Product Requirements Document (PRD)

Sistema de Gestión de Inventario y Almacén

1. Visión del Producto

Sistema modular para gestionar inventario de productos, control de stock, órdenes de compra/venta y proveedores, utilizando Spring Boot + Spring Modulith.

2. Objetivos

- Controlar stock de productos en tiempo real
- Gestionar proveedores y sus productos
- Procesar órdenes de compra y venta
- Generar alertas de stock bajo
- Mantener historial de movimientos de inventario

3. Módulos Principales (4 módulos básicos)

3.1 Módulo `inventory`

- **Responsabilidad:** Gestión de productos y control de stock
- **Funcionalidades:**
  - CRUD de productos (nombre, descripción, SKU, precio)
  - Control de stock actual
  - Actualizaciones de inventario (entrada/salida)
  - Alertas de stock mínimo

3.2 Módulo `suppliers`

- **Responsabilidad:** Gestión de proveedores
- **Funcionalidades:**
  - CRUD de proveedores (nombre, contacto, dirección)
  - Catálogo de productos por proveedor
  - Precios de compra por proveedor

3.3 Módulo `orders`

- **Responsabilidad:** Gestión de órdenes de compra y venta
- **Funcionalidades:**
  - Órdenes de compra a proveedores
  - Órdenes de venta a clientes
  - Estados de órdenes (pendiente, procesada, completada)
  - Recepción de mercancía

3.4 Módulo `notifications`

- **Responsabilidad:** Sistema de notificaciones

- **Funcionalidades:**
  - Alertas de stock bajo
  - Notificaciones de órdenes procesadas
  - Reportes de movimientos

4. Eventos del Sistema

- `ProductCreated` - Nuevo producto agregado
- `StockUpdated` - Cambio en cantidad de stock
- `LowStockAlert` - Stock por debajo del mínimo
- `PurchaseOrderPlaced` - Orden de compra creada
- `SaleOrderPlaced` - Orden de venta creada
- `OrderCompleted` - Orden procesada completamente

5. APIs REST Principales

```
/api/products
  POST   - Crear producto
  GET    - Listar productos
  PUT    - Actualizar producto
  DELETE - Eliminar producto

/api/inventory
  POST /update-stock - Actualizar stock
  GET  /low-stock    - Productos con stock bajo

/api/suppliers
  POST   - Crear proveedor
  GET    - Listar proveedores
  PUT    - Actualizar proveedor

/api/orders
  POST /purchase - Crear orden de compra
  POST /sale     - Crear orden de venta
  PUT  /complete - Completar orden
```

6. Modelo de Datos Básico

Product: id, sku, name, description, price, currentStock, minStock, supplierId

Supplier: id, name, email, phone, address

Order: id, type(PURCHASE/SALE), status, totalAmount, createdDate, supplierId

OrderItem: id, orderId, productId, quantity, unitPrice

7. Stack Tecnológico

- Spring Boot 3.5+
- Spring Modulith 1.4+
- Java 21
- MySQL 8.0
- JPA/Hibernate
- Lombok
- SpringDoc OpenAPI
- Maven

8. Criterios de Aceptación

- Cada módulo debe ser independiente
- Comunicación entre módulos solo por eventos
- APIs REST documentadas con OpenAPI
- Base de datos normalizada
- Validaciones de negocio implementadas
- Manejo de errores consistente

9. Casos de Uso Principales

9.1 Gestión de Productos

1. **Crear Producto:** Administrador agrega nuevo producto al catálogo
2. **Actualizar Stock:** Sistema actualiza inventario tras compra/venta
3. **Alerta Stock Bajo:** Notificación cuando producto está por agotarse

9.2 Gestión de Proveedores

1. **Registrar Proveedor:** Agregar nuevo proveedor al sistema
2. **Asociar Productos:** Vincular productos con sus proveedores

9.3 Gestión de Órdenes

1. **Orden de Compra:** Solicitar mercancía a proveedor
2. **Orden de Venta:** Procesar venta a cliente
3. **Recepción:** Actualizar inventario al recibir mercancía

10. Fase 1 - MVP (Minimum Viable Product)

- Módulo inventory con CRUD básico de productos
- Módulo suppliers con gestión básica
- Eventos básicos (ProductCreated, StockUpdated)
- APIs REST fundamentales
- Base de datos con tablas principales

11. Fase 2 - Funcionalidades Avanzadas

- Módulo orders completo
- Sistema de notificaciones
- Reportes de inventario
- Validaciones de negocio avanzadas
- Pruebas unitarias e integración