sobre XAMPP.

- **Backend:** PHP 8.2
- **Base de Datos:** MySQL (gestionado vía MariaDB en XAMPP)
- **Servidor Local:** Apache (vía XAMPP)
- **Frontend:** HTML5, CSS3 (con Media Queries) y JavaScript (Vanilla JS).
- **Formato de Documentación:** L<sup>A</sup>T<sub>E</sub>X

# 2. Análisis y Diseño

La base del proyecto se fundamenta en 9 Casos de Uso (CU) principales, que definen el alcance funcional del sistema.

## 2.1. Actores del Sistema

- **Cliente (Visitante):** Usuario anónimo que navega por el catálogo y puede realizar un pedido.

- **Empleado:** Usuario autenticado. Gestiona pedidos y registra ventas (POS), pero no tiene acceso a módulos financieros o de configuración.
- **Dueño (Administrador):** Super-usuario con acceso total a todos los módulos, incluyendo reportes, gestión de empleados y gastos.

## 2.2. Casos de Uso (CU) V1.0

- CU-1: **Ver Catálogo de Productos:** El cliente ve los productos disponibles.
- CU-2: **Empleado Registra Pedido Manual:** El empleado registra un pedido tomado por teléfono o en persona.
- CU-3: **Cliente Levanta Pedido en Línea:** El cliente arma su carrito y envía un pedido.
- CU-4: **Gestionar Pedidos Pendientes:** El empleado actualiza el estado de un pedido (Nuevo → En Preparación → Listo).
- CU-5: **Registrar Venta en Persona (POS):** El empleado registra una venta de mostrador (se completa al instante).
- CU-6: **Gestionar Inventario (CRUD):** El dueño crea, edita y elimina productos (incluyendo imágenes).
- CU-7: **Registrar Gasto Operativo:** El dueño registra las salidas de dinero.
- CU-8: **Consultar Reporte de Finanzas:** El dueño ve el Dashboard con Ingresos vs. Gastos.
- CU-9: **Gestionar Cuentas de Empleados (CRUD):** El dueño crea y elimina cuentas de usuario.

## 3. Diseño de la Base de Datos

La base de datos floryhojaldra se compone de 6 tablas principales para evitar redundancia y asegurar la integridad de los datos.

- **Categorias:** Almacena los tipos de producto (ej. "Flores", "Pan de Muerto").
- **Productos:** El inventario. Contiene precio, stock, y la ruta de la imagen.
- **Usuarios:** Almacena las cuentas de Dueño y Empleado con sus contraseñas hasheadas.
- **Gastos:** Registro de todos los egresos del negocio.
- **Pedidos:** Tabla maestra que almacena la información del cliente, tipo de pedido, estado (incluyendo 'Cancelado') y total.
- **Detalle\_Pedidos:** Tabla de unión que conecta un Pedido con múltiples Productos.

### 3.1. Archivo de Conexión (config/db\_connect.php)

Este es el archivo central que gestiona la conexión con MySQL.

- **Propósito:** Establece una conexión única con la base de datos usando `mysqli`.
- **Función Clave:** Define las 4 variables de conexión (servidor, usuario, contraseña, bd).
- **Seguridad:** Establece el charset a `utf8mb4` para manejar acentos y caracteres especiales correctamente.

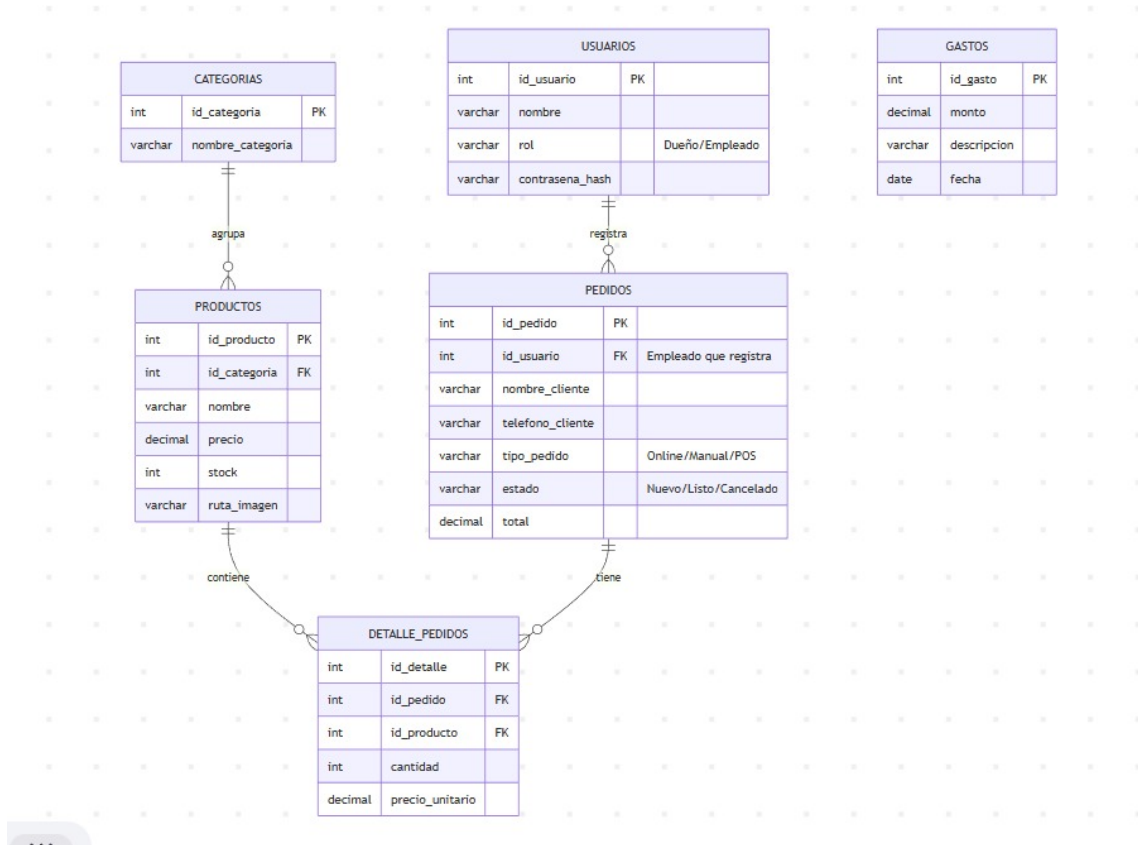


Figura 1: Diagrama Entidad-Relación de la Base de Datos V1.0.

## 4. Estructura de Archivos del Proyecto

La estructura se diseñó para separar claramente el frontend público (raíz) del backend privado (carpeta /admin).

```
/floryhojaldra
|
|-- /admin                                (Panel de Administración - Backend)
|   |-- /css
|   |   |-- admin_styles.css
|   |-- /js
|   |   |-- admin_scripts.js
|   |-- crear_empleado.php
|   |-- crear_gasto.php
|   |-- crear_pedido_manual.php
|   |-- crear_producto.php
|   |-- editar_empleado.php
|   |-- editar_gasto.php
|   |-- editar_producto.php
|   |-- eliminar_empleado.php
|   |-- eliminar_gasto.php
|   |-- eliminar_producto.php
|   |-- gestionar_empleados.php
|   |-- gestionar_gastos.php
|   |-- gestionar_pedidos.php
|   |-- gestionar_productos.php
|   |-- index.php                        (Dashboard de Reportes)
|   |-- login.php
|   |-- logout.php
|   |-- pos_venta.php
|   |-- proteger.php
|   |-- reporte_ventas.php
|   |-- sidebar.php
|   \-- ver_pedido.php
|
|-- /config
|   |-- db_connect.php
|
|-- /css
|   |-- store_styles.css
|
|-- /img
|   |-- placeholder.png
```

```
|  
|-- /uploads  
|   |-- /products  
|       |-- (Aquí se guardan las imágenes de productos)  
|  
|-- index.php                (Tienda Pública - Frontend)  
\-- pedido_exitoso.php
```

## 5. Descripción Detallada de Archivos PHP

A continuación, se describe el propósito ("para qué sirve") cada archivo PHP clave del proyecto.

### 5.1. Módulo de Tienda (Frontend - Raíz)

- **index.php (Raíz):**
  - **CU-01:** Lee la tabla `Productos` (junto con `Categorias`) y los muestra al cliente en un catálogo. Muestra el stock y la imagen.
  - **CU-03:** Contiene el formulario del carrito (manejado con JavaScript) y la lógica POST para registrar un pedido de tipo `.online`. Realiza la validación de 10 dígitos del teléfono.
- **pedido\_exitoso.php:** Página de confirmación a la que es redirigido el cliente tras completar un pedido (CU-03). Muestra el ID del pedido.

### 5.2. Módulo de Administración (Backend - /admin)

#### 5.2.1. Autenticación y Seguridad

- **login.php:** Muestra el formulario de inicio de sesión. Valida al usuario contra la tabla `Usuarios` usando `password_verify()`. Inicia la sesión `$_SESSION` y redirige según el rol (Dueño al Dashboard, Empleado a Pedidos). Incluye un enlace para regresar al catálogo.
- **proteger.php:** Script de seguridad que se incluye al inicio de *todas* las páginas del admin. Revisa si `$_SESSION['id_usuario']` existe; si no, redirige a `login.php`.
- **logout.php:** Destruye la sesión `$_SESSION` y redirige a `login.php`.
- **sidebar.php:** El menú de navegación lateral reutilizable. Contiene lógica PHP para mostrar/ocultar enlaces (ej. "Dashboard", "Gastos") basándose en el `$_SESSION['rol']`.

#### 5.2.2. Gestión de Productos (CU-06)

- **gestionar\_productos.php:** (CRUD - Read) Muestra la tabla de todos los productos, su stock, e imagen (o un botón para añadirla). Los enlaces de `.Editar/Eliminar` solo se muestran si el rol es "Dueño".
- **crear\_producto.php:** (CRUD - Create) Formulario para añadir un nuevo producto. Incluye la lógica de subida de archivos (`$_FILES`), validación de tipo/tamaño, y guardado de la imagen en `/uploads/products`. Solo accesible por "Dueño".



- **editar\_producto.php:** (CRUD - Update) Formulario para editar un producto. Muestra la imagen actual y permite subir una nueva (borrando la antigua del servidor). Solo accesible por "Dueño".
- **eliminar\_producto.php:** (CRUD - Delete) Script que procesa el borrado de un producto (requiere confirmación JS). Solo accesible por "Dueño".

### 5.2.3. Gestión de Pedidos y Ventas (CU-02, 04, 05)

- **gestionar\_pedidos.php:** (CU-04) El panel de control principal para Empleados. Muestra una lista de pedidos **pendientes** (estados 'Nuevo', 'En Preparacion', 'Listo para Recoger').
- **crear\_pedido\_manual.php:** (CU-02) Formulario para que el Empleado registre un pedido manualmente. Incluye JavaScript para el carrito y lógica PHP de transacción para insertar en Pedidos, Detalle\_Pedidos y descontar stock de Productos.
- **pos\_venta.php:** (CU-05) Punto de Venta para ventas de mostrador. Similar a la creación manual, pero guarda el pedido directamente como 'Completado y Pagado' y tipo 'POS'.
- **ver\_pedido.php:** (CU-04) Muestra el detalle completo de un pedido. Permite al empleado usar formularios para actualizar el estado del pedido (ej. 'Nuevo' a 'En Preparacion') o para "Cancelar Pedido"(lo cual restaura el stock).

### 5.2.4. Gestión Financiera y de Personal (CU-07, 08, 09) - (Solo Dueño)

- **index.php (Admin):** (CU-08) El Dashboard principal. Muestra KPIs (Ingresos, Gastos, Ganancia) basados en un filtro de fechas, y alertas (Pedidos Pendientes).
- **reporte\_ventas.php:** (CU-08b) Muestra una tabla de pedidos **completados** en un rango de fechas. Muestra qué empleado registró las ventas 'POS'.
- **gestionar\_gastos.php:** (CRUD - Read) Muestra la tabla de gastos registrados.
- **crear\_gasto.php:** (CRUD - Create) Formulario para registrar un nuevo gasto.
- **editar\_gasto.php:** (CRUD - Update) Formulario para editar un gasto.
- **eliminar\_gasto.php:** (CRUD - Delete) Script para eliminar un gasto.
- **gestionar\_empleados.php:** (CRUD - Read) Muestra la tabla de usuarios.
- **crear\_empleado.php:** (CRUD - Create) Formulario para crear un nuevo usuario. Incluye la lógica `password_hash()` para la contraseña.
- **editar\_empleado.php:** (CRUD - Update) Formulario para editar un usuario. Incluye lógica para actualizar la contraseña solo si se proporciona una nueva.
- **eliminar\_empleado.php:** (CRUD - Delete) Script para eliminar un usuario.

## 6. Hitos y Corrección de Errores

Durante el desarrollo de la V1.0, se presentaron varios desafíos y se implementaron mejoras iterativas.

### 6.1. Errores Técnicos Corregidos

- **Error 404 (Rutas):** Se presentaron múltiples errores 404.
  - **Caso 1: crear\_producto.php.** Un error tipográfico en el nombre del archivo (producto vs producto) impedía que el enlace funcionara.
  - **Caso 2: admin\_scripts.js.** El menú móvil no funcionaba porque el archivo .js no se estaba cargando. La causa fue que se omitió crear la carpeta /admin/js/.
- **Error de Lógica PHP (bind\_param):**
  - **Caso 1: crear\_producto.php.** Se produjo un `ArgumentCountError`. La cadena de tipos ("ssdiiss") tenía 7 caracteres, pero solo se estaban pasando 6 variables. Se corrigió a "ssdiis".
  - **Caso 2: pos\_venta.php.** Se produjo un error fatal al intentar leer un mensaje de excepción. Se usó la sintaxis incorrecta `$e.getMessage()` (concatenación) en lugar de la sintaxis correcta de objeto `$e->getMessage()`.
- **Error de Renderizado (Texto Fantasma):** En el panel de administración aparecía el texto "de PHP para evitar espacios...". Esto se debió a que dicho comentario en `db_connect.php` estaba fuera de las etiquetas `<?php ... ?>`, causando que se imprimiera como HTML. Se solucionó borrando el cierre `?>` del archivo.

### 6.2. Evolución Funcional (Mejoras V1.0)

El alcance inicial se expandió para incluir funcionalidades críticas:

- **Implementación de Roles:** Se añadió la lógica de `$_SESSION['rol']` para diferenciar entre "Dueños .<sup>E</sup>mpleado", ocultando menús (en `sidebar.php`) y protegiendo el acceso directo a páginas (en `index.php`, `gestionar_gastos.php`, etc.).
- **Subida de Imágenes:** Se modificó el CRUD de Productos para incluir `enctype="multipart/form-d` en los formularios y la lógica PHP (`$_FILES`, `move_uploaded_file`) para manejar la subida, guardado y borrado de imágenes.
- **Cancelación de Pedidos:** Se añadió el estado 'Cancelado' a la BD y se implementó la lógica en `ver_pedido.php` para no solo cancelar, sino también **restaurar el stock** de los productos (usando transacciones MySQL).
- **Reporte de Ventas Detallado:** Se creó `reporte_ventas.php` para ver el historial de ventas completadas, ya que `gestionar_pedidos.php` solo muestra pendientes.

### 6.3. Mejoras de Interfaz (UI/UX)

- **Diseño Responsivo (Admin):** Se implementó un menú lateral colapsable (hamburguesa) usando CSS Media Queries y JavaScript (`admin_scripts.js`) para que el panel sea 100 % funcional en dispositivos móviles.
- **Diseño Responsivo (Tienda):** Se usó CSS Flexbox y Media Queries en `store_styles.css` para que el catálogo y el carrito se apilen verticalmente en móviles.
- **Mejoras de Usabilidad:** Se añadieron enlaces como "Regresar al Catálogo" (en `login.php`) y "Añadir Imagen" (en `gestionar_productos.php`) para mejorar el flujo del usuario.

## 7. Evidencias (Capturas de Pantalla)



The image shows a login form for 'Flor y Hojaldra'. The form is centered on a light gray background. It has a title 'Flor y Hojaldra' in a large, bold, dark brown font, followed by the subtitle 'Acceso al Panel' in a smaller, gray font. Below the subtitle, there are two input fields: one for 'Usuario:' and one for 'Contraseña:'. Both fields are white with a thin gray border. Below the password field is a large, orange button with the text 'Entrar' in white. At the bottom of the form, there is a link that says '← Regresar al Catálogo' in a gray font.

Figura 2: Tienda pública en vista de escritorio.

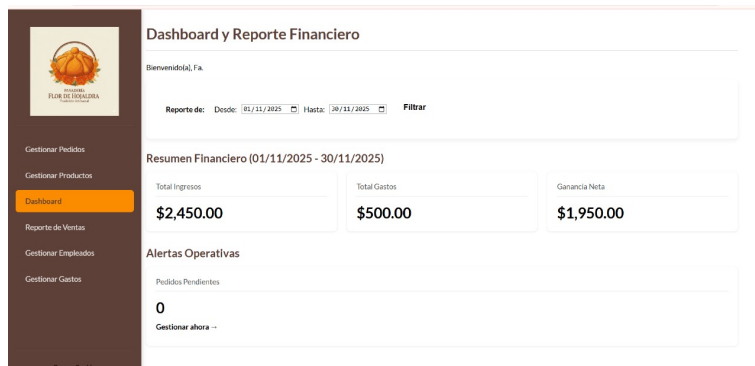
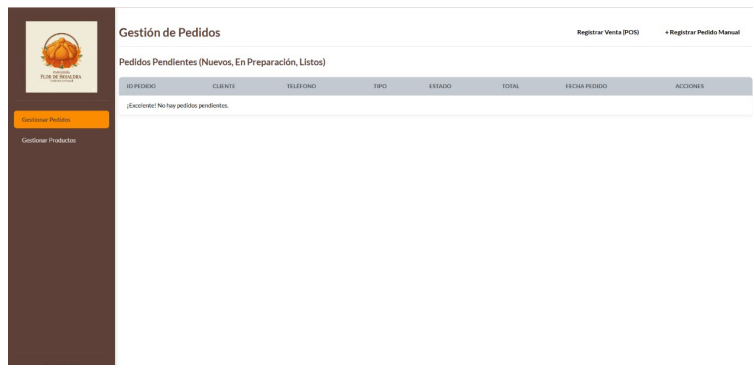


Figura 3: Dashboard principal del administrador (Rol Dueño).



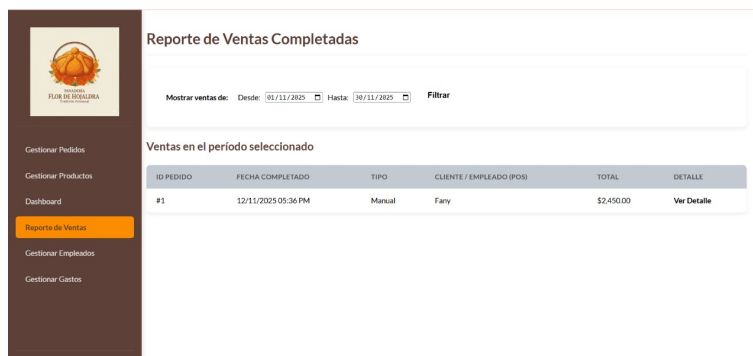
Productos Actuales							
ID	IMAGEN	NOMBRE DEL PRODUCTO	CATEGORIA	PRECIO	STOCK	ESTADO	ACCIONES
1		Cempasúchil	Flores de muerto	\$50.00	1	Activo	Ver

Figura 4: Módulo de gestión de inventario.



Pedidos Pendientes (Nuevos, En Preparación, Listos)							
ID PEDIDO	CLIENTE	TELEFONO	TIPO	ESTADO	TOTAL	FECHA PEDIDO	ACCIONES
(Excluído): No hay pedidos pendientes.							

Figura 5: Módulo de gestión de pedidos pendientes.



Mostrar ventas de: Desde: 01/11/2025 Hasta: 30/11/2025 Filtrar					
Ventas en el periodo seleccionado					
ID PEDIDO	FECHA COMPLETADO	TIPO	CLIENTE / EMPLEADO (POS)	TOTAL	DETALLE
#1	12/11/2025 05:36 PM	Manual	Fany	\$2,450.00	<a href="#">Ver Detalle</a>

Figura 6: Detalle y actualización de estado de un pedido.

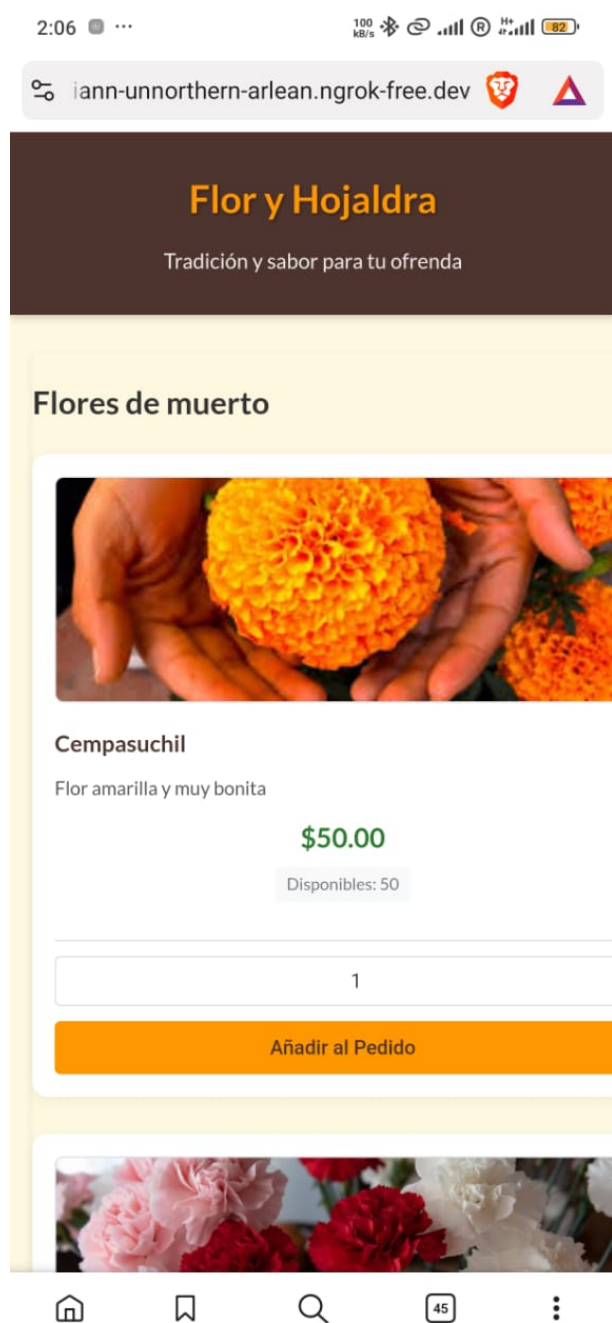


Figura 7: Vista como cliente en versión móvil.

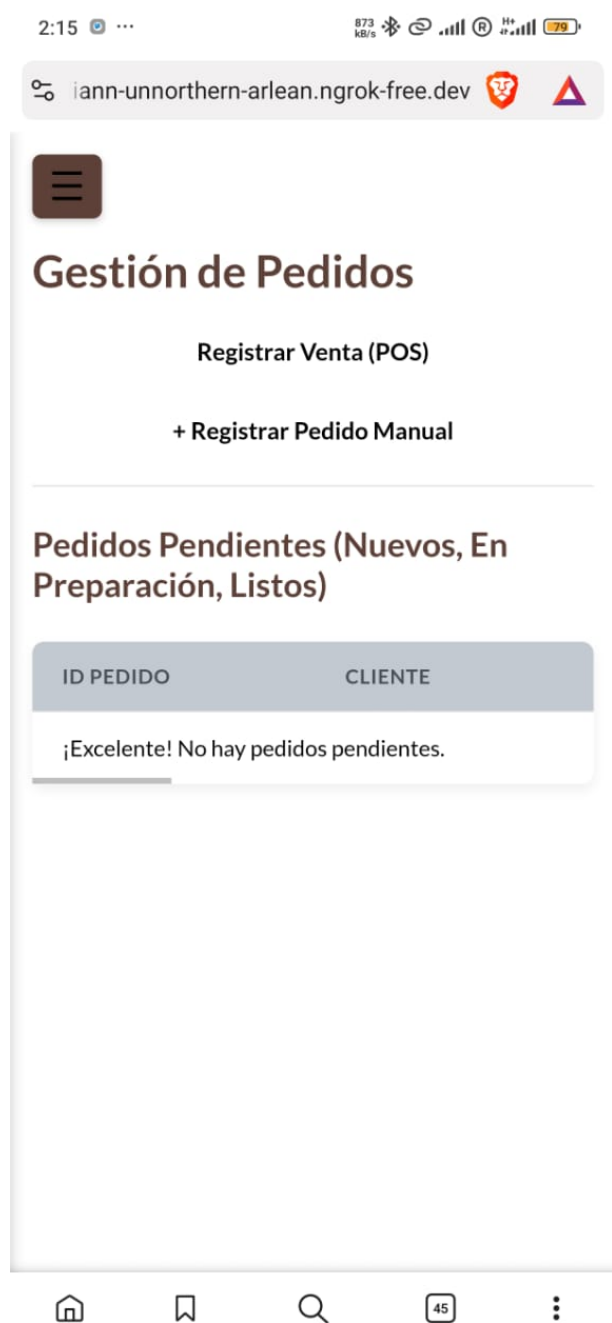


Figura 8: Vista móvil como administrador para gestión de pedidos.

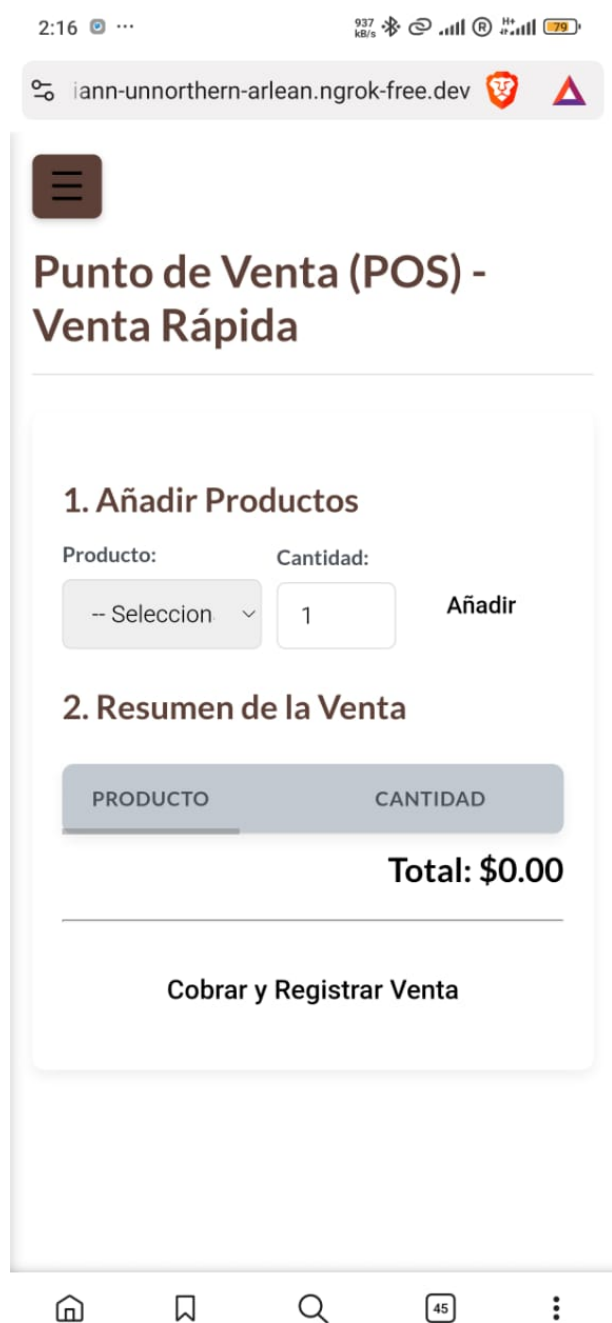


Figura 9: Vista móvil como administrador para registro de venta





Figura 10: Vista móvil de las opciones de gestión en el panel.



Figura 11: Módulo de gestión de empleados.

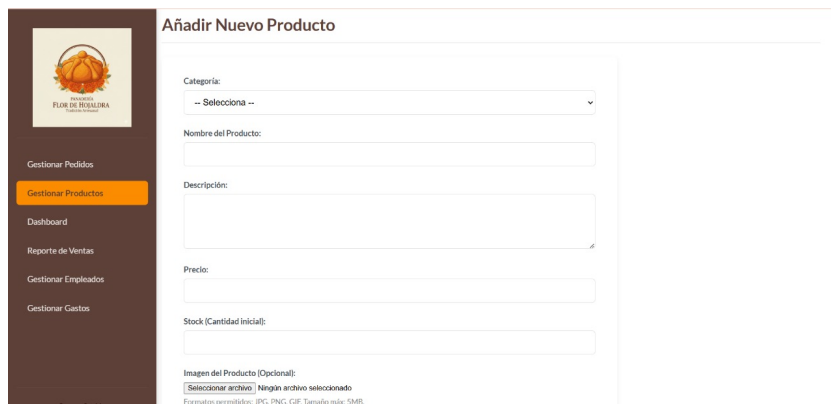


Figura 12: Módulo para agregar un nuevo producto.

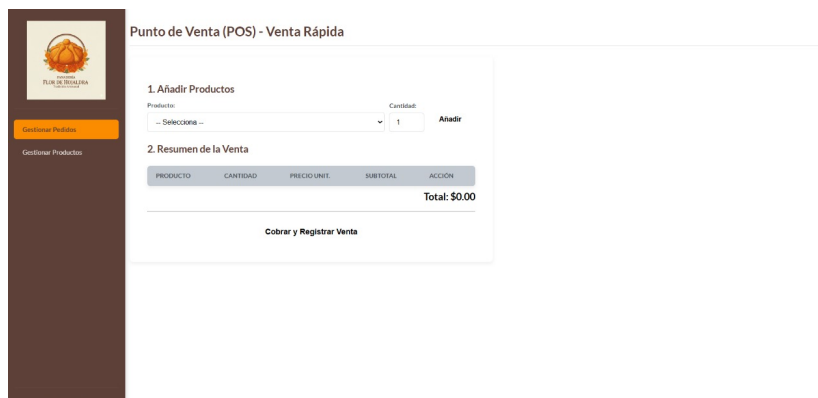


Figura 13: Módulo de punto de venta.

Registrar Pedido Manual (CU-02)

1. Datos del Cliente

Nombre:

Telefono (10 digitos):

2. Añadir Productos

Producto:  Cantidad:

3. Resumen del Pedido

PRODUCTO	CANTIDAD	PRECIO UNIT.	SUBTOTAL	ACCION
----------	----------	--------------	----------	--------

Total: \$0.00

Figura 14: Módulo para el registro manual de un pedido.

## 8. Conclusión

La Versión 1.0 del proyecto "Flor y Hojaldra" se ha completado exitosamente. Se ha construido un sistema web funcional, seguro y responsivo que cumple con todos los requisitos funcionales establecidos. La plataforma permite una gestión integral del negocio, desde el inventario hasta las finanzas, y proporciona una interfaz de cliente clara para realizar pedidos.

El desarrollo iterativo permitió identificar y corregir errores, así como añadir funcionalidades críticas (gestión de roles, cancelación de pedidos) que no se habían contemplado en el diseño inicial, resultando en un producto final mucho más robusto.

El sistema está listo para la siguiente fase: el despliegue en un servidor de hosting público y la planificación de la Versión 2.0 (pedidos programados, cuentas de cliente).