

Lista de Exercícios de Teste de Software com JUnit

Prof. Esp. Gilberto Falco Netto

12 de maio de 2025

Objetivo

Esta lista tem como objetivo proporcionar a prática de testes unitários utilizando a biblioteca JUnit 5. Os exercícios simulam cenários comuns de desenvolvimento de software e devem ser resolvidos com a criação de classes Java e seus respectivos testes automatizados.

Instruções

- Crie um projeto Java em sua IDE (como Eclipse ou IntelliJ).
- Configure a biblioteca JUnit 5 (via Maven, Gradle ou manualmente).
- Para cada exercício, crie a classe principal e a classe de teste correspondente.
- Execute os testes e verifique os resultados.

Exercício 1 – Calculadora Simples

Objetivo: Criar uma classe Calculadora com os métodos: somar, subtrair, multiplicar e dividir.

1. Crie a classe `Calculadora` com métodos públicos:
 - `int somar(int a, int b)`
 - `int subtrair(int a, int b)`
 - `int multiplicar(int a, int b)`
 - `double dividir(int a, int b)` – deve lançar exceção ao dividir por zero.
2. Crie a classe de testes `CalculadoraTest` e implemente testes para cada método.
3. Utilize o método `assertThrows` para testar divisão por zero.

Exercício 2 – Verificação de Números

Objetivo: Criar métodos que verifiquem se um número é par ou primo.

1. Crie a classe `NumeroUtil` com dois métodos:
 - `boolean ehPar(int n)`
 - `boolean ehPrimo(int n)`
2. Crie a classe `NumeroUtilTest` com testes para ambos os métodos.
3. Teste diferentes entradas, inclusive valores limite (0, 1, 2).

Exercício 3 – Carrinho de Compras

Objetivo: Simular a adição e remoção de produtos em um carrinho e calcular o total.

1. Crie a classe `CarrinhoDeCompras` com:
 - Um `Map<String, Double>` para armazenar produtos.
 - Métodos para `adicionarProduto`, `removerProduto` e `calcularTotal()`.
2. Crie testes em `CarrinhoDeComprasTest` para validar cada operação.
3. Teste valores flutuantes e adições consecutivas.

Exercício 4 – Validação de Senhas

Objetivo: Verificar se uma senha atende aos critérios mínimos de segurança.

1. Crie a classe `ValidadorSenha` com o método:
 - `boolean validar(String senha)`
2. A senha deve ter no mínimo 8 caracteres, conter pelo menos uma letra maiúscula, uma letra minúscula e um dígito.
3. Crie testes em `ValidadorSenhaTest` para diferentes tipos de senha: válidas, curtas, sem números, sem letras maiúsculas etc.

Entrega

Os alunos devem entregar os seguintes arquivos:

- Códigos-fonte das classes implementadas.
- Códigos de teste com cobertura adequada.
- Prints ou relatório gerado pelas ferramentas de cobertura (opcional).