

Manual de Instalación y Configuración - Sistema POS para Tienda de Accesorios Móviles

Fecha: 6 de junio de 2025

Versión: 1.0

Tabla de Contenidos

1. [Introducción](#)
2. [Requisitos del Sistema](#)
3. [Instalación del Backend](#)
4. [Instalación del Frontend](#)
5. [Configuración de la Base de Datos](#)
6. [Configuración del Sistema](#)
7. [Despliegue en Producción](#)
8. [Actualización del Sistema](#)
9. [Solución de Problemas](#)

Introducción

Este manual proporciona instrucciones detalladas para la instalación y configuración del Sistema POS para Tienda de Accesorios Móviles. El sistema está compuesto por un backend desarrollado en NestJS, un frontend desarrollado en ReactJS y Electron, y una base de datos PostgreSQL.

Componentes del Sistema

El sistema consta de los siguientes componentes:

- **Backend:** Desarrollado en NestJS, proporciona la API RESTful que el frontend consume.
- **Frontend:** Desarrollado en ReactJS y Electron, proporciona la interfaz de usuario.
- **Base de Datos:** PostgreSQL, almacena todos los datos del sistema.

Requisitos del Sistema

Requisitos de Hardware

- **Procesador:** Intel Core i3 o equivalente
- **Memoria RAM:** 4 GB mínimo (8 GB recomendado)
- **Espacio en Disco:** 1 GB mínimo
- **Conexión a Internet:** Requerida para la instalación y actualizaciones

Requisitos de Software

- **Sistema Operativo:** Windows 10/11, macOS 10.14 o superior, o Linux (Ubuntu 20.04 o superior)
- **Node.js:** v14.0.0 o superior
- **npm:** v6.0.0 o superior
- **PostgreSQL:** v12.0 o superior
- **Git:** Para clonar el repositorio (opcional)

Instalación del Backend

Paso 1: Clonar el Repositorio

Si tiene acceso al repositorio Git, puede clonar el proyecto:

```
git clone https://github.com/usuario/pos-system.git  
cd pos-system
```

Alternativamente, puede descargar el archivo ZIP del proyecto y descomprimirlo.

Paso 2: Instalar Dependencias

Navegue al directorio del backend e instale las dependencias:

```
cd backend/pos-backend  
npm install
```

Paso 3: Configurar Variables de Entorno

Cree un archivo `.env` en el directorio del backend con la siguiente configuración:

```
# Configuración de la base de datos
DB_HOST=localhost
DB_PORT=5432
DB_USERNAME=posadmin
DB_PASSWORD=posadmin123
DB_DATABASE=pos_db

# Configuración de JWT
JWT_SECRET=su_clave_secreta_aqui
JWT_EXPIRATION=1d

# Configuración del servidor
PORT=3001
NODE_ENV=production
```

Asegúrese de reemplazar `su_clave_secreta_aqui` con una clave segura para la generación de tokens JWT.

Paso 4: Compilar el Backend

Compile el backend para producción:

```
npm run build
```

Instalación del Frontend

Paso 1: Instalar Dependencias

Navegue al directorio del frontend e instale las dependencias:

```
cd ../../frontend/pos-frontend
npm install
```

Paso 2: Configurar Variables de Entorno

Cree un archivo `.env.production` en el directorio del frontend con la siguiente configuración:

```
REACT_APP_API_URL=http://localhost:3001/api
REACT_APP_ENV=production
GENERATE_SOURCEMAP=false
```

Si el backend se ejecuta en un servidor diferente, reemplace `localhost:3001` con la dirección correcta.

Paso 3: Compilar el Frontend

Compile el frontend para producción:

```
npm run build
```

Paso 4: Empaquetar la Aplicación Electron

Para crear una aplicación de escritorio con Electron:

```
npm run electron:build
```

Esto generará los archivos de instalación en el directorio `dist`.

Configuración de la Base de Datos

Paso 1: Instalar PostgreSQL

Si aún no tiene PostgreSQL instalado, siga las instrucciones para su sistema operativo:

Windows

1. Descargue el instalador desde [el sitio oficial de PostgreSQL](#).
2. Ejecute el instalador y siga las instrucciones.
3. Durante la instalación, establezca la contraseña para el usuario `postgres`.

macOS

Utilizando Homebrew:

```
brew install postgresql  
brew services start postgresql
```

Linux (Ubuntu)

```
sudo apt update  
sudo apt install postgresql postgresql-contrib
```

```
sudo systemctl start postgresql  
sudo systemctl enable postgresql
```

Paso 2: Crear Usuario y Base de Datos

Acceda a PostgreSQL como usuario `postgres` :

```
sudo -u postgres psql
```

Cree el usuario y la base de datos:

```
CREATE USER posadmin WITH PASSWORD 'posadmin123';  
CREATE DATABASE pos_db;  
GRANT ALL PRIVILEGES ON DATABASE pos_db TO posadmin;  
ALTER USER posadmin WITH SUPERUSER;  
\q
```

Paso 3: Importar el Esquema de la Base de Datos

Importe el esquema SQL proporcionado:

```
sudo -u postgres psql -d pos_db -f /ruta/al/esquema_pos.sql
```

Configuración del Sistema

Configuración Inicial

La primera vez que inicie el sistema, se creará automáticamente un usuario administrador con las siguientes credenciales:

- **Correo Electrónico:** admin@example.com
- **Contraseña:** admin123

Se recomienda cambiar esta contraseña inmediatamente después de iniciar sesión por primera vez.

Configuración de la Compañía

Para configurar la información de su compañía:

1. Inicie sesión como administrador.

2. Vaya a la sección "Administración".
3. Seleccione la pestaña "Compañía".
4. Complete la información de su compañía.
5. Haga clic en el botón "Guardar".

Configuración de Tiendas

Para configurar las tiendas:

1. Inicie sesión como administrador.
2. Vaya a la sección "Administración".
3. Seleccione la pestaña "Tiendas".
4. Haga clic en el botón "Nueva Tienda".
5. Complete la información de la tienda.
6. Haga clic en el botón "Guardar".

Configuración de Usuarios

Para configurar usuarios adicionales:

1. Inicie sesión como administrador.
2. Vaya a la sección "Administración".
3. Seleccione la pestaña "Usuarios".
4. Haga clic en el botón "Nuevo Usuario".
5. Complete la información del usuario.
6. Haga clic en el botón "Guardar".

Despliegue en Producción

Para un entorno de producción, se recomienda utilizar un servidor web como Nginx o Apache para servir el frontend, y un gestor de procesos como PM2 para el backend.

Configuración de PM2

Instale PM2 globalmente:

```
npm install -g pm2
```

Inicie el backend con PM2:

```
cd /ruta/al/backend/pos-backend
pm2 start dist/main.js --name pos-backend
```

Configure PM2 para iniciar automáticamente al arrancar el sistema:

```
pm2 startup
pm2 save
```

Configuración de Nginx

Instale Nginx:

```
sudo apt update
sudo apt install nginx
```

Cree un archivo de configuración para el frontend:

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

Agregue la siguiente configuración:

```
server {
    listen 80;
    server_name example.com;
    root /ruta/al/frontend/pos-frontend/build;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api {
        proxy_pass http://localhost:3001/api;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Reemplace `example.com` con su dominio y `/ruta/al/frontend/pos-frontend/build` con la ruta correcta al directorio `build` del frontend.

Active la configuración:

```
sudo ln -s /etc/nginx/sites-available/pos-frontend /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx
```

Configuración de HTTPS

Para configurar HTTPS con Let's Encrypt:

```
sudo apt install certbot python3-certbot-nginx  
sudo certbot --nginx -d example.com
```

Reemplace `example.com` con su dominio.

Actualización del Sistema

Actualización del Backend

Para actualizar el backend:

1. Detenga el servicio: `bash pm2 stop pos-backend`
2. Actualice el código: `bash cd /ruta/al/backend/pos-backend git pull #`
`Si está utilizando Git`
3. Instale las dependencias: `bash npm install`
4. Compile el código: `bash npm run build`
5. Reinicie el servicio: `bash pm2 restart pos-backend`

Actualización del Frontend

Para actualizar el frontend:

1. Actualice el código: `bash cd /ruta/al/frontend/pos-frontend git pull #`
`Si está utilizando Git`
2. Instale las dependencias: `bash npm install`
3. Compile el código: `bash npm run build`

4. Si está utilizando Electron, empaquete la aplicación: `bash npm run electron:build`

Actualización de la Base de Datos

Si hay cambios en el esquema de la base de datos, aplique las migraciones:

```
cd /ruta/al/backend/pos-backend  
npm run migration:run
```

Solución de Problemas

Problemas de Conexión a la Base de Datos

Problema: El backend no puede conectarse a la base de datos.

Solución: 1. Verifique que PostgreSQL esté en ejecución: `bash sudo systemctl status postgresql`

1. Verifique que las credenciales en el archivo `.env` sean correctas.
2. Verifique que el usuario tenga los permisos necesarios: `bash sudo -u postgres psql -c "ALTER USER posadmin WITH SUPERUSER;"`

Problemas de Inicio del Backend

Problema: El backend no inicia correctamente.

Solución: 1. Verifique los logs: `bash pm2 logs pos-backend`

1. Verifique que todas las dependencias estén instaladas: `bash cd /ruta/al/backend/pos-backend npm install`
2. Verifique que el puerto especificado en el archivo `.env` no esté siendo utilizado por otra aplicación.

Problemas de Acceso al Frontend

Problema: No se puede acceder al frontend.

Solución: 1. Verifique que Nginx esté en ejecución: `bash sudo systemctl status nginx`

1. Verifique la configuración de Nginx: `bash sudo nginx -t`
2. Verifique que el directorio `build` del frontend exista y contenga los archivos necesarios.

Problemas de Autenticación

Problema: No se puede iniciar sesión en el sistema.

Solución: 1. Verifique que el backend esté en ejecución y accesible.

1. Verifique que la clave JWT en el archivo `.env` sea correcta.
2. Intente restablecer la contraseña del usuario administrador: `bash sudo -u postgres psql -d pos_db -c "UPDATE usuarios SET password = '$2b$10$X7VYVy9HmKi3dQicZKWZa0u4aLB0t0jYR1wFN1S1H.U9iZt2T5W4e' WHERE email = 'admin@example.com';"` Esto restablecerá la contraseña a `admin123`.

Si encuentra algún problema que no esté cubierto en este manual, por favor contacte al soporte técnico en support@possystem.com o llame al +1-800-123-4567.