Guia de Testes Automatizados no Projeto Techseller

# 1. Introdução

Este guia tem como objetivo explicar e exemplificar a implementação de testes automatizados no projeto Java Spring Boot 'Techseller'. Os testes são fundamentais para garantir a qualidade do sistema, facilitar a manutenção e permitir crescimento seguro do projeto.

# 2. Mockito

O Mockito é uma biblioteca do Java utilizada para criar objetos 'mock' (falsos) que simulam o comportamento de objetos reais em testes. É muito útil para testar unidades de código de forma isolada, sem depender de banco de dados ou serviços externos.

Exemplo de uso:

when(userRepository.save(any(User.class))).thenReturn(user);

Isso simula a chamada ao método `save()` do repositório, sem precisar acessar o banco de verdade.

# 3. MockMvc

O MockMvc é uma ferramenta fornecida pelo Spring Boot para simular requisições HTTP sem subir um servidor. Com ele, podemos testar os controladores (controllers) do sistema como se estivéssemos usando Postman ou navegador.

Exemplo de uso:

mockMvc.perform(post("/salvar")  
 .contentType(MediaType.APPLICATION\_JSON)  
 .content(jsonDoProduto))  
 .andExpect(status().isOk());

# 4. Explicação das Annotations

* @WebMvcTest

→ Anotação do Spring que configura apenas os componentes web (controllers) para teste isolado.

* @MockBean

→ Cria um mock (falso) gerenciado pelo Spring para ser injetado no teste (usado em controladores).

* @InjectMocks

→ Anotação do Mockito que injeta os mocks criados em um objeto real (usado com serviços).

* @Mock

→ Cria um objeto mock simples do Mockito (sem integração com Spring).

* @Test

→ Anotação padrão do JUnit para indicar um método de teste.

* @BeforeEach

→ Método executado antes de cada teste (usado para configurar os mocks e preparar dados).

# 5. Estrutura Recomendada para Projetos em Crescimento

src/test/java/

├── controller/

│ └── ProdutoControllerTest.java

├── service/

│ └── UserServiceTest.java

└── ...

# 6. Considerações Finais

Com esses fundamentos, você e seu grupo poderão escrever novos testes sempre que novas funcionalidades forem adicionadas. Isso garante segurança nas mudanças e ajuda na documentação técnica do sistema.