

Food Store - Backend (Spring Boot)

Versión 1.0 — Proyecto académico basado en el diagrama UML proporcionado.

1. Introducción

Este documento describe la implementación del backend del sistema Food Store. Está desarrollado en Java utilizando Spring Boot e incluye el manejo de usuarios, productos, categorías, pedidos y detalles de pedido según el diagrama UML.

2. Tecnologías utilizadas

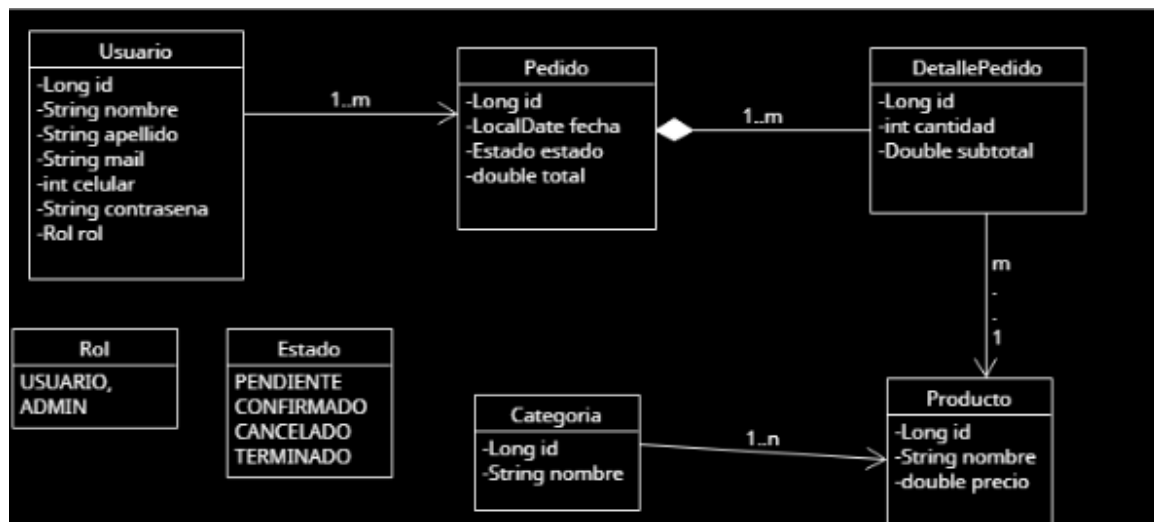
- Java 17/21
- Spring Boot (Web, Data JPA)
- PostgreSQL / H2 /MySQL
- Lombok
- Maven o Gradle
- SHA-256 para hash de contraseñas

3. Estructura de paquetes

com.example.foodstore

- └─ controller
- └─ entity
 - └─ dto
 - └─ mapper
- └─ repository
- └─ service
 - └─ impl

4. UML



5. Seguridad

Las contraseñas se almacenan de manera segura mediante SHA-256 antes de guardarse en la base de datos. Esto se implementa en el AuthService, utilizando la clase utilitaria Sha256Util.

Ejemplo: Sha256Util.java

```
public static String hash(String input) {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] dig = md.digest(input.getBytes(StandardCharsets.UTF_8));
    return bytesToHex(dig);
}
```

6. Repositorios

Se utilizan interfaces que extienden JpaRepository, permitiendo CRUD automático. Ejemplo:

```
public interface UserRepository extends JpaRepository<User, Long> {
    Optional<User> findByEmail(String email);
    boolean existsByEmail(String email);
}
```

7. Servicios

Los servicios implementan la lógica de negocio. Ejemplo AuthService para registrar y autenticar usuarios:

```
public User register(User user) {
    user.setPassword(Sha256Util.hash(user.getPassword()));
    return userRepository.save(user);
}
```

8. Controladores (Controllers)

Los controladores exponen endpoints REST para interactuar con el sistema. Ejemplo:

```
@RestController
@RequestMapping("/api/auth")
public class AuthController {
    @PostMapping("/register")
    public ResponseEntity<User> register(@RequestBody User user) {
        return ResponseEntity.status(201).body(authService.register(user));
    }
}
```

10. Base de datos

El esquema relacional se genera automáticamente con Hibernate según las relaciones JPA definidas. Las tablas se corresponden con las entidades Usuario, Pedido, DetallePedido, Producto y Categoría.