

Trabajo Práctico 2

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: 07-10-2025



Índice

Trabajo Práctico 2: Sistema para Floreria.....	3
Breve descripción de la idea.....	3
Objetivos generales.....	3
Objetivos específicos.....	3
Requisitos.....	4
Requerimientos funcionales (RF).....	4
Requerimientos no funcionales (RNF).....	5
Análisis y Diseño del Sistema.....	5
Actores del sistema.....	5
Casos de uso.....	5
Implementaciones actuales.....	6
Mejoras Implementadas con JS.....	7
Tecnologías y herramientas.....	7
Próximos pasos.....	8
AA19 - Plugin jQuery.....	8
Owl Carousel 2.....	8



Trabajo Práctico 2: Sistema para Florería

Nombre del Proyecto → **Arteplantas 360**

Breve descripción de la idea

Arteplantas 360 será una aplicación web integral para administrar una florería y ofrecer una experiencia de compra online a los clientes. Permitirá:

- Gestionar inventario, precios, ventas y reportes internos (módulo de gestión).
- Mostrar un catálogo de productos en la web mediante una galería, fichas individuales y formulario de compra (módulo cliente).
- Optimizar tiempos de gestión, reducir pérdidas por desabastecimiento y facilitar la toma de decisiones comerciales.

Objetivos generales

- Facilitar la administración del inventario, precios y ventas de la florería.
- Mejorar la eficiencia en la gestión de stock y ventas mediante un sistema automatizado.
- Proveer información confiable y actualizada para la toma de decisiones.
- 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 registrar ventas o recibir mercadería.
- 3) Generar alertas automáticas cuando un producto esté próximo a agotarse.
- 4) Gestionar los precios de los productos y mantener un historial de cambios. Permitir la modificación masiva y por unidad.



- 5) Registrar ventas y permitir filtrarlas por fecha, producto o cliente.
- 6) Generar reportes sobre productos más vendidos y evolución de stock y ventas.
- 7) Integrar códigos de barras para registro rápido de productos y ventas (sujeto a modificación).
- 8) 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 según ventas o recepción de mercadería.
- **RF03:** El sistema debe generar alertas cuando el stock de un producto esté por debajo de un mínimo definido.
- **RF04:** El sistema debe registrar ventas y permitir filtrarlas por fecha, producto o cliente.
- **RF05:** El sistema debe generar reportes de ventas y evolución de stock.
- **RF06:** El sistema debe mantener un historial de cambios de precios.
- **RF07:** El sistema debe permitir registrar productos mediante un lector de código de barras (sujeto a modificación).
- **RF08:** El sistema debe permitir búsquedas y ventas rápidas a través del código de barras (sujeto a modificación).
- **RF09:** El sistema debe permitir generar e imprimir etiquetas con código de barras para los productos (sujeto a modificación).
- **RF10:** Mostrar catálogo online mediante galería y fichas de productos.



- **RF11:** Permitir al cliente realizar pedidos con formulario de compra.
- **RF12:** Acceso mediante login y diferenciación de roles (administrador y empleado).

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, diferenciando permisos por rol (Administrador / Empleado).
- **RNF03:** Respaldo automático de la base de datos.

Análisis y Diseño del Sistema

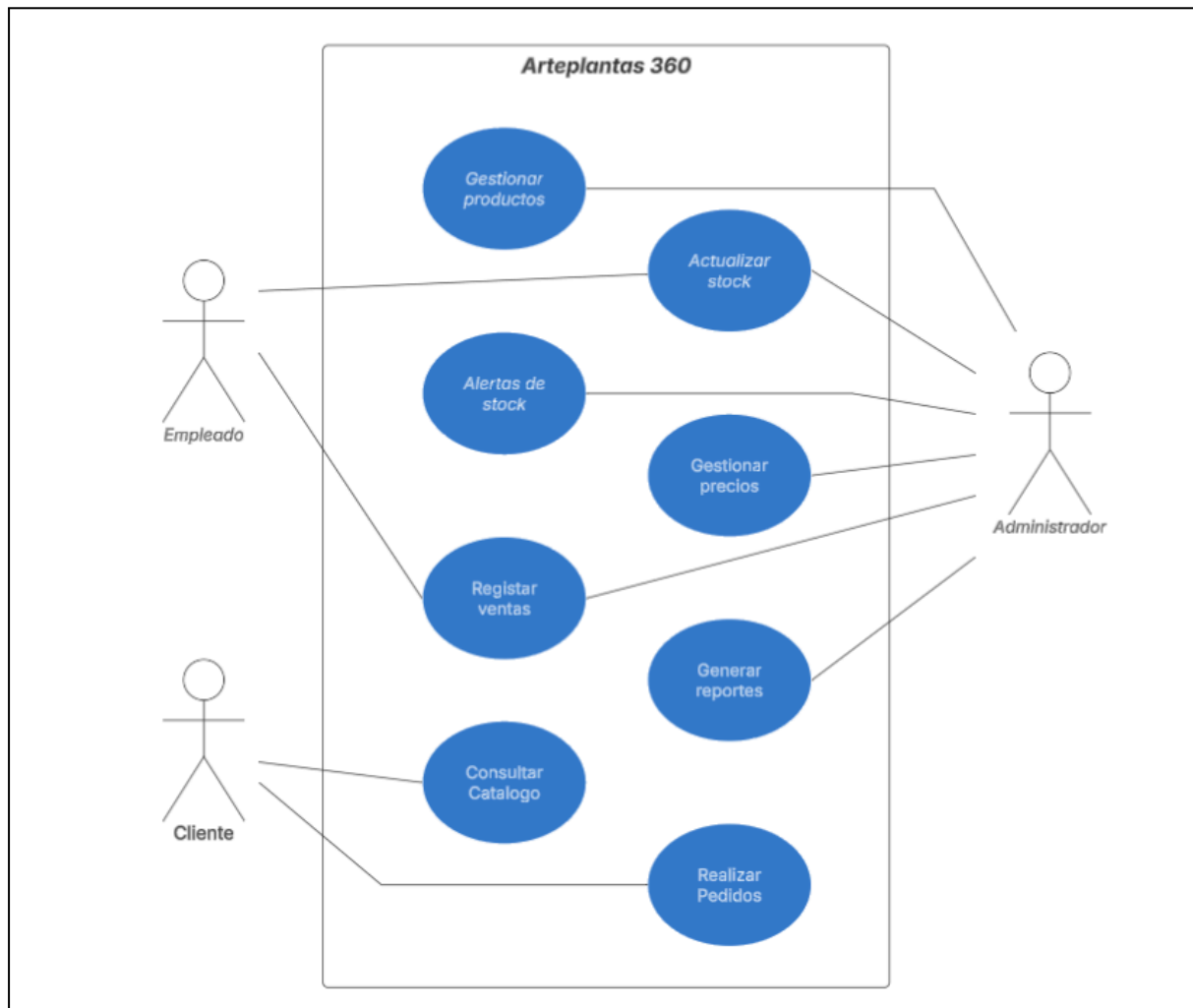
Actores del sistema

- **Administrador:** Gestiona inventario, precios, reportes y ventas.
- **Empleado:** Registra ventas y actualiza stock, sin modificar precios ni acceder a reportes.
- **Cliente:** Consulta catálogo, ve fichas de productos y realiza pedidos.

Casos de uso

- 1) **Gestionar productos:** Agregar, modificar, eliminar productos.
- 2) **Actualizar stock:** Restar stock al vender o sumar al recibir mercadería.
- 3) **Alertas de stock:** Notificar cuando un producto está por agotarse.
- 4) **Gestionar precios:** Modificar precios y consultar historial.
- 5) **Registrar ventas**
- 6) **Generar reportes:** Listado de productos más vendidos, evolución de stock y ventas.
- 7) **Consultar catálogo** online.

8) Realizar pedidos con formulario de compra.



Implementaciones actuales

- **Index:** Galería de imágenes estáticas, datos de contacto.
- **Menú lateral:** Secciones “Inicio”, “Productos” (en boxes). Secciones “Quienes somos” y “Preguntas frecuentes”.
- **Fichas individuales de productos y formulario** de compra para realizar pedidos.
- **Login:** Acceso con usuario y contraseña para administradores (usuario: *admin*; contraseña: *arteplantas1988*)



- **Gestión de productos:**

- Agregar productos (nombre, cantidad, precio, foto, categoría, visible/no visible).
- Editar productos.
- Eliminar productos.

Mejoras Implementadas con JS

- Sistema de Carrito de Compras →
 - Implementación de contador que muestra la cantidad total de productos en el ícono del carrito
 - Funcionalidad para agregar, eliminar y vaciar productos del carrito
 - Cálculo automático de totales y subtotales
- Acordeón de Preguntas Frecuentes
 - Sistema que permite expandir y contraer respuestas al hacer clic
 - Indicadores visuales (+/-) que cambian según el estado de cada pregunta
- Panel de Administración de Productos
 - Gestión completa de productos (crear, editar, eliminar)
 - Tabla interactiva que muestra todos los productos disponibles
 - Sincronización automática entre el panel admin y la tienda

Tecnologías y herramientas

- IDE: Visual Studio Code
- Repositorio: GitHub
- Hosting: GitHub Pages
- Base de datos: Por definir (MySQL / PostgreSQL)



- Frontend: HTML, CSS, JS
- Backend: Por definir (PHP / Node.js / Python)

Próximos pasos

- Completar funcionalidades de gestión interna: registrar ventas, generar reportes.
- Implementar galería dinámica.
- Completar diagramas de diseño.

[Link del repositorio](#)

[Link del proyecto en línea](#)

AA19 - Plugin jQuery

Owl Carousel 2

- **Creación de la rama:** Primero, desde la terminal creé una nueva rama en mi repositorio con el comando: `git checkout -b plugin`

Esto me permitió trabajar en una versión separada del proyecto principal para implementar el plugin sin afectar el resto del código.
- **Búsqueda y selección del plugin:** Busqué diferentes opciones de plugins de JavaScript y jQuery, y elegí **Owl Carousel 2**, ya que es fácil de implementar, responsivo y permite crear un carrusel de imágenes personalizable. Además ya tenía previamente la idea de que la **galería** de la página sea **dinámica**.
- **Integración en el proyecto:** Agregué los enlaces a las librerías en el head de mi archivo `index.html`.



- **Estructura del carrusel:** Agregué un bloque HTML en la parte principal del sitio con las imágenes para la galería.
- **Inicialización y configuración del plugin:** Antes de cerrar el body, incluí código js para activar el carrusel con 3 imágenes visibles y efecto automático.
- **Ajustes de estilo:** En el archivo CSS personalice la apariencia para que el carrusel ocupe casi todo el ancho, tenga espacio arriba y a los costados, y las imágenes con bordes redondeados.
- **Pruebas y resultados:** Abrí la página en el navegador y verifiqué que el carrusel funcione correctamente:
 - En PC se muestran 3 imágenes.
 - En tablets, 2 imágenes.
 - En celulares, 1 imagen.

Además, las imágenes cambian automáticamente cada 3 segundos y se adaptan al tamaño de la pantalla.