

Inventory AI Manager

Sistema de gestión de inventario con asistente de IA integrado

Angular 20

Node.js + Express

PostgreSQL

IA Generativa

TDD

¿Qué es Inventory AI Manager?

Aplicación web fullstack que permite a empresas gestionar su inventario distribuido en múltiples almacenes, con un asistente de IA integrado que responde preguntas sobre el stock en lenguaje natural.

Gestión de Almacenes

Crea, edita y organiza múltiples almacenes. Visualiza el inventario de cada uno con alertas de stock bajo.

Catálogo de Productos

CRUD completo con SKU único por almacén, control de stock mínimo y filtrado en tiempo real.

Chat con IA

Sidebar siempre visible con asistente inteligente que conoce todo tu inventario y responde en lenguaje natural.

Stack Tecnológico

FRONTEND

- Angular 20 — Standalone Components + Signals
- TypeScript Strict Mode
- PrimeNG 21 — Componentes UI
- Tailwind CSS v4 — Estilos utilitarios
- ngx-translate — Internacionalización ES/EN

BACKEND

- Node.js + Express 5 — API REST
- TypeScript — Tipado estricto
- Prisma ORM — Acceso a base de datos
- JWT — Autenticación segura
- Rate Limiting — Protección de endpoints

DATOS & IA

- PostgreSQL 18 — Base de datos relacional
- Anthropic Claude — Asistente de IA
- Prisma Migrations — Control de esquema
- Context injection — Inventario en cada llamada

Arquitectura del Sistema



Flujo del Chat IA

1. Usuario envía mensaje
2. Backend construye contexto de inventario
3. Anthropic Claude genera respuesta
4. Sidebar muestra respuesta + enlaces

El Asistente de IA — Pieza Central del Proyecto

¿Qué puede hacer el asistente?

- Consultar stock de cualquier producto
- Identificar productos con stock bajo o agotado
- Comparar inventario entre almacenes
- Navegar directamente a un producto desde el chat
- Historial de conversación persistente
- Siempre visible en el sidebar derecho

Implementación técnica

Modelo:	Anthropic Claude Haiku
Contexto:	Inventario completo en cada llamada
Historial:	Signal con conversationHistory[]
Respuesta:	JSON con texto + productLink opcional
Rate limit:	20 req/min por usuario (express-rate-limit)
API key:	Solo en backend, nunca expuesta al cliente

Desarrollo Guiado por Tests (TDD)



280

Tests Frontend

120

Tests Backend

19

Tests E2E

419

Tests Totales

Calidad y Buenas Prácticas

Husky + Git Hooks

- Pre-commit: lint + typecheck
- Pre-push: tests + build
- Código roto no llega al repositorio

ESLint + SonarJS

- Reglas estrictas en frontend y backend
- Detección de code smells
- 0 errores de lint en main

TypeScript Strict

- noImplicitAny
- strictNullChecks
- Tipos compartidos frontend/backend

Git Flow

- Una rama por feature/step
- Pull Requests con revisión
- Historial limpio y trazable

Prettier

- Formato consistente
- Integrado en pre-commit
- Mismas reglas en todo el proyecto

Playwright E2E

- Page Object Pattern
- 19 tests de flujo completo
- Auth, CRUD, Chat IA

Conclusiones



Aplicación fullstack completa y desplegada en producción, accesible desde cualquier navegador



419 tests automatizados garantizan la estabilidad del código en cada cambio



Integración de IA generativa real que aporta valor diferencial al sistema de inventario



Arquitectura moderna con Angular 20 Signals, zoneless, standalone components y TypeScript strict



Pipeline de calidad completo: TDD + ESLint + Husky + CD automático en Railway y Vercel