



MANUAL TECNICO DEL SOFTWARE

Sistema de Inventario web

Grupo A:

Cantos Miranda María Gabriela

Delgado Ordóñez Giancarlos Alejandro

Maldonado Maldonado Miguel Ángel

Tinoco Jiménez David Alejandro (líder)

1.	INTRODUCCIÓN	3
2.	OBJETIVOS DEL SISTEMA	3
3.	ALCANCE DEL SISTEMA	3
4.	HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	4
5.	ARQUITECTURA DEL SISTEMA	5
5.1.	Arquitectura General del Sistema.....	5
5.2.	Flujo de Peticiones	5
6.	DESCOMPOSICIÓN DE DATOS	7
6.1	Diagrama entidad-relación.....	7
6.2	Definiciones	8
5.	DISEÑO DE INTERFACES.....	10
7.	DISEÑO DETALLADO (DETAILED DESIGN)	16
7.1.	Diseño detallado de módulos	16
7.1.1.	Módulo de Productos	16
7.1.2.	Módulo de Ventas	16
7.1.3.	Módulo de Proveedores	16
7.1.4.	Módulo de Devoluciones.....	16
7.2.	Diseño detallado de datos	16
7.2.1.	Entidad Producto.....	16
7.2.2.	Entidad Venta	17
7.2.3.	Entidad Proveedor	17
7.2.4.	Entidad Devolución	17

1. INTRODUCCIÓN

Este documento tiene como objetivo proporcionar información técnica detallada sobre el Sistema de Inventario Web desarrollado en HTML, CSS, JavaScript y PHP bajo el patrón de arquitectura MVC (Modelo-Vista-Controlador). El manual está dirigido a personal técnico, administradores de sistemas, desarrolladores y departamentos de auditoría de sistemas, con el fin de facilitar el mantenimiento, la extensión y la supervisión del sistema.

2. OBJETIVOS DEL SISTEMA

Desarrollar un sistema web de inventario que permita gestionar productos, ventas, proveedores y devoluciones para negocios pequeños o medianos, con el objetivo de mejorar la eficiencia operativa, el control de stock y la toma de decisiones a partir de reportes de inventario y ventas.

3. ALCANCE DEL SISTEMA

- Módulo de Productos

Objetivo: Administrar los productos disponibles para la venta, permitiendo control y consulta en tiempo real.

- Registro, edición y eliminación de productos.
- Visualización del listado con filtros por nombre, código y categoría.

- Módulo de Ventas

Objetivo: Registrar las ventas de productos, actualizar automáticamente el inventario.

- Registro de ventas por cliente, fecha y productos.
- Cálculo automático de total y cambio.

- Módulo de Proveedores

Objetivo: Gestionar la información de los proveedores que abastecen productos.

- Registro de proveedores (nombre, RUC, dirección, teléfono, email).
- Listado, edición y eliminación de proveedores.

- Módulo de Devoluciones

Objetivo: Registrar y gestionar devoluciones de productos por parte de clientes o al proveedor.

- Registro de devoluciones con motivo
- Actualización del stock según tipo de devolución.

4. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

1. HTML5

Utilizado para la estructura semántica de las páginas web. Permite definir formularios, tablas, botones y la disposición general del contenido del sistema.

2. CSS3

Responsable del diseño visual y la presentación del sistema. Se emplearon estilos personalizados y clases para lograr una interfaz limpia y responsiva.

3. JavaScript (JS)

Implementado en el lado del cliente para validaciones de formularios, interacciones dinámicas con el DOM.

4. PHP

Lenguaje de programación del lado del servidor utilizado para gestionar la lógica del sistema, conectarse con la base de datos, ejecutar controladores del modelo MVC y generar vistas dinámicas.

5. MySQL

Sistema de gestión de bases de datos relacional utilizado para almacenar información estructurada del sistema como productos, ventas, proveedores y devoluciones. Se diseñaron tablas con claves foráneas y relaciones normalizadas.

6. GitHub

Plataforma de control de versiones utilizada para la gestión del código fuente, colaboraciones y respaldo del proyecto. Se emplearon ramas para el desarrollo modular y seguimiento de cambios.

7. Apache

Servidor web utilizado para ejecutar la aplicación en entornos de desarrollo y producción. Permite interpretar archivos PHP y servirlos al navegador.

5. ARQUITECTURA DEL SISTEMA

5.1. Arquitectura General del Sistema

El Sistema de Inventario Web se fundamenta en el patrón de arquitectura Modelo-Vista-Controlador (MVC), implementado en PHP, HTML, CSS y JavaScript. Esta elección promueve la separación de responsabilidades, facilitando el desarrollo, mantenimiento y escalabilidad del sistema.

La estructura de carpetas refleja esta arquitectura:

- **Config/:** Contiene archivos de configuración esenciales, como `conexion.php`, que gestiona la conexión a la base de datos.
- **Controladores/:** Aloja la lógica de negocio y la orquestación de las operaciones. Cada módulo principal (Productos, Proveedores, Ventas y Devoluciones) tiene su propio controlador (ej., `ControladorProductos.php`, `ControladorProveedor.php`, `ControladorVentas.php` y `ControladorDevoluciones.php`).
- **Modelos/:** Contiene las clases que interactúan directamente con la base de datos y encapsulan la lógica de persistencia de datos (ej., `ProductoModelo.php`, `ProveedorModelo.php`, `VentasModelo.php`, `DevolucionesModelo.php`).
- **Vistas/:** Almacena los archivos HTML/PHP responsables de la interfaz de usuario y la presentación de los datos (ej., `productos.php`, `proveedores.php`, `Principal.php`, `ventas.php`, `devoluciones.php`).
- **assets/:** Contiene recursos estáticos como hojas de estilo CSS (`assets/css/`) y archivos JavaScript (`assets/JS/`).

5.2. Flujo de Peticiones

- 1) *Inicio de la Sesión y Navegación:* El usuario accede al sistema a través de `Vistas/login.php` y, tras una autenticación exitosa, es redirigido a `Vistas/Principal.php`.
- 2) *Carga del Contenido Principal:* `Vistas/Principal.php` actúa como un layout principal que contiene una barra de navegación lateral y un `<iframe>`. Los enlaces de la barra lateral (gestionados por JavaScript en

assets/JS/principal.js) cambian dinámicamente el src del <iframe> para cargar las diferentes vistas de los módulos (ej., Vistas/productos.php, Vistas/provedores.php).

3) *Procesamiento en la Vista del Módulo:*

- I. Cuando una vista de módulo (ej., Vistas/productos.php o Vistas/provedores.php) es cargada en el <iframe>, esta instancia directamente su controlador correspondiente (ej., ControladorProductos.php o ControladorProveedor.php).
- II. El constructor del Controlador es el encargado de detectar el tipo de petición HTTP (GET o POST) y la acción solicitada (ej., "registrar", "actualizar", "eliminar")
- III. Basado en la acción, el Controlador invoca los métodos apropiados de su Modelo (ej., ProductoModelo o ProveedorModelo) para realizar operaciones CRUD en la base de datos.
- IV. Después de procesar la acción, el Controlador actualiza sus propiedades públicas (ej., \$productos, \$proveedores) con los datos más recientes o cualquier mensaje de estado.

4) Renderizado de la Vista: La misma vista del módulo utiliza las propiedades públicas del Controlador para:

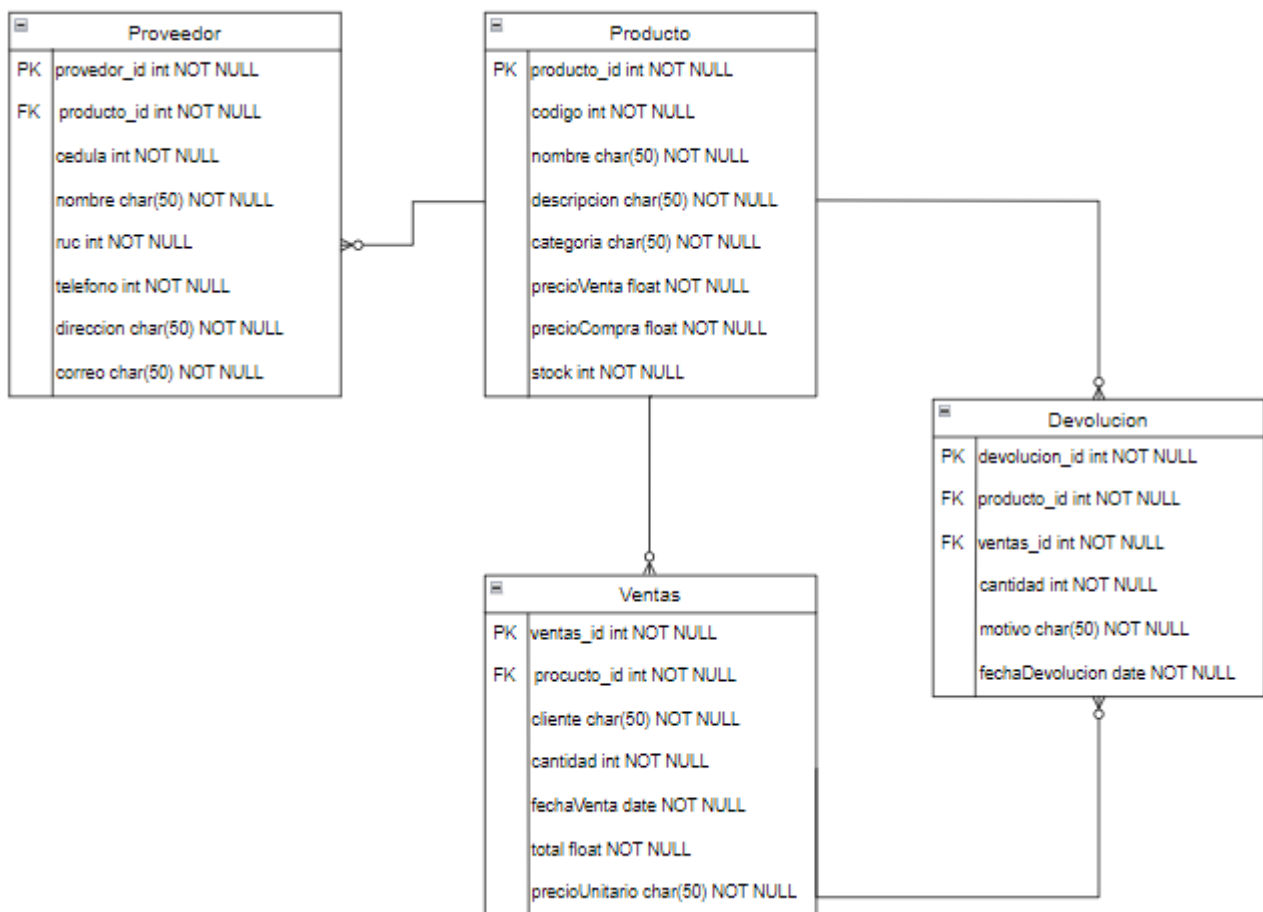
- I. Pre-llenar formularios en caso de edición.
- II. Mostrar la lista actualizada de elementos (productos, proveedores, ventas y devoluciones).
- III. Presentar mensajes de éxito o error al usuario.
- IV. Los formularios dentro de la vista envían sus peticiones (POST o GET) de vuelta a la misma vista, lo que reinicia el ciclo de procesamiento en el constructor del controlador.

5) *Interacción del Lado del Cliente:* JavaScript se utiliza para validaciones de formularios en tiempo real, manipulación del DOM y funcionalidades de interfaz de usuario como el modo oscuro y la barra de búsqueda en la navegación principal.

Esta arquitectura, aunque no utiliza un "Front Controller" centralizado explícito para todas las peticiones (como un index.php en la raíz), cada vista de módulo actúa como su propio punto de entrada y controlador de presentación, permitiendo un desarrollo modular y una clara separación de las capas de Modelo, Vista y Controlador.

6. DESCOMPOSICIÓN DE DATOS

6.1 Diagrama entidad-relación



6.2 Definiciones

NOMBRE OBJETO: Producto

DESCRIPCIÓN: Información del producto como nombre. Se relaciona con proveedor, ventas y devolución

Columnas:

Campo	Tipo de Dato	Restricciones	Descripción
producto_id	int	PK, NOT NULL	Identificador único del producto
codigo	int	NOT NULL	Código del producto
nombre	char(50)	NOT NULL	Nombre del producto
descripción	char(50)	NOT NULL	Descripción del producto
categoría	char(50)	NOT NULL	Categoría del producto
precioVenta	float	NOT NULL	Precio de venta
precioCompra	float	NOT NULL	Precio de compra
stock	int	NOT NULL	Cantidad actual en inventario

NOMBRE OBJETO: Proveedor

DESCRIPCIÓN: Información de proveedor como nombre. Se relaciona producto

Campo	Tipo de Dato	Restricciones	Descripción
proveedor_id	int	PK, NOT NULL	Identificador del proveedor
producto_id	int	FK, NOT NULL	Relación con productos suministrados
cedula	int	NOT NULL	Cédula o ID del proveedor
nombre	char(50)	NOT NULL	Nombre del proveedor
ruc	int	NOT NULL	Registro Único de Contribuyentes
telefono	int	NOT NULL	Número telefónico
direccion	char(50)	NOT NULL	Dirección física del proveedor
correo	char(50)	NOT NULL	Correo electrónico

NOMBRE OBJETO: Venta

DESCRIPCIÓN: Información de la venta realizada como nombre. Se relaciona producto

Campo	Tipo de Dato	Restricciones	Descripción
ventas_id	int	PK, NOT NULL	Identificador único de la venta
producto_id	int	FK, NOT NULL	Producto vendido
cliente	char(50)	NOT NULL	Nombre del cliente
cantidad	int	NOT NULL	Cantidad vendida
fechaVenta	date	NOT NULL	Fecha de la venta
total	float	NOT NULL	Monto total de la venta
precioUnitario	char(50)	NOT NULL	Precio por unidad

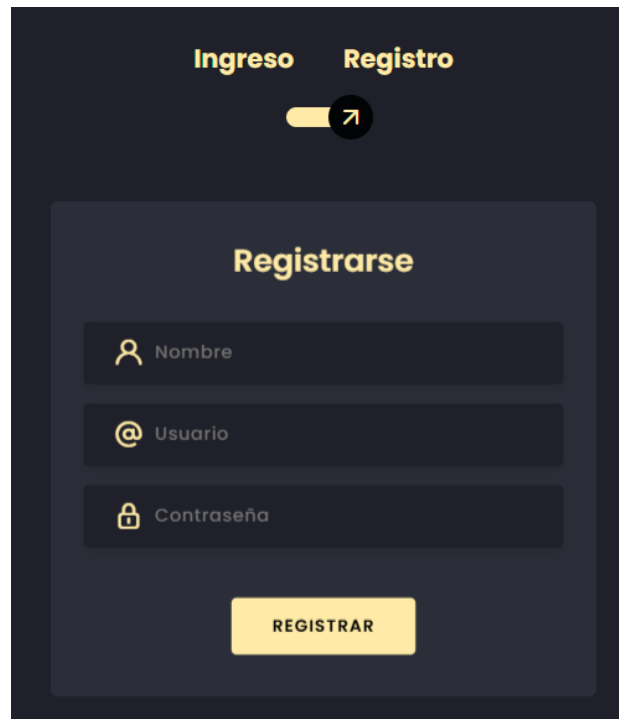
NOMBRE OBJETO: Devolución

DESCRIPCIÓN: Información de las devoluciones de productos como la fecha de devolución y cantidad

Campo	Tipo de Dato	Restricciones	Descripción
devolucion_id	int	PK, NOT NULL	Identificador de la devolución
producto_id	int	FK, NOT NULL	Producto devuelto
ventas_id	int	FK, NOT NULL	Venta relacionada
cantidad	int	NOT NULL	Cantidad devuelta
motivo	char(50)	NOT NULL	Razón de la devolución
fechaDevolucion	date	NOT NULL	Fecha en que se registró la devolución

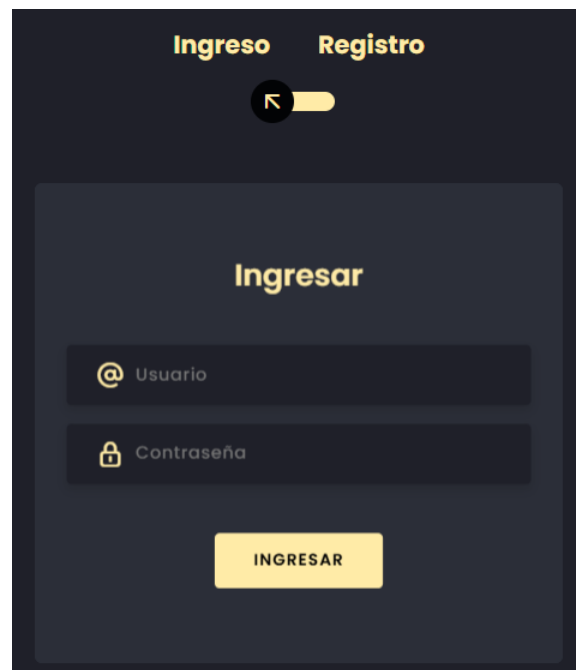
5. DISEÑO DE INTERFACES

- Registro al sistema



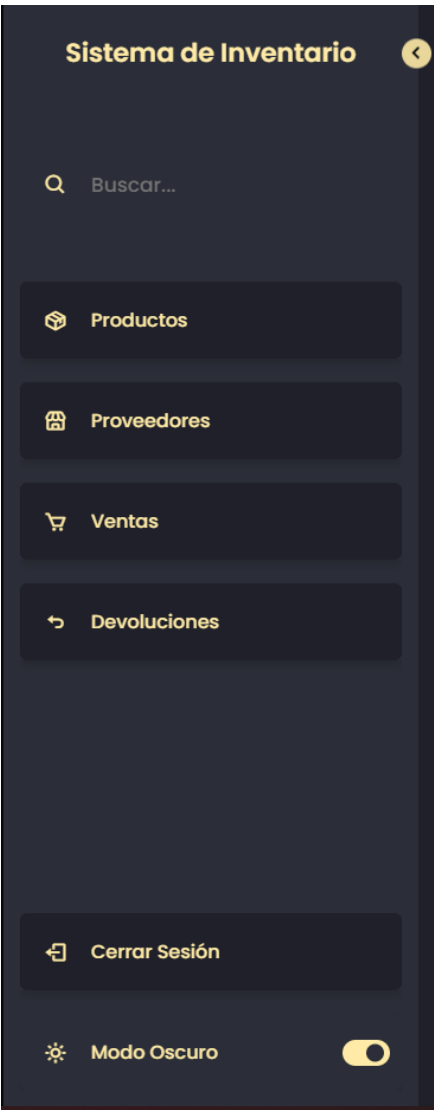
The registration form is displayed on a dark background. At the top, there are two tabs: 'Ingreso' and 'Registro'. The 'Registro' tab is selected, indicated by a yellow highlight and a right-pointing arrow icon. Below the tabs, the title 'Registrarse' is centered. The form contains three input fields: 'Nombre' (with a person icon), 'Usuario' (with an '@' icon), and 'Contraseña' (with a lock icon). A yellow button labeled 'REGISTRAR' is positioned at the bottom of the form.

- Ingreso al Sistema



The login form is displayed on a dark background. At the top, there are two tabs: 'Ingreso' and 'Registro'. The 'Ingreso' tab is selected, indicated by a yellow highlight and a left-pointing arrow icon. Below the tabs, the title 'Ingresar' is centered. The form contains two input fields: 'Usuario' (with an '@' icon) and 'Contraseña' (with a lock icon). A yellow button labeled 'INGRESAR' is positioned at the bottom of the form.

- Panel de Navegación



- Módulo de Productos.

REGISTRAR PRODUCTOS

Código

Nombre

Descripción

Categoría

Electrónica

Precio Venta

Precio Compra

Stock Inicial

Stock Mínimo

REGISTRAR

Filtrar por Código:

Filtrar por Categoría:

Todas

Filtrar por Stock:

Todos

FILTRAR

Código	Nombre	Categoría	Precio de Venta	Stock	Acciones
008	Cable OTG	electrónica	5.00	28	<div>EDITAR</div> <div>ELIMINAR</div>
002	Parlante Sony	electrónica	10.00	48	<div>EDITAR</div> <div>ELIMINAR</div>

- **Módulo de Proveedores.**

REGISTRAR PROVEEDOR

Cédula

Nombre

RUC

Teléfono

Dirección

Correo

Producto Asociado

Seleccione un producto

REGISTRAR

LISTA DE PROVEEDORES

Cédula	RUC	Nombre	Teléfono	Dirección	Correo	Producto Asociado	Acciones
No hay proveedores registrados.							

- Módulo de Ventas.

REGISTRAR VENTA

Cliente

Fecha de Venta

dd/mm/aaaa

AGREGAR PRODUCTO

Producto	Cantidad	Precio Unitario	Total Línea	Acciones
Total General:			0.00	

REGISTRAR VENTA

LISTA DE VENTAS

ID	Cliente	Productos	Total	Fecha	Acciones
18	Miguel	VER DETALLES	\$20.00	2025-05-31	EDITAR ELIMINAR
17	Miguel	VER DETALLES	\$25.00	2025-06-04	EDITAR ELIMINAR
16	Juan Carlos	VER DETALLES	\$40.00	2025-06-07	EDITAR ELIMINAR

- Módulo de Devoluciones.

REGISTRAR DEVOLUCIÓN

Devolución registrada correctamente

Cliente

Seleccione un cliente

Venta

Seleccione una venta

Producto

Seleccione un producto

Cantidad

Fecha de Devolución

dd/mm/aaaa

Motivo de Devolución

REGISTRAR DEVOLUCIÓN

LISTA DE DEVOLUCIONES

Cliente	Venta	Producto	Cantidad	Motivo	Fecha	Acciones
Juan Carlos	16	Cable OTG	1	Dañado	2025-07-09	ELIMINAR

7. DISEÑO DETALLADO (DETAILED DESIGN)

Esta sección presenta una descripción detallada de los módulos, procesos y entidades de datos del sistema, considerando su estructura interna, lógica y comportamiento esperado. Es clave para facilitar el mantenimiento, la auditoría técnica y futuras ampliaciones del sistema.

7.1. Diseño detallado de módulos

7.1.1. Módulo de Productos

- Validación de entradas mediante JavaScript.
- Estructura MVC: el controlador recibe los datos del formulario, llama al modelo para guardar en la base de datos y actualiza la vista.
- Funcionalidad: registrar, editar, eliminar y filtrar productos por nombre, código y categoría.

7.1.2. Módulo de Ventas

- Validación de cantidad y cálculo automático de total y cambio.
- Actualización en tiempo real del inventario.
- Registro en tabla de ventas con clave foránea al producto.
- Controlador centraliza la lógica de negocio y controla las operaciones de base de datos.

7.1.3. Módulo de Proveedores

- Formulario para registrar datos del proveedor con validación.
- Vista muestra la lista de proveedores con acciones de editar y eliminar.
- Asociación con productos mediante clave foránea en la tabla producto.

7.1.4. Módulo de Devoluciones

- Lógica para incrementar o decrementar stock según el tipo de devolución (cliente o proveedor).
- Registro asociado a la venta correspondiente mediante `ventas_id`.
- Motivo de devolución registrado para auditoría.

7.2. Diseño detallado de datos

7.2.1. Entidad Producto

- Gestionada a través de formularios web.
- Operaciones: Create, Read, Update, Delete (CRUD).

- Validaciones de tipo de dato, longitud, y existencia de claves foráneas (proveedor).

7.2.2. Entidad Venta

- Cálculo automático del total (cantidad × precioUnitario).
- Asociaciones con producto_id y cliente.
- Registro de fecha con función NOW() del sistema.

7.2.3. Entidad Proveedor

- Validación del campo RUC y correo electrónico.
- Asociación a múltiples productos.
- Acciones CRUD disponibles en la interfaz de administrador.

7.2.4. Entidad Devolución

- Registro detallado de motivos de devolución.
- Afecta directamente al stock del producto.
- Asociación directa con la venta y producto.