

Documentación Técnica - Sistema de Inventarios v02

1. Descripción General

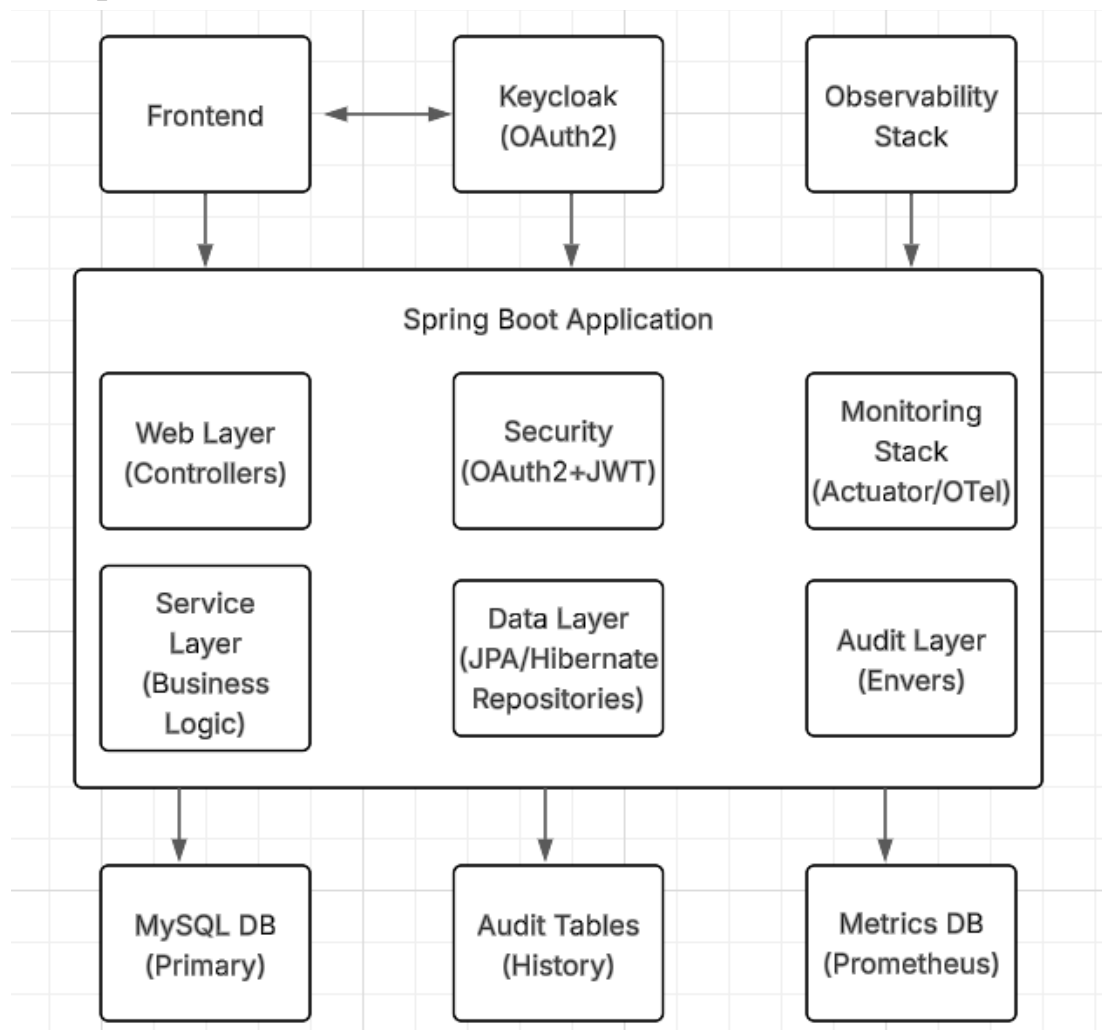
Propósito

Sistema de gestión de inventarios desarrollado con Spring Boot que permite el control de productos, stock, categorías y generación de reportes con funcionalidades de auditoría completa.

Características Principales

- Gestión completa de productos e inventario
- Sistema de autenticación OAuth2 con Keycloak
- API REST con diferentes niveles de autorización
- Auditoría completa con Hibernate Envers
- Monitoreo y métricas con Prometheus/Grafana
- Trazabilidad distribuida con OpenTelemetry
- Migraciones de base de datos con Flyway

2. Arquitectura del Sistema



Componentes Principales

Capa de Presentación

Controllers Web: Manejo de vistas Thymeleaf

API REST Controllers: Endpoints para integración externa

Security Filters: JWT y OAuth2 authentication

Capa de Negocio

ProductService: Lógica de gestión de productos

ProductHistoryService: Gestión de auditoría

Validation: Validación de datos de entrada

Capa de Datos

JPA Repositories: Acceso a datos

Entity Models: Mapeo de entidades

Flyway Migrations: Control de versiones de BD

3. Tecnologías y Dependencias

Framework Principal

- Spring Boot 3.5.3
- Java 21
- Maven para gestión de dependencias

Dependencias Principales

Persistencia

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.33</version>
</dependency>
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
```

Seguridad

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-oauth2-client</artifactId>
</dependency>
<dependency>
  <groupId>io.jsonwebtoken</groupId>
  <artifactId>jjwt-api</artifactId>
  <version>0.11.5</version>
</dependency>
```

Auditoría

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-envers</artifactId>
```

```
<version>6.6.4.Final</version>
</dependency>
```

Monitorio

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
  <groupId>io.opentelemetry.instrumentation</groupId>
  <artifactId>opentelemetry-spring-boot-starter</artifactId>
  <version>1.32.0-alpha</version>
</dependency>
<dependency>
  <groupId>io.micrometer</groupId>
  <artifactId>micrometer-registry-prometheus</artifactId>
</dependency>
```

4. Base de Datos

Esquema Principal

Tabla Products

```
CREATE TABLE products (  
    id BIGINT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    description VARCHAR(500),  
    category VARCHAR(70) NOT NULL,  
    price DECIMAL(10,2) NOT NULL,  
    stock INT NOT NULL DEFAULT 0,  
    minimum_stock INT DEFAULT 0,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
    PRIMARY KEY (id)  
);
```

Tablas de Auditoría

```
CREATE TABLE REVINFO (  
    REV INT NOT NULL AUTO_INCREMENT,  
    REVTSMP BIGINT NOT NULL,  
    username VARCHAR(255) NULL,  
    user_id BIGINT NULL,  
    PRIMARY KEY (REV)  
);
```

```
CREATE TABLE products_audit (  
    id BIGINT NOT NULL,  
    REV INT NOT NULL,  
    REVTYPE TINYINT NULL,  
    name VARCHAR(100) NULL,  
    description VARCHAR(500) NULL,  
    category VARCHAR(70) NULL,  
    price DECIMAL(10,2) NULL,  
    stock INT NULL,  
    minimum_stock INT NULL,  
    PRIMARY KEY (id, REV)  
);
```

Migraciones Flyway

Estructura de Versiones

V1: Esquema inicial de productos

V2: Adición de timestamps y índices

V3: Tablas de usuarios y API users

V4-V6: Configuración de auditoría con Envers

Ejemplo de Migración

sql-- V1__Initial_schema.sql

```
CREATE TABLE products (  
  id BIGINT NOT NULL AUTO_INCREMENT,  
  name VARCHAR(100) NOT NULL,  
  description VARCHAR(500),  
  category VARCHAR(70) NOT NULL,  
  price DECIMAL(10,2) NOT NULL,  
  stock INT NOT NULL,  
  minimum_stock INT,  
  PRIMARY KEY (id)  
);
```

5. Auditoría y Trazabilidad

Hibernate Envers Configuration

Configuración Principal

```
property spring.jpa.properties.org.hibernate.envers.revision_field_name=REV
spring.jpa.properties.org.hibernate.envers.revision_type_field_name=REVTYPE
spring.jpa.properties.org.hibernate.envers.audit_table_suffix=_audit
spring.jpa.properties.org.hibernate.envers.store_data_at_delete=true
```

Entidad Auditada

```
java@Entity
@Table(name = "products")
@Audited
public class Product {
    // Campos de la entidad
}
```

Servicios de Auditoría

ProductHistoryService

```
javapublic List<Product> getProductHistory(Long productId)
public Product getProductAtRevision(Long productId, Long revisionId)
public List<Number> getProductRevisionsList(Long productId)
```

Endpoints de Auditoría

GET /api/products/{id}/history - Historial completo

GET /api/products/{id}/history/{revisionId} - Estado específico

GET /api/products/{id}/revisions - Lista de revisiones

6. Monitoreo y Observabilidad

Stack de Monitoreo

Componentes

- OpenTelemetry Collector: Recolección de trazas
- Jaeger: Visualización de trazas distribuidas
- Prometheus: Métricas y alertas
- Grafana: Dashboards y visualización

