



INFORME DE PROYECTO PARCIAL 2

Sistema de Gestión de Inventario y
Órdenes de Compra

Universidad de las Fuerzas Armadas ESPE

Arico Cesar, Herrera Alan, Suquillo Fernando

17 de noviembre de 2025, Sangolquí - Pichincha

Descripción General del Sistema

Este sistema apoya a empresas de importación y distribución nacional, gestionando productos, proveedores, inventario en múltiples bodegas y el ciclo completo de órdenes de compra.



Catálogo de Productos

Administración integrada de productos.



Gestión de Proveedores

Control de la base de datos de proveedores.



Inventario Multibodega

Control de stock en diversas ubicaciones.

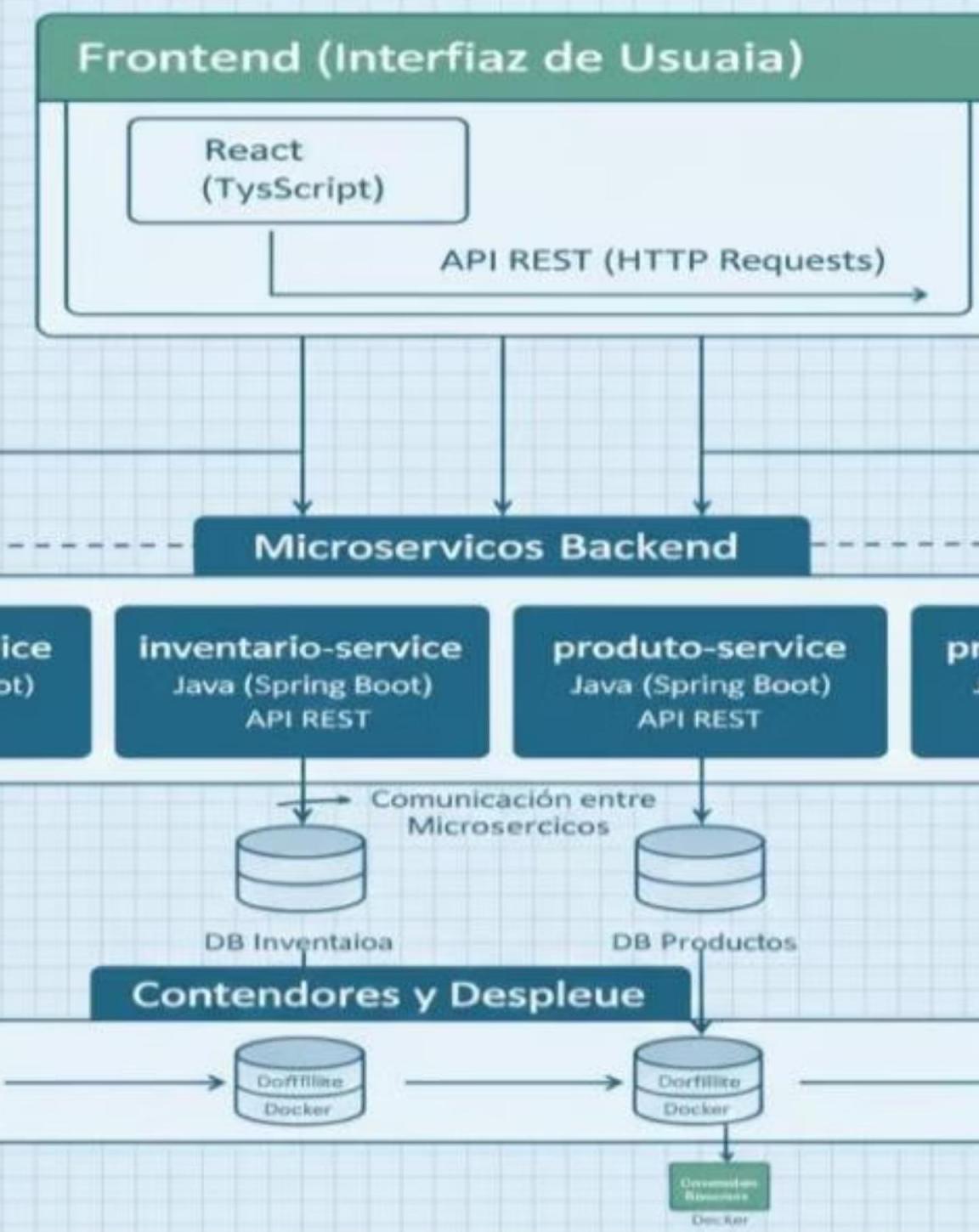


Órdenes de Compra

Ciclo completo desde creación hasta recepción.

Facilita la creación, aprobación y recepción de órdenes, actualizando el stock automáticamente. Genera reportes de abastecimiento, como productos críticos e historial de compras.

Diagrama de Arquitectura del Sistema

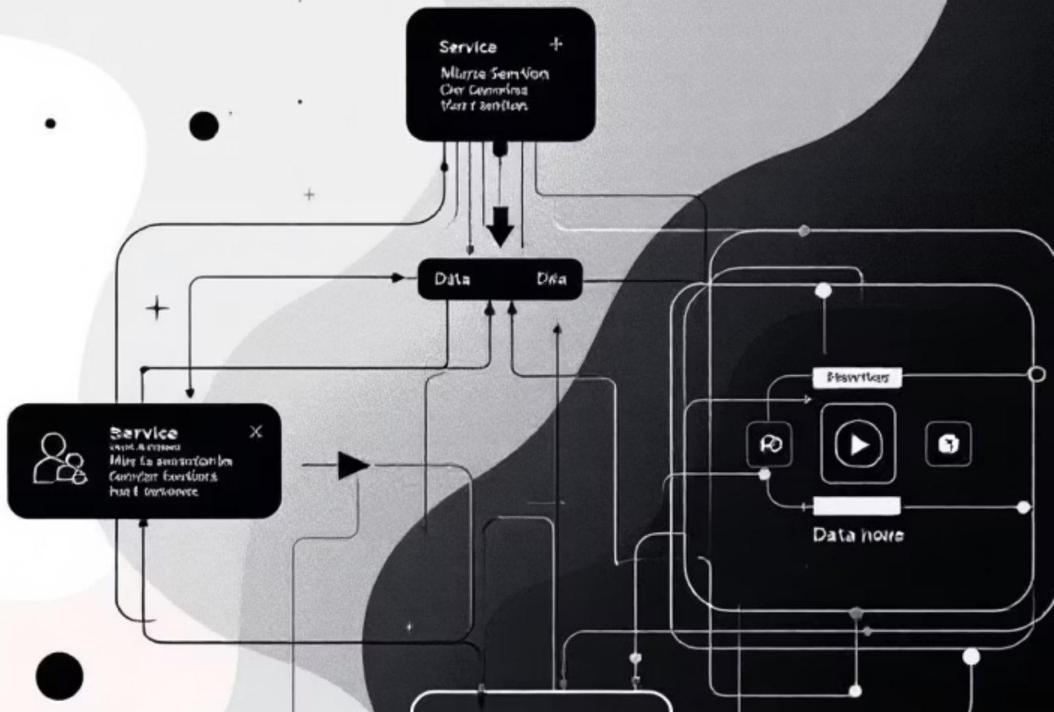


Arquitectura General del Sistema

El sistema se basa en microservicios independientes desarrollados con Java 17 y Spring Boot 3. Cada microservicio tiene su propia base de datos y se comunica mediante APIs RESTful (HTTP y JSON).

El sistema se ejecuta en un entorno contenerizado con Docker, asegurando consistencia entre entornos de desarrollo y despliegue.

Componentes Principales



Frontend (Interfaz de Usuario)

- Desarrollado en React con TypeScript.
- Permite la interacción del usuario con el sistema.
- Se comunica con microservicios vía APIs REST.

Persistencia de Datos

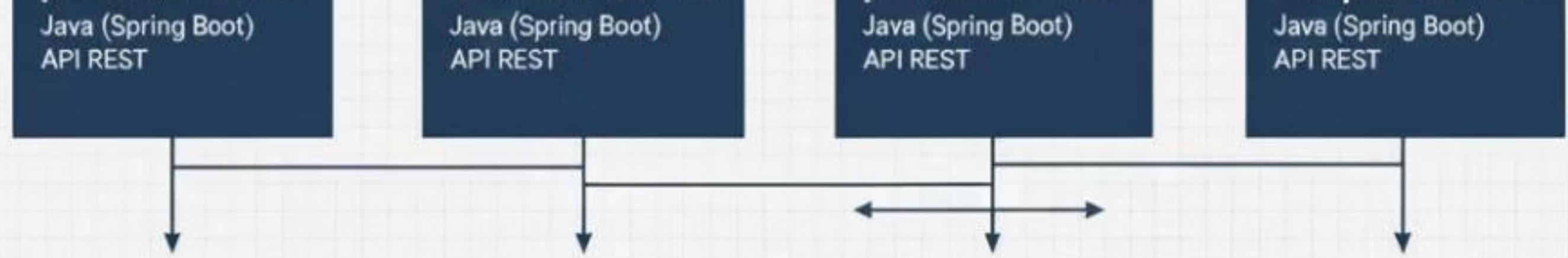
- Cada microservicio tiene su propia base de datos.
- Asegura independencia y encapsulamiento de datos.

Microservicios Backend

- Implementados con Java y Spring Boot.
- Principales:
 - compras-service
 - inventario-service
 - producto-service
 - proveedor-service
- Cada uno expone su API REST y gestiona su DB.

Contenedores y Despliegue

- Todos los servicios y el frontend tienen Dockerfile.
- Facilita el despliegue en contenedores.



Descripción de Microservicios

1

Servicio de Productos

Gestiona la información de productos: creación, actualización, consulta y eliminación. Valida existencia y detalles para otros módulos.

2

Servicio de Proveedores

Encargado de la gestión integral de proveedores: registro, actualización de información, consulta y eliminación.

3

Servicio de Inventory

Controla y actualiza existencias en almacenes, registra movimientos de entrada/salida y consulta stock disponible.

4

Servicio de Órdenes de Compra

Gestiona el ciclo de vida de las órdenes de compra: creación, historial, actualización de estados y asociación de productos/proveedores.

Arquitectura del Frontend

Desarrollado con React y TypeScript, el frontend tiene una arquitectura modular y basada en componentes reutilizables para facilitar el mantenimiento y la escalabilidad.

Componentes de Presentación

Muestran información y la interfaz gráfica (Layout, tablas, formularios).

Páginas

Vistas principales del sistema (Dashboard, Inventario, Productos, Proveedores, Órdenes de Compra).

Servicios

Encapsulan la lógica de comunicación con microservicios backend (APIs REST).

Tipos y Utilidades

Definición de tipos TypeScript y funciones utilitarias para manejo de datos.

Router

Arquitectura del Frontend (React + TypeScript)

Path: / (React + TypeScript)

Frontend (React + TypeScript)

Arquitectura Modular por Componentes

Páginas (Vistas Principales)

Path: / → DashboardPage → InventardPage

en componente

Componentes de

Presentación

out

tas de Datos

hulamias

Tipos y

Utilidades

Servicios (API)

productoServ

provedorServ

inventarioSe

ordenCompr

Path: / → DashboardPage → InventardPage

en componente

Componentes de

Presentación

out

tas de Datos

hulamias

Tipos y

Utilidades

usán + usán

Peticiones HTTP (API) a Microservicios Backend



Decisiones de Diseño Clave

Microservicios

Mayor escalabilidad, mantenibilidad y flexibilidad.
Permite evolución independiente y mejora la tolerancia a fallos.

División del Dominio

Separación en servicios (productos, proveedores, inventario, órdenes de compra) para responsabilidades claras y menor acoplamiento.

Tecnologías

Java + Spring Boot (backend), React + TypeScript (frontend), Docker (contenerización) y APIs RESTful (comunicación) por robustez y escalabilidad.

Mejoras Futuras y Conclusión

Mejoras Futuras

- Autenticación y autorización.
- Integración con sistemas externos (ERP, facturación).
- Monitoreo y logging avanzado.
- Automatización de pruebas y CI/CD.
- Optimización de UI/UX y accesibilidad.
- Escalabilidad y alta disponibilidad.
- Notificaciones y alertas.

Conclusión

El sistema de gestión de inventario es una solución moderna, escalable y robusta. La arquitectura de microservicios con Java, Spring Boot, React y Docker ofrece flexibilidad y facilidad de mantenimiento.

Las buenas prácticas en validación, manejo de errores y experiencia de usuario aseguran una solución confiable y satisfactoria. La estructura modular y contenerización facilitan futuras mejoras y escalabilidad.

