

Documentación del Proyecto: Sistema de Inventario

Arquitectura General

El proyecto está estructurado como una aplicación web de inventario simple, diseñada para pequeños y medianos negocios. Tiene un enfoque en facilidad de uso, portabilidad y rapidez de despliegue.

La arquitectura se basa en tres componentes principales:

- 1. Frontend (HTML/CSS + JS)
- 2. Backend (Node.js con Express)
- 3. Base de Datos (SQLite)

Tecnologías Utilizadas

Componente	Tecnología	Descripción
Backend	Node.js + Express	Framework ligero para crear APIs REST
Base de Datos	SQLite	Base de datos local sin servidor
Gestor de Paquetes	npm	Manejo de dependencias de Node.js
Autenticación	Básica	Validación de usuario contra la base de datos
Frontend	HTML & CSS	Interfaz visual servida desde el backend

Estructura de Carpetas

...

```
inventario-app/  
├── backend/  
│   ├── server.js    # Archivo principal del servidor Express  
│   └── db.js        # Archivo de conexión a la base de datos SQLite  
├── inventario.db    # Base de datos SQLite local  
├── package.json     # Definición de dependencias y scripts npm  
├── node_modules/    # Dependencias instaladas con npm  
└── ...
```

Sistema de Login

El sistema de autenticación es básico. El usuario y la contraseña son verificados directamente desde la base de datos SQLite. En versiones iniciales se utilizaba un usuario estático ('admin' / 'admin')

Funcionalidades Principales

- Iniciar sesión (autenticación simple)
- Alta, edición y eliminación de productos
- Registro manual o por lectura de código QR
- Persistencia de datos con base de datos local (.db)
- Posibilidad de acceder desde dispositivos móviles con cámara

Requisitos Funcionales

- El sistema debe permitir iniciar sesión mediante un formulario de login.
- El usuario debe poder registrar productos manualmente.
- El sistema debe permitir editar o eliminar productos existentes.
- Debe poder registrar productos escaneando un código QR.
- El sistema debe mostrar todos los productos en un catálogo.

Requisitos No Funcionales

- La aplicación debe ejecutarse en entornos locales con pocos recursos.
- El tiempo de respuesta del servidor debe ser menor a 2 segundos.
- No requiere conexión a internet para funcionar.
- La base de datos debe ser fácilmente portable (archivo único .db).
- El sistema debe ser fácil de instalar y mantener.