

Sistema de Carrito de Compras

Módulo: 09_Carrito

1 09_Carrito

1.1 Descripción General

El módulo **09_Carrito** implementa un sistema web de **Carrito de Compras** desarrollado en Python utilizando el framework **Flask**, conectado a una base de datos **MySQL**. El sistema permite administrar tres componentes principales mediante programación orientada a objetos:

- **Productos:** gestión CRUD de productos, incluyendo carga de imágenes.
- **Clientes:** gestión CRUD de clientes.
- **Pedidos (Carritos):** creación de pedidos y administración de productos dentro del pedido (carrito).

La interfaz utiliza **Bootstrap 5** para mantener un diseño moderno, responsive y consistente.

1.2 Objetivo

Implementar un sistema funcional que permita:

- Conectarse a la base de datos **Productos_BD**.
- Navegar entre Pedidos, Productos y Clientes mediante una **NavBar**.
- Ejecutar operaciones CRUD sobre Productos y Clientes.
- Crear pedidos y administrar su carrito: agregar, quitar, vaciar y actualizar cantidades.
- Calcular el **total** del pedido en tiempo real.
- Servir imágenes de productos sin romper la ruta del proyecto.

1.3 Base de Datos y Conexión

El sistema utiliza la base de datos:

Productos_BD

La conexión se configura en un archivo de parámetros y se establece con `mysql.connector`.

Tablas utilizadas (según la lógica del sistema):

- **Productos**
- **Clientes**
- **Pedidos**
- **PedidosItems**

1.4 Interfaz y Navegación con NavBar

La aplicación incluye una barra de navegación superior (*NavBar*) con accesos directos a:

- **Pedidos**
- **Productos**
- **Clientes**

Esto permite que el usuario cambie de módulo sin perder el diseño general de la aplicación.

1.5 Arquitectura de la Aplicación

El proyecto se organiza en componentes:

- **app.py**: controlador principal (única ruta) que recibe GET/POST y delega la lógica a clases.
- **Producto, Cliente, Pedido**: clases que generan formularios HTML, listados, detalles y ejecutan SQL.
- **db.py y config.py**: conexión y parámetros de base de datos y puerto.

1.6 CRUD de Productos

El módulo de productos implementa:

- Listado de productos con imagen.
- Formulario para crear producto (con carga de imagen opcional).
- Formulario para actualizar producto.
- Vista de detalle.
- Eliminación de producto.

1.7 CRUD de Clientes

El módulo de clientes implementa:

- Listado general.
- Formulario de creación.
- Formulario de actualización.
- Vista detalle.
- Eliminación.

1.8 CRUD de Pedidos y Carrito

El módulo de pedidos administra carritos de compra:

- Listado de pedidos (carritos).
- Creación de pedido seleccionando cliente.
- Pantalla para elegir productos:
 - Agregar productos al carrito.
 - Quitar productos del carrito.
 - Vaciar carrito completo.
 - Actualizar cantidades.
 - Cálculo automático del total.
 - Opción de pago (marcar como pagado si la tabla lo permite).
- Vista detalle del pedido con su total.

1.9 Manejo de Imágenes

Para garantizar que las imágenes se visualicen correctamente, el sistema sirve archivos desde la carpeta de imágenes del proyecto utilizando parámetros en la misma ruta, evitando problemas de rutas absolutas.

1.10 Resultados Esperados

Al ejecutar el módulo, el usuario podrá:

- Administrar productos y clientes mediante CRUD completo.
- Crear pedidos asociados a clientes.
- Agregar y gestionar productos en el carrito.
- Actualizar cantidades y ver el total calculado.
- Visualizar imágenes de productos en listados y detalle.

Carritos de Compras (Pedidos)				
Cliente	Fecha	Borrar	Actualizar	Detalle
Comercial Tapia S.A.	2026-02-12 07:37:24	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-02-09 07:38:58	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-02-09 07:31:56	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-01-28 17:12:13	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-01-28 13:26:42	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-01-28 14:19:25	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Distribuidora Guayas Cía. Ltda.	2026-01-28 13:34:26	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Distribuidora Guayas Cía. Ltda.	2026-01-28 03:47:07	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Distribuidora Guayas Cía. Ltda.	2026-01-27 22:42:20	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-01-27 22:32:00	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>
Comercial Tapia S.A.	2026-01-27 22:30:29	<button>Borrar</button>	<button>Actualizar</button>	<button>Detalle</button>

Figura 1: NavBar del sistema (Pedidos, Productos, Clientes)

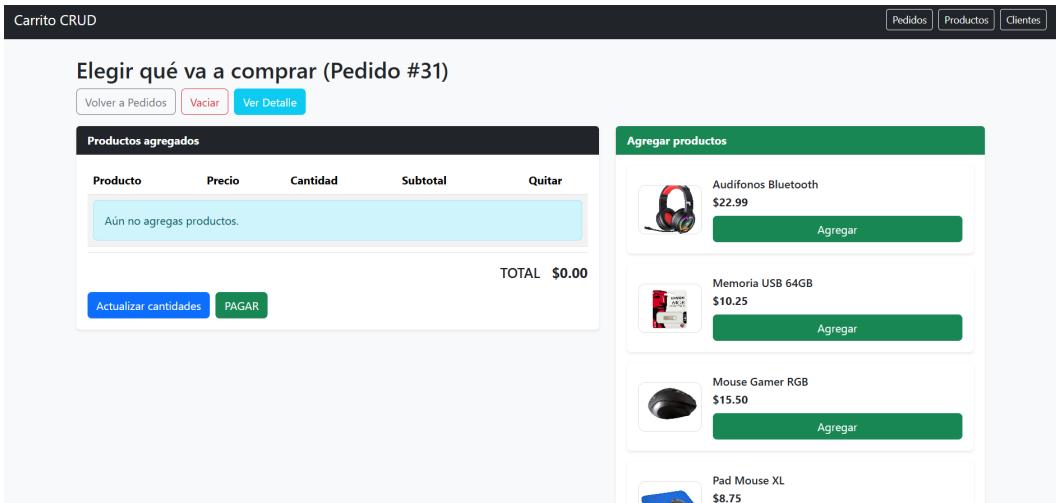


Figura 2: Pantalla de gestión del carrito: productos agregados y catálogo para agregar

Detalle del Pedido #2				
Cliente: Distribuidora Guayas Cia. Ltda. Fecha: 2026-01-21 16:40:00				
Producto	Precio	Cantidad	Subtotal	
Teclado Mecánico	\$45.00	2	\$90.00	
Audifonos Bluetooth	\$22.99	2	\$45.98	
Mouse Gamer RGB	\$15.50	1	\$15.50	

TOTAL \$151.48

[Actualizar \(Productos\)](#) [Volver](#)

Figura 3: Detalle del pedido con listado de productos y total

1.11 Importancia del Módulo

Este módulo integra un flujo completo de compra: gestión de datos maestros (clientes y productos) y operación principal (pedidos/carrito). Además, demuestra el uso práctico de POO en Flask, consultas SQL reales, control de navegación, y presentación de datos con Bootstrap, dejando una base sólida para futuras mejoras como autenticación, estados de pedido y reportes.