

## **Trabajo Integrador**

### **Sistema para Floreria “Arteplantas”**

**CARRERA:** Ingeniería en Sistemas de Información

**MATERIA:** Paradigmas y Lenguajes de Programación III

**COMISIÓN:** “U” (única)

**PROFESOR:** Mgter. Ing. Encina Agustín.

**ESTUDIANTE:** Brycki Angela Selene

**FECHA:** Noviembre 2025



---

## Índice

Trabajo Integrador: Sistema para Florería.....	3
Descripción general del proyecto.....	3
Objetivos generales.....	3
Objetivos específicos.....	3
Requisitos.....	4
Requerimientos funcionales (RF).....	4
Requerimientos no funcionales (RNF).....	4
Análisis y diseño del sistema.....	4
Actores del sistema.....	4
Casos de uso principales.....	4
Estructura del Proyecto.....	5
Desarrollo e Implementación del Sistema.....	5
Funcionalidades Implementadas en el Módulo Cliente.....	5
Funcionalidades Implementadas en el Módulo Administrador.....	7
Tecnologías y herramientas.....	8
Aportes y uso de tecnologías.....	8
Plugin jQuery utilizado.....	9
Owl Carousel 2.....	9
Link del repositorio.....	9



---

## **Trabajo Integrador: Sistema para Floreria**

Nombre del Proyecto → **Arteplantas 360**

### **Descripción general del proyecto**

**Arteplantas360** es una aplicación web creada para la gestión integral de una florería, permitiendo administrar inventario, productos y ventas mientras ofrece a los clientes una experiencia de compra ágil, moderna e intuitiva.

El sistema está compuesto por dos grandes módulos:

- **Módulo Cliente** → muestra catálogo, fichas de productos, carrito y formulario de compra.
- **Módulo Administrador** → permite gestionar productos, stock, imágenes y visibilidad mediante una interfaz protegida con login.

Toda la información se administra desde una base de datos MySQL, y se opera mediante PHP + JavaScript consumiendo APIs internas.

### **Objetivos generales**

- Facilitar la administración del inventario, precios y ventas de la florería.
- Mejorar la eficiencia en la gestión de stock.
- Proveer una experiencia de cliente moderna y accesible desde la web.

### **Objetivos específicos**

- 1) Registrar, modificar y eliminar productos con información completa (nombre, foto, descripción, stock, precio, categoría).
- 2) Actualizar automáticamente el stock al recibir mercadería o al restar stock.
- 3) Gestionar los precios de los productos.



- 
- 4) Permitir la realización de pedidos mediante formulario de compra.

### Requisitos

#### ***Requerimientos funcionales (RF)***

- **RF01:** El sistema debe permitir agregar, modificar y eliminar productos.
- **RF02:** El sistema debe permitir actualizar el stock automáticamente.
- **RF03:** El sistema debe permitir búsquedas.
- **RF04:** Mostrar catálogo online mediante galería y fichas de productos.
- **RF05:** Permitir al cliente realizar pedidos con formulario de compra.
- **RF06:** Acceso mediante login al módulo de gestión.

#### ***Requerimientos no funcionales (RNF)***

- **RNF01:** El sistema debe contar con una interfaz intuitiva y responsive.
- **RNF02:** El sistema debe permitir el acceso mediante usuario y contraseña (Administrador).
- **RNF03:** Respaldo automático de la base de datos.

### Análisis y diseño del sistema

#### ***Actores del sistema***

- **Cliente:** consulta catálogo, agrega productos al carrito y realiza pedidos.
- **Administrador:** gestiona productos, stock, precios e imágenes.

#### ***Casos de uso principales***

- UC01: Iniciar sesión (Administrador)



- UC02: Agregar producto
- UC03: Editar producto
- UC04: Eliminar producto
- UC05: Visualizar catálogo (Cliente)
- UC06: Visualizar ficha individual
- UC07: Agregar producto al carrito
- UC08: Vaciar o modificar carrito
- UC09: Realizar pedido

### Estructura del Proyecto

```
arteplantas360/  
├── config/  
│   └── db.php                # Conexión a MySQL  
├── api/  
│   ├── productos.php        # CRUD productos (pública)  
│   ├── admin-productos.php  # CRUD productos (admin)  
│   └── auth.php              # Login y autenticación  
├── css/  
│   ├── styles.css           # Estilos principales  
│   └── admin.css             # Estilos panel admin  
├── js/  
│   ├── main.js               # Funciones principales  
│   ├── admin.js              # Funciones admin  
│   └── productos-page.js     # Carga de fichas  
├── scripts/  
│   └── setup.sql             # Script BD  
└── [archivos HTML y assets]
```

### Desarrollo e Implementación del Sistema

#### ***Funcionalidades Implementadas en el Módulo Cliente***



---

## **Catálogo dinámico de productos**

- Los productos se cargan desde MySQL a través de la API `api/productos.php`.
- `main.js` procesa la respuesta JSON y renderiza las tarjetas de productos.
- En `productos.html` se utilizan plantillas dinámicas para mostrar fichas individuales.

## **Galería principal (Index)**

- Implementación del plugin Owl Carousel 2 mediante jQuery.
- Carga automática de imágenes almacenadas en la BD.
- Animaciones, transición táctil y diseño responsive.

## **Carrito de compras**

- Lógica desarrollada completamente en JavaScript:
  - Agregar productos al carrito.
  - Eliminar ítems.
  - Vaciar carrito.
  - Cálculo de subtotal, total e ítems.
- Contador dinámico sobre el ícono del carrito (actualizado en tiempo real).

## **Formulario de compra**

- Envío de información del pedido.
- Validaciones realizadas mediante JS.

## **FAQ interactivo**

- Sistema de acordeón desarrollado con JavaScript:



- Expande/contrae cada pregunta.
- Íconos (+ / -) que cambian según el estado.

### ***Funcionalidades Implementadas en el Módulo Administrador***

#### **Login de administrador**

- Formulario en admin-login.html.
- Autenticación en api/auth.php con creación de sesión.
- Redirección automática al panel de administración.
- Validación de credenciales (usuario: admin).

#### **Gestión completa de productos**

Implementada mediante admin.js + api/admin-productos.php:

- Crear productos:
  - Nombre, descripción, cantidad, precio, categoría, visibilidad e imagen.
- Editar productos:
  - Modales dinámicos que cargan información existente.
- Eliminar productos:
  - Confirmación previa y actualización automática.
- Sincronización instantánea con el catálogo del cliente.

#### **Administración de imágenes**

- Soporte para imágenes grandes mediante migración SQL (migrate-images.sql).
- Las imágenes se guardan en base64 en la BD.



## Tecnologías y herramientas

Área	Tecnología
IDE	Visual Studio Code
Repositorio	GitHub
Base de Datos	MySQL
Frontend	HTML, CSS, JavaScript
Backend	PHP
APIs	REST internas (productos / admin / auth)

## Aportes y uso de tecnologías

### HTML

- Estructura de todas las vistas principales del sitio.
- Maquetación de galerías, fichas de producto, formularios y panel admin.

### CSS

- Diseño responsivo con `styles.css`.
- Estilos personalizados del panel administrador en `admin.css`.
- Adaptación para dispositivos móviles.

### JavaScript

- Consumo de APIs con `fetch()`.
- Render dinámico de productos.
- Carrito de compras completo.
- Panel administrativo totalmente interactivo.
- Validaciones y animaciones.



- Implementación de Owl Carousel 2 usando jQuery.

## PHP

- APIs REST internas:
  - `productos.php` → listado público de productos.
  - `admin-productos.php` → CRUD completo (admin).
  - `auth.php` → manejo de sesiones.
- Conexión a MySQL mediante `config/db.php`.

## MySQL

- Diseño de tablas
- Scripts de instalación (`setup.sql`).
- Migraciones para soportar imágenes.

## GitHub

- Control de versiones del proyecto.
- Estructura limpia con carpetas separadas.

## Plugin jQuery utilizado

### ***Owl Carousel 2***

Utilizado en `index.html` para cargar de forma dinámica el carrusel principal con imágenes provenientes de la base de datos. Permite transiciones animadas, navegación táctil y adaptación responsive.

## Link del repositorio

<https://github.com/selenebrycki/artepplantas360>