

Documentación Técnica y Manual de Despliegue: CRM Tienda Online

CRM Dashboard

1. Stack Tecnológico del Sistema

El sistema "CRM Tienda Online" ha sido desarrollado utilizando un stack moderno basado en Javascript/Typescript, conocido comúnmente como el stack **MERN/PERN** (reemplazando Mongo/Postgres por MySQL).

Tech Stack

Frontend (Interfaz de Usuario)

- **React 18**: Biblioteca principal para la construcción de interfaces de usuario interactivas.
- **Vite**: Herramienta de compilación y servidor de desarrollo extremadamente rápido.
- **Tailwind CSS 3.4**: Framework de CSS de utilidad (Utility-first) para un diseño rápido, responsive y moderno.
- **React Router DOM 6**: Para la navegación y enrutamiento entre las diferentes vistas (SPA).
- **Recharts**: Para la visualización de datos, métricas y gráficos en los dashboards.
- **Axios**: Cliente HTTP para realizar peticiones al backend.
- **React Hot Toast**: Para notificaciones y alertas en tiempo real.

Backend (Servidor y API)

- **Node.js**: Entorno de ejecución para Javascript en el servidor.
- **Express.js 4.18**: Framework minimalista para la creación de la API REST.
- **MySQL 2 (mysql2)**: Driver para la conexión y consultas a la base de datos relacional.
- **JSON Web Tokens (JWT)**: Para la autenticación y autorización segura basada en tokens.
- **Bcrypt.js**: Para el encriptado y hashing seguro de contraseñas de usuarios.
- **Multer**: Middleware para el manejo y subida de archivos (imágenes de productos, logos, etc.).
- **Express Validator**: Middleware para la validación de los datos de entrada en los endpoints de la API.

Base de Datos

- **MySQL 8.x o superior**: Sistema de gestión de bases de datos relacionales utilizado para persistir todos los datos del CRM, inventario y e-commerce.

2. Funcionalidades Core del Sistema

El sistema es un completo CRM y ERP ligero orientado a comercio electrónico y gestión de tienda física, abarcando las siguientes funcionalidades:

Gestión de Catálogo y Productos

- Creación de productos simples y con **variantes** (talla, color, modelo, procesador, etc.).
- Gestión de categorías anidadas (Jerarquía de categorías).
- Gestión de Marcas y Unidades de Medida.
- Asignación de múltiples imágenes por producto, atributos flexibles y precios dinámicos (costo, venta, oferta).
- Alertas para productos perecederos o que requieren receta.

Gestión de Inventario y Almacenes

- Manejo de **Multi-almacén** / Sucursales (permitiendo definir stock independiente por local).
- Kardex y Movimientos de Stock (Entradas, Salidas, Ajustes, Devoluciones y Transferencias).
- Definición de Stock Mínimo para alertas de reabastecimiento y ubicaciones físicas (pasillo/estante).

CRM: Clientes y Usuarios

- Registro detallado de clientes (Personas naturales o Empresas con RUC/DNI).
- Sistema de roles de usuario: **Administrador**, **Vendedor**, **Almacenero**, **Supervisor**.
- Autenticación segura y control de acceso según permisos.

Ventas y Pedidos (Punto de Venta y Online)

- Creación de pedidos/ventas desde múltiples canales: Tienda física, Online, Teléfono, Delivery.
- Control de estados de pedido (Borrador, Confirmado, En Proceso, Enviado, Entregado).
- Soporte para múltiples métodos de pago (Efectivo, Tarjeta, Transferencia, Yape, Plin).
- Cálculo automático de impuestos, descuentos (por porcentaje o monto fijo) y costos de envío.

Compras y Proveedores

- Base de datos de proveedores.
- Generación y seguimiento de **Órdenes de Compra** (Estados: Borrador, Enviada, Recibida Parcial/Total).
- Recepción de stock directo desde órdenes de compra.

Marketing y Promociones

- Sistema de **Cupones de Descuento** con restricciones (uso máximo, monto mínimo, fecha de expiración).

Configuración del Sistema

- Ajustes globales dinámicos: Moneda, país, nombre de tienda, logo, impuestos y prefijos de comprobantes.

3. Guía de Instalación Local Paso a Paso (Computadora Nueva)

Si tienes una PC nueva (con Windows), estos son los pasos para poner a funcionar el sistema.

Fase A: Programas a Descargar e Instalar

1. **Node.js:**
 - Descargar la versión **LTS (Long Term Support)** desde nodejs.org (<https://nodejs.org/>).
 - Instalar dando clic en "Siguiente" a todo (asegurarse de que "Add to PATH" esté marcado).
2. **XAMPP** (Para MySQL):
 - Descargar desde [apachefriends.org](https://www.apachefriends.org/) (<https://www.apachefriends.org/>).
 - En la instalación, puedes desmarcar todo excepto **MySQL** y **phpMyAdmin**.
3. **Visual Studio Code (VSCode)** (Opcional, pero recomendado):
 - Descargar desde code.visualstudio.com (<https://code.visualstudio.com/>) como editor de código.
4. **Git** (Opcional, si usarás control de versiones):
 - Descargar de git-scm.com (<https://git-scm.com/>).

Fase B: Configuración de la Base de Datos

1. Abre **XAMPP Control Panel** y da clic en "Start" al lado de **MySQL** (y **Apache** si deseas usar phpMyAdmin).
2. Abre tu navegador y ve a <http://localhost/phpmyadmin>.
3. Haz clic en "Nueva" (base de datos). Nómbrala `tiendaonline_crm` y selecciona "Crear" (con cotejamiento `utf8mb4_unicode_ci`).
4. Selecciona la base de datos recién creada, ve a la pestaña "Importar".
5. Selecciona el archivo `c:\CRMMyERP\crm-tiendaonline\database\schema.sql` y dale a "Importar". Esto creará todas las tablas y datos iniciales (incluyendo el usuario admin).

Fase C: Configurar e Iniciar el Backend

1. Abre una terminal (Símbolo del sistema o PowerShell).
2. Navega a la carpeta del backend:

```
cd c:\CRMMyERP\crm-tiendaonline\backend
```

3. Instala las dependencias:

```
npm install
```

4. Copia el archivo de entorno de ejemplo:

```
copy .env.example .env
```

5. Abre el archivo `.env` en un bloc de notas o VSCode y verifica las credenciales de la BD. Si usas XAMPP por defecto:

```
DB_HOST=127.0.0.1
DB_USER=root
DB_PASSWORD=
DB_NAME=tiendaonline_crm
```

6. Inicia el servidor backend:

```
npm run dev
```

(Debería mostrar un mensaje diciendo que está corriendo en el puerto 5000 o similar y conectado a la BD).

Fase D: Configurar e Iniciar el Frontend

1. Abre una **nueva** ventana de terminal.
2. Navega a la carpeta del frontend:

```
cd c:\CRMMyERP\crm-tiendaonline\frontend
```

3. Instala las dependencias:

```
npm install
```

4. Inicia el servidor de desarrollo:

```
npm run dev
```

5. La consola mostrará una URL local (generalmente `http://localhost:5173/`). Ábrela en tu navegador y verás el CRM funcionando.

4. Hosting Web Sugerido

Para una aplicación basada en **Node.js**, **React** y **MySQL**, NO es recomendable utilizar hosting compartido tradicional (como HostGator o Bluehost) ya que no están optimizados para Node.js y a menudo no te dan acceso por terminal completo.

La recomendación profesional:

- **Hostinger (VPS):** Ofrece servidores privados virtuales a bajo costo con panel de control integrado (CyberPanel o aaPanel).
- **DigitalOcean (Droplet):** Extremadamente confiable, ideal para desarrolladores, pero requiere conocimientos de terminal en Linux.
- **Render.com / Railway.app** (Plataformas PaaS): Las opciones más fáciles. Puedes alojar la API y el Frontend separadamente y usar un proveedor de MySQL en la nube (como PlanetScale o Aiven).

El Mejor Balance (Precio / Facilidad): Hostinger (Plan VPS KVM 1 o superior con CyberPanel/aaPanel) o Railway.app para evitar configurar el servidor desde cero.

A continuación, proveemos la guía para **Railway.app** (por su extrema simplicidad) y **VPS con aaPanel** (por si prefieres control total). Abordaremos la guía para **VPS genérico (Ubuntu) / Hostinger** ya que es lo más solicitado en producción.

5. Guía de Despliegue en VPS (Hostinger / DigitalOcean con Ubuntu 22.04)

Paso 1: Preparación del Servidor

1. Adquiere el VPS y accede vía SSH (usando PuTTY o la terminal de Windows): `ssh root@TU_IP_PUBLICA`
2. Instala Node.js, MySQL y Nginx:

```
sudo apt update && sudo apt upgrade -y
sudo apt install nginx mysql-server -y
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-install nodejs
```

3. Instala PM2 (Gestor de procesos para mantener el backend encendido):

```
sudo npm install -g pm2
```

Paso 2: Configuración de Base de Datos en Producción

1. Entra a MySQL: `sudo mysql`
2. Crea la base de datos y usuario:

```
CREATE DATABASE tiendaonline_crm;
CREATE USER 'crm_user'@'localhost' IDENTIFIED BY 'TuContraseñaSegura';
GRANT ALL PRIVILEGES ON tiendaonline_crm.* TO 'crm_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

3. Importa tu `schema.sql` (Puedes subirlo por SFTP usando FileZilla):

```
mysql -u crm_user -p tiendaonline_crm < schema.sql
```

Paso 3: Despliegue del Backend (API)

1. Sube la carpeta `/backend` al servidor usando FileZilla a `/var/www/crm-backend`.
2. Vía SSH, navega a esa carpeta:

```
cd /var/www/crm-backend
npm install
```

3. Crea tu archivo `.env` de producción con los datos de BD del paso 2.
4. Inicia la API con PM2:

```
pm2 start src/index.js --name "crm-api"
pm2 save
pm2 startup
```

Paso 4: Despliegue del Frontend (React/Vite)

1. En tu computadora local, abre la terminal en `/frontend` y cambia la URL base de Axios para que apunte a tu API de producción (ej. `https://api.tudominio.com`).
2. Genera los archivos estáticos de producción:

```
npm run build
```

3. Esto generará una carpeta `dist`. Sube el *contenido* de esta carpeta `dist` por FileZilla a `/var/www/crm-frontend` en el servidor.

Paso 5: Configurar Nginx (Proxy y Dominio)

1. Crea un archivo de configuración para el frontend:

```
sudo nano /etc/nginx/sites-available/crm-frontend
```

Agrega:

```
server {
    listen 80;
    server_name app.tudominio.com;
    root /var/www/crm-frontend;
    index index.html;
    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

2. Crea configuración para el backend:

```
sudo nano /etc/nginx/sites-available/crm-api
```

Agrega:

```
server {
    listen 80;
    server_name api.tudominio.com;
    location / {
        proxy_pass http://localhost:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

3. Activa los sitios y reinicia Nginx:

```
sudo ln -s /etc/nginx/sites-available/crm-frontend /etc/nginx/sites-enabled/  
sudo ln -s /etc/nginx/sites-available/crm-api /etc/nginx/sites-enabled/  
sudo systemctl restart nginx
```

4. *(Opcional pero Obligatorio hoy en día)* Instala **Certbot** para obtener SSL gratis (HTTPS):

```
sudo apt install certbot python3-certbot-nginx  
sudo certbot --nginx -d app.tudominio.com -d api.tudominio.com
```

¡Felicidades! Tu sistema ahora está en producción.