Roteiro de Exercícios
Testes de Software com JUnit

Prof. Esp. Gilberto Falco Netto

12 de maio de 2025

Objetivo

Este roteiro tem como objetivo a prática de testes unitários com **JUnit 5**, utilizando exemplos progressivos de classes Java. Os alunos deverão implementar as classes de produção e seus respectivos testes automatizados.

Pré-requisitos

- Ter o **JUnit 5** configurado no ambiente de desenvolvimento (preferencialmente no IntelliJ ou Eclipse).
- Conhecimentos básicos de programação em Java.
- Conhecimento introdutório sobre testes unitários.

Exercícios

Exercício 1 – Teste de Métodos Simples

Implemente uma classe `Calculadora` com os métodos:

- `int somar(int a, int b);`
- `int subtrair(int a, int b);`
- `int multiplicar(int a, int b);`
- `double dividir(int a, int b);`

Tarefa: Escreva testes unitários para cada método usando `@Test`. Teste também a divisão por zero com `assertThrows`.

Exercício 2 – Testando Comparações e Condições

Crie a classe `NumeroUtil` com os métodos:

- `boolean ehPar(int n);`
- `boolean ehPrimo(int n);`

Tarefa: Escreva testes cobrindo diferentes entradas e condições.

Exercício 3 – Testando Classes com Estado

Crie a classe `CarrinhoDeCompras` com os métodos:

- `void adicionarProduto(String nome, double preco);`
- `void removerProduto(String nome);`
- `double calcularTotal();`

Tarefa: Teste o comportamento da adição, remoção e cálculo do total. Use `@BeforeEach` para criar um novo carrinho antes de cada teste.

Exercício 4 – Teste de Cobertura com Condições

Classe: `Desconto`

- Método: `double calcularDesconto(double valor, boolean vip);`

Regra de Negócio:

- 10% de desconto para compras acima de R$200,00;
- +5% para clientes VIP.

Tarefa: Crie testes cobrindo todas as possibilidades (cliente comum e VIP, valores acima e abaixo do limite).

Exercício 5 – Teste com Coleções

Crie a classe `Aluno` e a classe `Turma`, com método:

- `List<Aluno> getAprovados();`

Tarefa: Teste se o método retorna corretamente os alunos aprovados.

Exercício 6 – Testando Exceções

Classe: `ContaBancaria`

- Método: `void sacar(double valor);`

Tarefa: Verifique se uma exceção `IllegalArgumentException` é lançada ao tentar sacar mais que o saldo.

Exercício 7 – Testes Parametrizados

Implemente o método `int fatorial(int n);`.

Tarefa: Use `@ParameterizedTest` para testar com vários valores de entrada, incluindo 0, 1, 3, 5 e 10.

Entrega

Os alunos devem entregar os seguintes arquivos:

- Códigos-fonte das classes implementadas.
- Códigos de teste com cobertura adequada.