

# Projeto web services com Spring Boot e JPA / Hibernate

## Objetivos

- Criar projeto Spring Boot Java
- Implementar modelo de domínio
- Estruturar camadas lógicas: resource, service, repository
- Configurar banco de dados de teste (H2)
- Povoar o banco de dados
- CRUD - Create, Retrieve, Update, Delete
- Tratamento de exceções

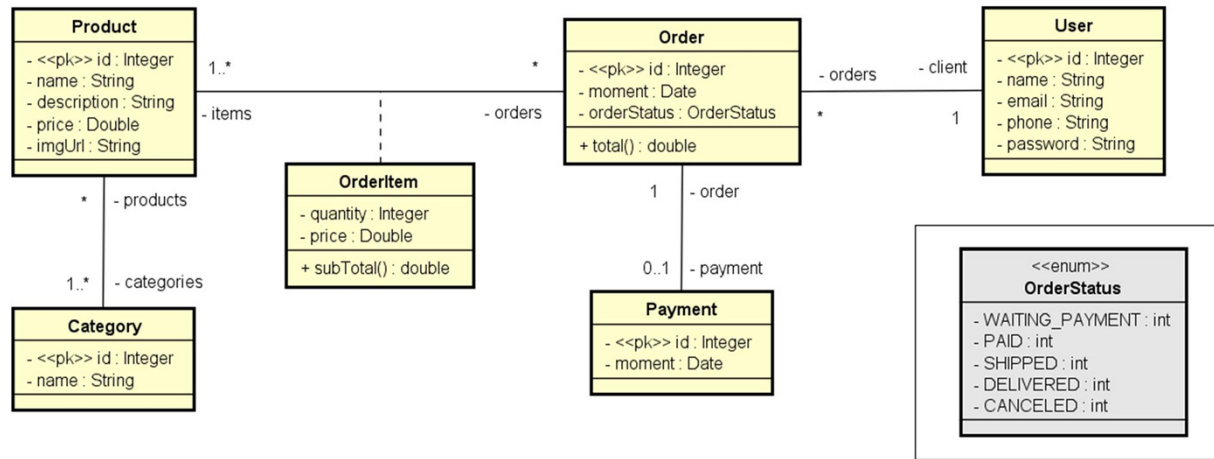
Github:



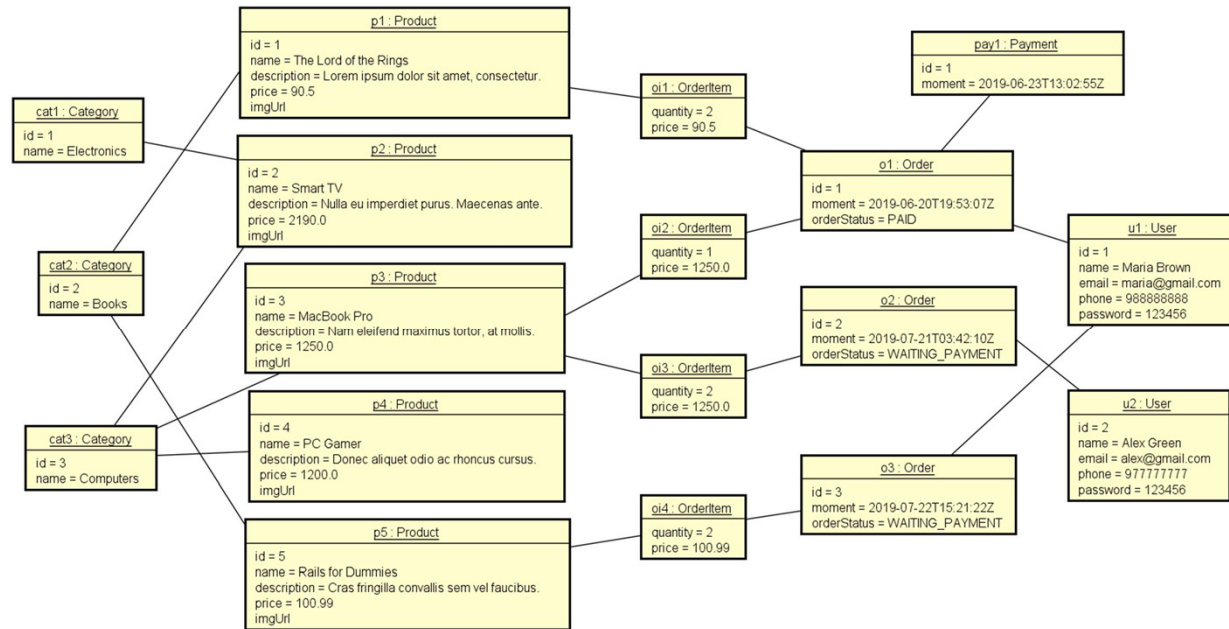
PostgreSQL



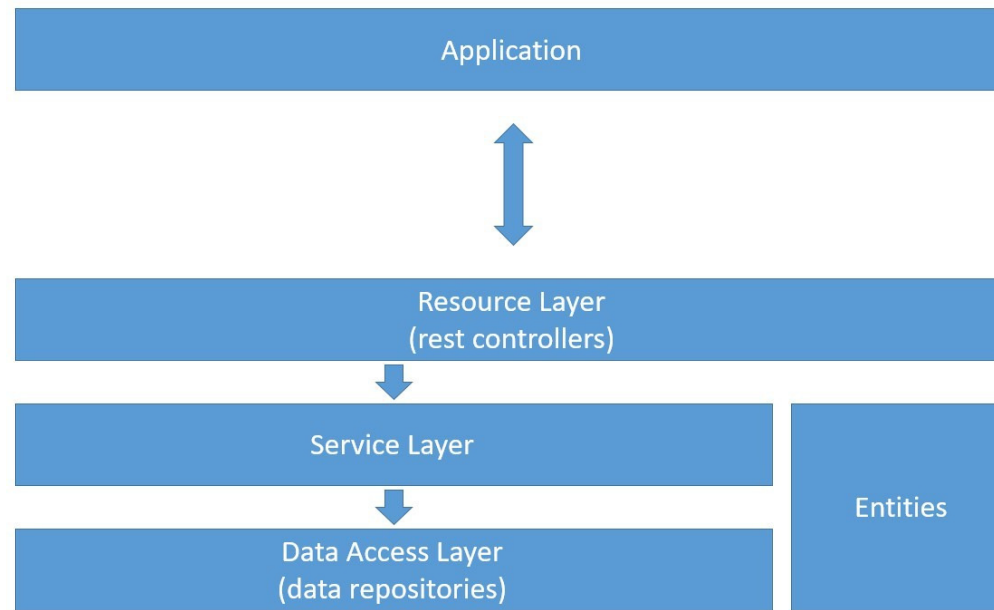
## Domain Model



## Domain Instance



## Logical Layers



## Project created

### Checklist:

- Spring Initializr
  - Maven
  - Java 17
- Packing JAR
- Dependencies: Spring Web

## User entity and resource

### Basic entity checklist:

- Basic attributes
- Associations (instantiate collections)
- Constructors
- Getters & Setters (collections: only get)
- hashCode & equals
- Serializable

## H2 database, test profile, JPA

### Checklist:

- JPA & H2 dependencies
- application.properties
- application-test.properties
- Entity: JPA mapping

### Dependencies:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-
    jpa</artifactId>
</dependency>

<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

### application.properties:

spring.profiles.active=test

spring.jpa.open-in-view=true

## application-test.properties:

```
# DATASOURCE
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.username=sa spring.datasource.password=

# H2 CLIENT
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console

# JPA, SQL
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.defer-datasource-initialization=true spring.jpa.show-
sql=true spring.jpa.properties.hibernate.format_sql=true
```

JPA repository, dependency injection, database seeding

### Checklist:

- UserRepository extends JpaRepository<User, Long>
- Configuration class for "test" profile
- @Autowired UserRepository
- Instantiate objects in memory
- Persist objects

### Objects:

```
User u1 = new User(null, "Maria Brown", "maria@gmail.com", "988888888", "123456");
User u2 = new User(null, "Alex Green", "alex@gmail.com", "977777777", "123456");
```

## Service layer, component registration