### Exercício 11.1 – Testes Unitários com JUnit

**Objetivo:** Demonstrar como implementar e executar testes unitários em um projeto Spring Boot utilizando JUnit, com Java 21 e a IDE VS Code.

### **Parte 1: Configuração do Projeto no Spring Initializr**

1. **Acessar o Spring Initializr:**
   * Vá ao site [Spring Initializr](https://start.spring.io/).
2. **Configurar o projeto:**
   * **Project:** Maven
   * **Language:** Java
   * **Spring Boot:** Última versão estável
   * **Group:** com.example
   * **Artifact:** demo
   * **Name:** demo
   * **Package Name:** com.example.demo
   * **Packaging:** Jar
   * **Java Version:** 21
3. **Adicionar dependências:**
   * **Spring Web:** Para criar APIs REST.
   * **Spring Boot Starter Test:** Para suporte a testes unitários com JUnit.
4. **Gerar o projeto:**
   * Clique em **Generate** para descarregar o projeto como um ficheiro ZIP.
5. **Abrir o projeto no VS Code:**
   * Extraia o ficheiro ZIP.
   * No VS Code, clique em **File > Open Folder** e selecione a pasta do projeto.

### **Parte 2: Estrutura do Projeto**

1. **Criar as pastas necessárias:**
   * No VS Code, crie as seguintes pastas em src/main/java/com/example/demo:
     + **entity** para a classe do modelo de dados.
     + **repository** para o repositório.
     + **service** para a lógica de negócios.
   * Em src/test/java/com/example/demo, crie:
     + **service** para os testes dos serviços.
2. **Estrutura do projeto:**

src/  
├── main/  
│ ├── java/  
│ │ └── com.example.demo/  
│ │ ├── DemoApplication.java  
│ │ ├── entity/  
│ │ ├── repository/  
│ │ └── service/  
├── test/  
│ └── java/  
│ └── com.example.demo/  
│ └── service/

### **Parte 3: Implementação do Modelo de Dados**

1. **Criar a classe Produto em entity:**

package com.example.demo.entity;  
  
public class Produto {  
  
 private Long id;  
 private String nome;  
 private Double preco;  
  
 public Produto() {}  
  
 public Produto(Long id, String nome, Double preco) {  
 this.id = id;  
 this.nome = nome;  
 this.preco = preco;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getNome() {  
 return nome;  
 }  
  
 public void setNome(String nome) {  
 this.nome = nome;  
 }  
  
 public Double getPreco() {  
 return preco;  
 }  
  
 public void setPreco(Double preco) {  
 this.preco = preco;  
 }  
}

### **Parte 4: Criar o Repositório**

1. **Criar a interface ProdutoRepository em repository:**

package com.example.demo.repository;  
  
import com.example.demo.entity.Produto;  
import java.util.Optional;  
  
public interface ProdutoRepository {  
 Optional<Produto> findById(Long id);  
}

### **Parte 5: Implementar o Serviço**

1. **Criar a classe ProdutoService em service:**

package com.example.demo.service;  
  
import com.example.demo.entity.Produto;  
import com.example.demo.repository.ProdutoRepository;  
import org.springframework.stereotype.Service;  
  
import java.util.Optional;  
  
@Service  
public class ProdutoService {  
  
 private final ProdutoRepository produtoRepository;  
  
 public ProdutoService(ProdutoRepository produtoRepository) {  
 this.produtoRepository = produtoRepository;  
 }  
  
 public Produto obterProdutoPorId(Long id) {  
 return produtoRepository.findById(id)  
 .orElseThrow(() -> new RuntimeException("Produto não encontrado"));  
 }  
}

### **Parte 6: Testes Unitários do Serviço com JUnit**

1. **Criar ProdutoServiceTest em test/service:**

package com.example.demo.service;  
  
import com.example.demo.entity.Produto;  
import com.example.demo.repository.ProdutoRepository;  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
import org.mockito.InjectMocks;  
import org.mockito.Mock;  
import org.mockito.MockitoAnnotations;  
  
import java.util.Optional;  
  
import static org.junit.jupiter.api.Assertions.assertEquals;  
import static org.mockito.Mockito.when;  
  
public class ProdutoServiceTest {  
  
 @Mock  
 private ProdutoRepository produtoRepository;  
  
 @InjectMocks  
 private ProdutoService produtoService;  
  
 @BeforeEach  
 public void setup() {  
 MockitoAnnotations.openMocks(this);  
 }  
  
 @Test  
 public void testObterProdutoPorId() {  
 Produto produto = new Produto(1L, "Produto Teste", 10.0);  
  
 when(produtoRepository.findById(1L)).thenReturn(Optional.of(produto));  
  
 Produto resultado = produtoService.obterProdutoPorId(1L);  
  
 assertEquals("Produto Teste", resultado.getNome());  
 assertEquals(10.0, resultado.getPreco());  
 }  
}

### **Parte 7: Executar os Testes**

1. **Executar testes no VS Code:**
   * Clique com o botão direito sobre o ficheiro ProdutoServiceTest.java e selecione **Run Test**.
2. **Confirmar resultados:**
   * Todos os testes deverão passar com sucesso.

Esta versão foca-se exclusivamente no uso de **JUnit** para testes unitários.