

## Manual técnico

### Introducción

Objetivo del Sistema: Permitir la gestión eficiente del inventario y registro de ventas en Ferrepinturas Colormax, facilitando la introducción y seguimiento del stock, así como la relación con los proveedores.

### Tecnologías utilizadas

- **Lenguaje de Programación:** PHP, JavaScript, HTML5, CSS3.
- **Gestor de estilos:** BootStrap.
- **Base de Datos:** MariaDB (versión 10.4.32).
- **Gestor de Base de Datos:** phpMyAdmin (versión 5.2.1) para la administración de MySQL y gestión de la base de datos a través de una interfaz web.

### Modelos de datos

- **Entidades Claves:**
  - **Categorías:** Almacena las categorías de productos (ID, nombre).
  - **Clientes:** Información de clientes (NID, nombres, apellidos, teléfono, correo).
  - **Proveedores:** Gestión de proveedores (RUT, nombre, contacto, dirección).
  - **Productos:** Detalles de los productos (ID, nombre, descripción, categoría, precio, ganancia, stock).
  - **Ventas:** Registro de cada venta realizada (ID, código de venta, producto, cliente, usuario, método de pago, cantidad, notas).

### Estructura de la Base de Datos

- Tablas:

```
Unset
-- Crear la tabla Clientes
CREATE TABLE IF NOT EXISTS Clientes (
    NID VARCHAR(20) PRIMARY KEY,
    primer_nombre VARCHAR(100) NOT NULL,
    segundo_nombre VARCHAR(100) DEFAULT 'Desconocido',
    primer_apellido VARCHAR(100) NOT NULL,
    segundo_apellido VARCHAR(100) DEFAULT 'Desconocido',
```

```

        telefono VARCHAR(20) DEFAULT NULL,
        correo VARCHAR(100) DEFAULT NULL
    );

-- Crear la tabla Usuarios
CREATE TABLE IF NOT EXISTS Usuarios (
    NID VARCHAR(20) PRIMARY KEY,
    primer_nombre VARCHAR(100) NOT NULL,
    segundo_nombre VARCHAR(100) DEFAULT 'Desconocido',
    primer_apellido VARCHAR(100) NOT NULL,
    segundo_apellido VARCHAR(100) DEFAULT 'Desconocido',
    telefono1 VARCHAR(20) NOT NULL,
    telefono2 VARCHAR(20) DEFAULT NULL,
    correo VARCHAR(100) NOT NULL,
    direccion TEXT DEFAULT NULL,
    tipo ENUM('admin', 'normal') NOT NULL
);

-- Crear la tabla Métodos de pago
CREATE TABLE IF NOT EXISTS Metodos_de_pago (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

-- Crear la tabla Categorías
CREATE TABLE IF NOT EXISTS Categorías (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

-- Crear la tabla Productos
CREATE TABLE IF NOT EXISTS Productos (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT DEFAULT NULL,
    id_categoria INT,
    precio_compra DECIMAL(10,2) NOT NULL,
    porcentaje_ganancia DECIMAL(5,2) NOT NULL,
    cantidad_actual INT NOT NULL,
    cantidad_minima INT NOT NULL,
    FOREIGN KEY (id_categoria) REFERENCES Categorías(ID)
);

-- Crear la tabla Ventas

```

```

CREATE TABLE IF NOT EXISTS Ventas (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    codigo_venta VARCHAR(20) NOT NULL,
    id_producto INT,
    id_cliente VARCHAR(20) DEFAULT NULL,
    id_usuario VARCHAR(20) NOT NULL,
    id_metodo_pago INT NOT NULL,
    fecha DATE NOT NULL,
    cantidad int(10) UNSIGNED NOT NULL,
    notas TEXT DEFAULT NULL,
    FOREIGN KEY (id_producto) REFERENCES Productos(ID),
    FOREIGN KEY (id_cliente) REFERENCES Clientes(NID),
    FOREIGN KEY (id_usuario) REFERENCES Usuarios(NID),
    FOREIGN KEY (id_metodo_pago) REFERENCES Metodos_de_pago(ID)
);

-- Crear la tabla Proveedores
CREATE TABLE IF NOT EXISTS Proveedores (
    RUT VARCHAR(20) PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    telefono1 VARCHAR(20) NOT NULL,
    telefono2 VARCHAR(20) DEFAULT NULL,
    correo VARCHAR(100) DEFAULT NULL,
    direccion TEXT DEFAULT NULL
);

-- Crear la tabla Compras
CREATE TABLE IF NOT EXISTS Compras (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    id_proveedor VARCHAR(20) NOT NULL,
    fecha_orden DATE NOT NULL,
    total_pagar DECIMAL(10,2) NOT NULL,
    fecha_pago DATE NOT NULL,
    FOREIGN KEY (id_proveedor) REFERENCES Proveedores(RUT)
);

-- Crear la tabla Productos-Compras
CREATE TABLE IF NOT EXISTS Productos_Compras (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    id_producto INT,
    id_compra INT,
    FOREIGN KEY (id_producto) REFERENCES Productos(ID),
    FOREIGN KEY (id_compra) REFERENCES Compras(ID)
);

```

```
-- Crear la tabla Vendedores
CREATE TABLE IF NOT EXISTS Vendedores (
    NID VARCHAR(20) PRIMARY KEY,
    primer_nombre VARCHAR(100) NOT NULL,
    segundo_nombre VARCHAR(100) DEFAULT 'Desconocido',
    primer_apellido VARCHAR(100) NOT NULL,
    segundo_apellido VARCHAR(100) DEFAULT 'Desconocido',
    telefono1 VARCHAR(20) NOT NULL,
    telefono2 VARCHAR(20) DEFAULT NULL,
    correo VARCHAR(100) DEFAULT NULL,
    id_proveedor VARCHAR(20) NOT NULL,
    FOREIGN KEY (id_proveedor) REFERENCES Proveedores(RUT)
);
```

- Vistas:

Unset

```
-- Vista para ventas.php
create view ventas_view as
SELECT
    ventas.ID AS id,
    ventas.codigo_venta AS codigo,
    productos.nombre AS producto,
    ventas.cantidad,
    productos.precio_compra AS producto_precio,
    productos.porcentaje_ganancia AS producto_ganancia,
    ventas.fecha AS fecha_registro,
    clientes.primer_nombre AS cliente_primer_nombre,
    clientes.segundo_nombre AS cliente_segundo_nombre,
    clientes.primer_apellido AS cliente_primer_apellido,
    clientes.segundo_apellido AS cliente_segundo_apellido,
    usuarios.primer_nombre AS usuario_primer_nombre,
    usuarios.segundo_nombre AS usuario_segundo_nombre,
    usuarios.primer_apellido AS usuario_primer_apellido,
    usuarios.segundo_apellido AS usuario_segundo_apellido,
    metodos_de_pago.nombre AS metodo_de_pago,
    ventas.notas AS notas
FROM Ventas
JOIN Productos ON Ventas.id_producto = Productos.ID
JOIN Clientes ON Ventas.id_cliente = Clientes.NID
JOIN Usuarios ON Ventas.id_usuario = Usuarios.NID
```

```

JOIN Metodos_de_pago ON Ventas.id_metodo_pago = Metodos_de_pago.ID;

-- Vista para inventario.php
CREATE VIEW inventario AS
SELECT
    productos.ID as id,
    productos.nombre as nombre,
    productos.descripcion as descripcion,
    categorias.nombre as categoria,
    productos.precio_compra as precio_compra,
    productos.porcentaje_ganancia as ganancia,
    productos.cantidad_actual as stock_actual,
    productos.cantidad_minima as stock_minimo
from productos join categorias on productos.id_categoria = categorias.ID;

-- Vista para proveedores.php
create view proveedores_con_vendedores as
select
    proveedores.rut,
    proveedores.nombre as nombre_proveedor,
    proveedores.telefono1 as proveedor_telefono_1,
    proveedores.telefono2 as proveedor_telefono_2,
    proveedores.correo as proveedor_correo,
    proveedores.direccion,
    vendedores.NID,
    vendedores.primer_nombre,
    vendedores.segundo_nombre,
    vendedores.primer_apellido,
    vendedores.segundo_apellido,
    vendedores.telefono1,
    vendedores.telefono2,
    vendedores.correo as vendedor_correo
from proveedores left join vendedores on proveedores.RUT =
vendedores.id_proveedor
order by proveedores.nombre, vendedores.primer_nombre;

-- Vista para el selector de productos en registro_venta.php
create view registro_venta_productos as
select
    productos.ID as id,
    productos.nombre,
    categorias.nombre as categoria,
    productos.precio_compra,
    productos.porcentaje_ganancia as ganancia

```

```
from productos join categorias on productos.id_categoria = categorias.ID
order by productos.nombre;
```

## Interfaz de usuario

- **Diseño Amigable:** Incluir capturas de pantalla y explicaciones de las funciones accesibles desde la interfaz.
- **Guía de Uso:** Instrucciones paso a paso sobre cómo registrar una venta y acceder al inventario.

## Seguridad

- **Acceso a Datos:** Implementar validaciones y roles de usuarios para evitar accesos no autorizados. Utilizar conexiones seguras a la base de datos.

## Mantenimiento y Escalabilidad

- **Copia de Seguridad:** Protocolo para realizar backups de la base de datos usando phpMyAdmin al menos una vez cada semana.
- **Optimización:** Monitorizar el uso del disco y la memoria debido a las restricciones de hardware.

## Implementación

- **Guía de Instalación:**
  1. Instalar XAMPP para ejecutar un servidor apache en local.
  2. Ejecutar el algoritmo de creación de la base de datos.
  3. Copiar los documentos de la aplicación en una carpeta dentro del directorio htdocs de XAMPP
- **Configuración Inicial:** Explicaciones sobre cómo importar la base de datos desde un archivo de exportación SQL. Para esto, simplemente se debe abrir phpMyAdmin, abrir una tableta de SQL, copiar el código de exportación de la Base de Datos y ejecutar.

## Documentación Adicional

- [Diagrama entidad-relación](#)
- [Diagrama relacional](#)
- [Diccionario de datos](#)