

# DOCUMENTACIÓN DEL DESARROLLO DEL SOFTWARE

## 1. DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema es una aplicación web para la gestión del inventario y ventas de la botica **NovaSalud**, permitiendo llevar el control digital de los productos, ventas, usuarios y alertas de stock bajo.

Permite:

- Registrar productos y stock.
- Registrar ventas en tiempo real.
- Actualizar inventario automáticamente.
- Generar alertas por productos agotados.
- Asignar roles (admin / vendedor).
- Consultar reportes de ventas e inventario.

Su objetivo principal es **automatizar** procesos internos, reducir errores y mejorar la atención al cliente.

---

## 2. ARQUITECTURA DEL SOFTWARE

Se utiliza una arquitectura **Fullstack basada en componentes independientes**.

### Frontend

- Framework: **React.js**
- Estilos: **TailwindCSS**
- Características:
  - Interfaz moderna y rápida
  - Formularios dinámicos
  - Componentes reutilizables
  - Panel administrativo e interfaz de ventas

## Backend

- Framework: **Node.js + Express**
- Funciones:
  - API REST
  - Gestión de productos, ventas y usuarios
  - Generación de reportes
  - Seguridad con JWT

## Base de Datos

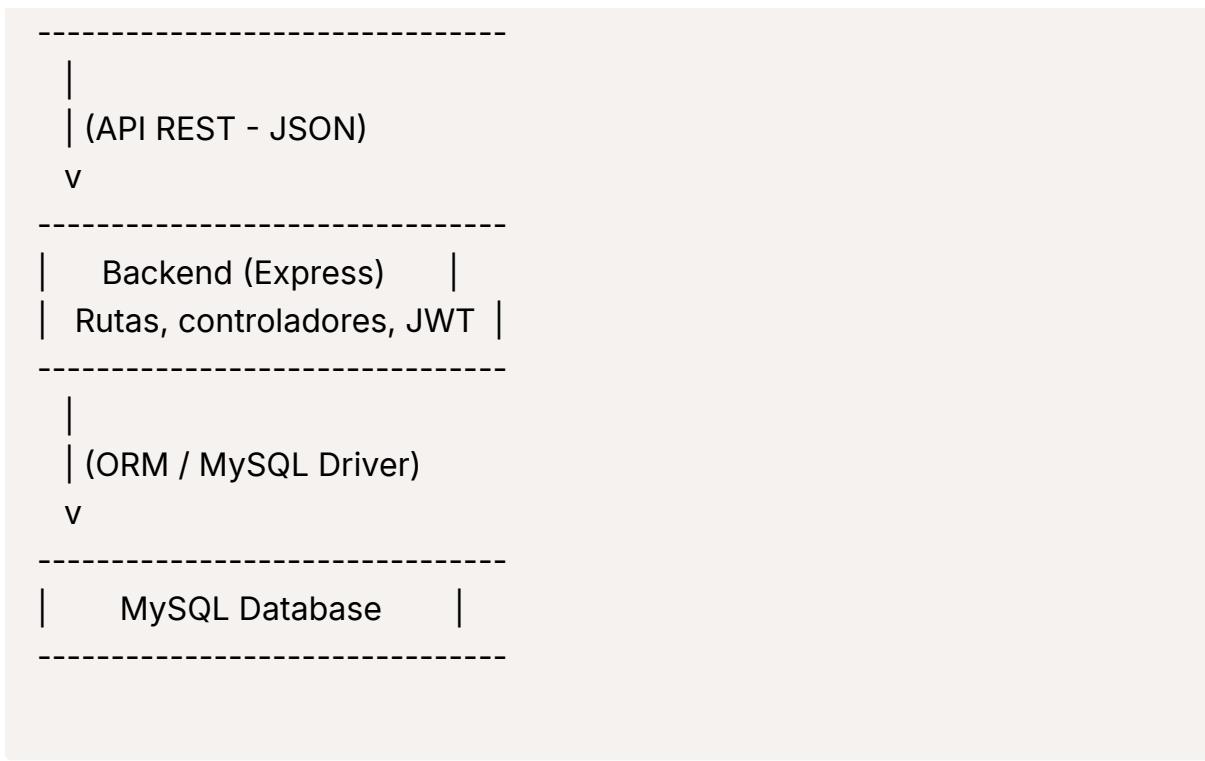
- Motor: **MySQL**
- Tablas:
  - `productos`
  - `usuarios`
  - `ventas`
  - `detalle_ventas`
  - `movimientos_stock`

## Infraestructura

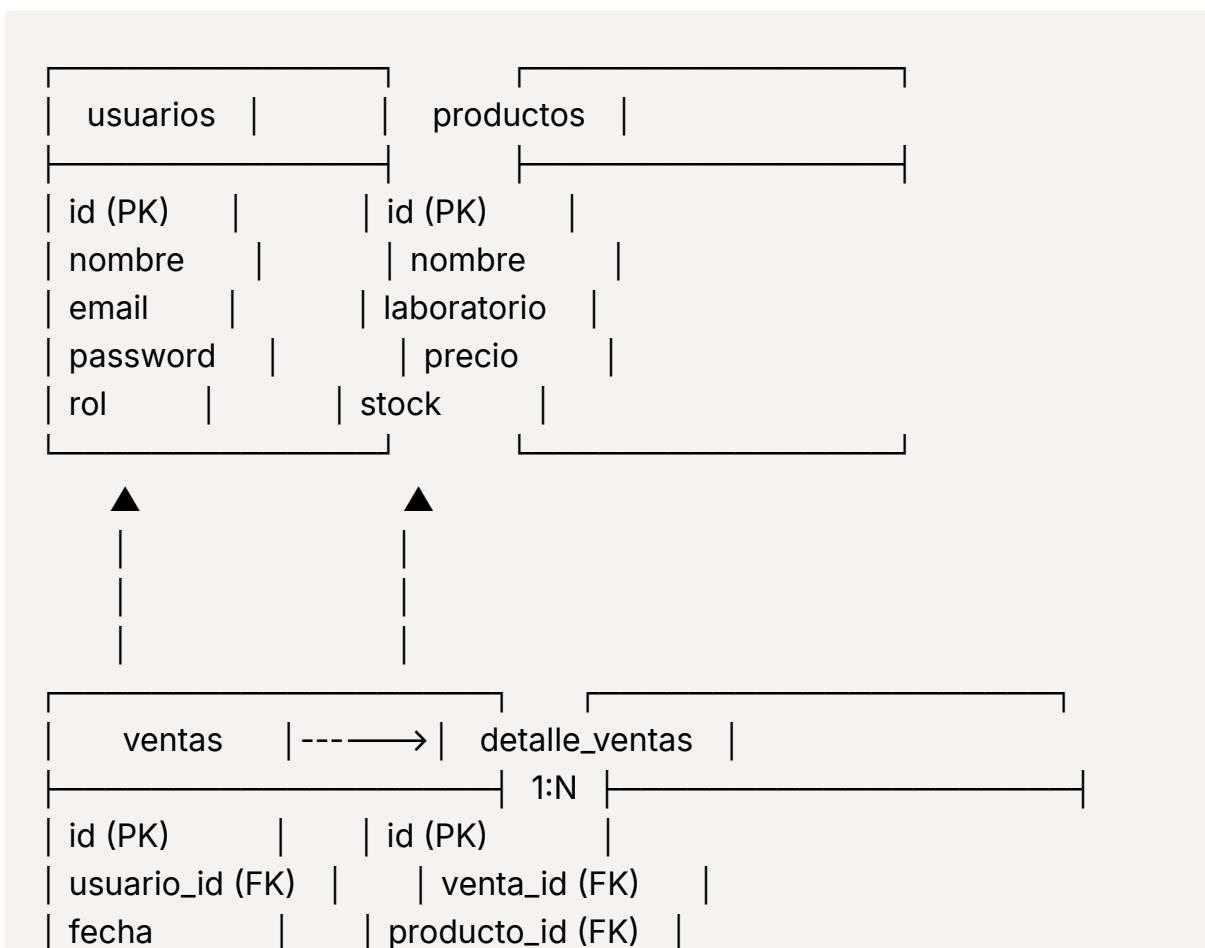
- Backend → Render
- Frontend → Vercel
- Base de datos → MySQL Cloud o local

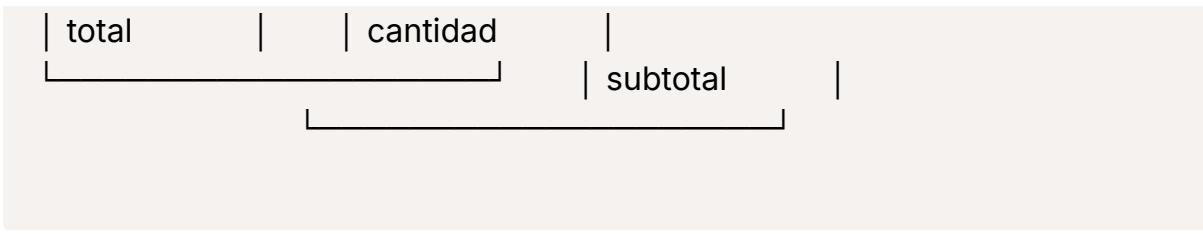
# 3. DIAGRAMA DE ARQUITECTURA





## 4. DIAGRAMA DE BASE DE DATOS (ERD)





## 5. CASOS DE USO (USE CASE)

### Actores

- Administrador
- Vendedor

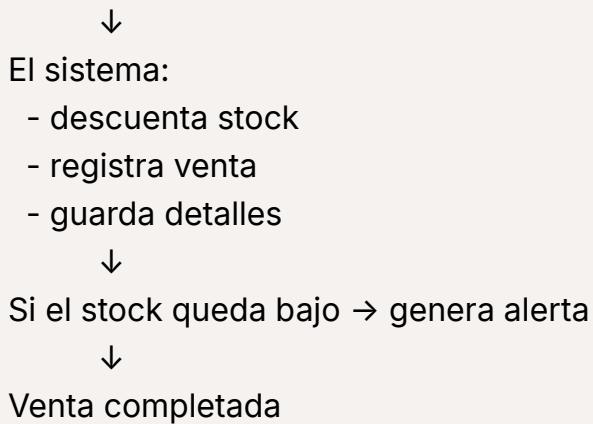
### Casos de Uso Principales

1. Registrar un producto
2. Editar stock
3. Registrar venta
4. Buscar producto
5. Generar alerta por stock bajo
6. Crear usuario
7. Iniciar sesión
8. Ver reportes de ventas

## 6. FLUJO PRINCIPAL DEL SISTEMA

### Flujo: Registrar Venta

Vendedor inicia sesión  
↓  
Accede al módulo de ventas  
↓  
Busca el producto por nombre/código  
↓  
Selecciona cantidad



## 7. TECNOLOGÍAS Y JUSTIFICACIÓN

### React

- Ágil, modular, fácil de mantener.
- Experiencia moderna para el usuario.

### Node.js + Express

- Permite desarrollar APIs rápidas y escalables.
- Utiliza JavaScript tanto en frontend como backend.

### MySQL

- Robusto para inventarios.
- Fácil de implementar y administrar.

### JWT

- Seguridad moderna, estándar en aplicaciones web.

## 8. ESTRUCTURA DE CARPETAS DEL PROYECTO

### Backend

```
/backend  
  /src  
    /routes  
    /controllers  
    /models  
    /middleware  
    /config  
  index.js
```

## Frontend

```
/frontend  
  /src  
    /components  
    /pages  
    /services  
    /context
```

# 9. ENDPOINTS DEL BACKEND (API REST)

## /auth

- POST `/login`
- POST `/register`

## /productos

- GET `/`
- POST `/`
- PUT `/:id`
- DELETE `/:id`

## /ventas

- POST /
  - GET /historial
- 

## 10. ROLES Y PERMISOS

### Administrador

- CRUD productos
- CRUD usuarios
- Ver reportes
- Ver historial de ventas

### Vendedor

- Registrar ventas
  - Consultar stock
  - Generar boletas/recibos
- 

## 11. NORMAS TÉCNICAS Y BUENAS PRÁCTICAS

- Validación de datos en backend.
- Normalización de tablas (1FN–3FN).
- Buenas prácticas de UI/UX.
- Código limpio (Clean Code).
- Variables de entorno en .env .
- Pruebas de endpoints con Postman.
- Uso responsable del hosting.