Implementación de Ambiente de Desarrollo

Guía práctica para preparar y ejecutar el entorno de desarrollo del proyecto POZINOX (tienda web + backoffice + API).

Datos del documento

- Versión: 1.0
- Fecha: 17/10/2025
- Responsable: Benjamín Véliz

Información del Proyecto

- Nombre del proyecto: POZINOX – Tienda web con inventario y bot de ventas
- Institución: Duoc UC – Escuela de Informática y Telecomunicaciones
- Descripción: Plataforma web con flujo catálogo → cotización → pago e inventario con roles.

Integrantes

- Benjamín Véliz – Desarrollo y documentación
- Bastián Sepúlveda – Backoffice de productos y UI
- Sebastián Cerón – Usuarios/roles, pagos e integraciones

Introducción

Este documento define los prerrequisitos, instalación y configuración para que cualquier integrante pueda levantar el proyecto en su equipo sin fricción.

Alcance

- Entorno local (DEV). PostgreSQL local o Supabase.
- Repos web y API, variables de entorno, migraciones y ejecución.
- No incluye despliegue productivo ni documentación de interfaz.

Importancia de un entorno configurado

- Asegura estabilidad y reproducibilidad.
- Facilita colaboración y pruebas controladas.
- Reduce errores por diferencias entre equipos.

Requisitos del sistema

Hardware (sugerido)

- CPU 2 núcleos, RAM 8 GB (ideal 16 GB)
- 20 GB libres en disco
- Internet estable

Software

- Windows 10/11
- Node.js 18+ y npm
- Git
- Docker Desktop + Docker Compose (opcional recomendado)
- PostgreSQL 14+ o Supabase
- VS Code

Instalación de herramientas

Node.js y Git

Instalar desde los sitios oficiales y verificar en terminal:

```
node -v
npm -v
git --version
```

PostgreSQL (opción local)

Instalar y crear usuario/base de datos:

```
sudo -u postgres psql -c "CREATE USER pozinox_user WITH PASSWORD 'tu_clave';"
sudo -u postgres psql -c "CREATE DATABASE pozinox OWNER pozinox_user;"
```

Supabase (opción nube)

- Crear proyecto y copiar la cadena de conexión de Postgres.
- Usar esos datos en el archivo .env de la API.

Clonación del repositorio

```
git clone <repo-api>
cd pozinox-api

git clone <repo-web>
cd pozinox-web
```

Variables de entorno (.env)

Crear un archivo .env en cada repositorio con las siguientes claves mínimas.

API (.env):

```
APP_PORT=3000
DB_HOST=localhost
DB_PORT=5432
DB_NAME=pozinox
DB_USER=pozinox_user
DB_PASSWORD=tu_clave
JWT_SECRET=cadena_larga_segura
MP_ACCESS_TOKEN=APP_USR_xxxx
MP_PUBLIC_KEY=TEST_xxxx
SMTP_HOST=smtp.tu-dominio.cl
SMTP_PORT=587
SMTP_USER=notificaciones@pozinox.cl
SMTP_PASS=clave
```

WEB (.env):

```
VITE_API_URL=http://localhost:3000
```

Instalación de dependencias y migraciones

```
# API
cd pozinox-api
npm install
npm run migrate && npm run seed

# WEB
cd ../pozinox-web
npm install
```

Ejecución y desarrollo

Sin Docker

```
# API
cd pozinox-api
npm run start

# WEB
cd ../pozinox-web
npm run dev  # o build + preview
```

Con Docker (opcional)

```
docker compose up -d
docker compose exec api npm run migrate
docker compose exec api npm run seed
```

Estructura general del proyecto

```
pozinox-api/
  src/ (controladores, servicios, modelos)
  migrations/
pozinox-web/
  src/ (vistas, componentes)
  public/
```

Consideraciones técnicas

- Ramas por feature y PR con revisión.
- CORS solo para orígenes definidos.
- RBAC: CLIENTE / TRABAJADOR / ADMIN en endpoints.
- Logs y auditoría para movimientos de inventario.

Conclusión

Con estos pasos el equipo puede levantar POZINOX rápidamente, probar funcionalidades clave y continuar con el desarrollo.