



# Introdução aos Testes de Software com Jest

Bem-vindos à nossa aula teórica e prática sobre testes de software, focada em Jest. Preparem-se para desvendar a importância dos testes e como eles podem elevar a qualidade dos seus projetos de programação.

## Fundamentos dos Testes

# Por que testar?

### Erros, Falhas e Bugs

Entenda as diferenças entre eles e como os testes nos ajudam a identificá-los e corrigi-los antes que afetem o usuário final.

### Ciclo de Desenvolvimento

Os testes são uma etapa crucial. Eles garantem que cada nova funcionalidade não quebre o que já está funcionando.

# Tipos de Testes



## Teste Unitário

Verifica pequenas partes isoladas do código, como funções e métodos, garantindo que cada unidade funcione como esperado.



## Teste de Integração

Valida a interação entre diferentes módulos ou componentes do sistema, assegurando que funcionem bem juntos.



## Teste de Sistema

Testa o sistema completo para garantir que atenda aos requisitos funcionais e não-funcionais, simulando o uso real.



## Teste de Aceitação

Realizado pelos usuários finais ou clientes para verificar se o software atende às suas expectativas e necessidades de negócio.

# Vantagens dos Testes Automatizados



## Agilidade no Desenvolvimento

Permitem identificar e corrigir problemas rapidamente, acelerando o ciclo de desenvolvimento.



## Menor Risco de Regressão

Evitam que novas alterações introduzam bugs em funcionalidades existentes, protegendo o código.



## Confiabilidade

Aumentam a confiança na base de código, garantindo que o software se comporte como esperado.



## Documentação Viva

Os próprios testes servem como exemplos claros de como o código deve ser usado e o que ele faz.

# O Que É o Jest?

Jest é um framework de teste JavaScript popular e muito utilizado por grandes empresas como Facebook, Airbnb e Spotify. Ele foi criado com foco em facilidade de uso e performance.



**Facilidade e Rapidez**



**Cobertura Integrada**



**Comunidade Ativa**

# Jest



# Estrutura Básica de Um Teste com Jest

```
describe('Função soma', () => {  it('soma 2 + 3 corretamente', () => {    expect(2 + 3).toBe(5);  });});
```

## describe()

Agrupa testes relacionados, descrevendo a funcionalidade que está sendo testada.

## it()

Define um caso de teste individual, descrevendo o comportamento esperado da função.

## expect().toBe()

Realiza a asserção, comparando o valor recebido com o valor esperado.

# Criando Testes Para Funções Simples

Vamos aplicar os conceitos de Jest em funções básicas para solidificar o aprendizado.

**Função Soma:** Testa se a função adiciona dois números corretamente.

**Função É Par:** Verifica se um número é par ou ímpar.

**Função Multiplica:** Confere se a multiplicação está exata.

```
// soma.jsfunction soma(a, b) { return a + b; }module.exports = soma;// soma.test.jsconst soma = require('./soma');describe('Função soma', () => { it('deve somar 2 + 3 corretamente', () => { expect(soma(2, 3)).toBe(5); });});
```

# Entendendo a Cobertura de Código

A cobertura de código indica a porcentagem do seu código que é executada pelos testes. É uma métrica importante para avaliar a qualidade dos seus testes.

Alcançar uma meta de 80% de cobertura é um bom objetivo inicial para garantir a qualidade do seu código.

```
npx jest --coverage
```

```

1  npx jest --coverage
2
3  {
4    [
5      {
6        filePath: 'test/coverage/coverage-summary.json',
7        data: {
8          'test/coverage/coverage-summary.json': {
9            'test/coverage/coverage-summary.json': {
10              npx jest --coverage (6)
11              npx jest coverage ( tests fail, 1%
12            }
13          }
14        }
15      }
16    ]
17  }
18 }

```



# Atividade Prática Proposta

1

## Crie Uma Função

Desenvolva uma função JavaScript simples.

2

## Escreva Testes

Use Jest para escrever pelo menos 3 testes automatizados para sua função.

3

## Teste e Reflita

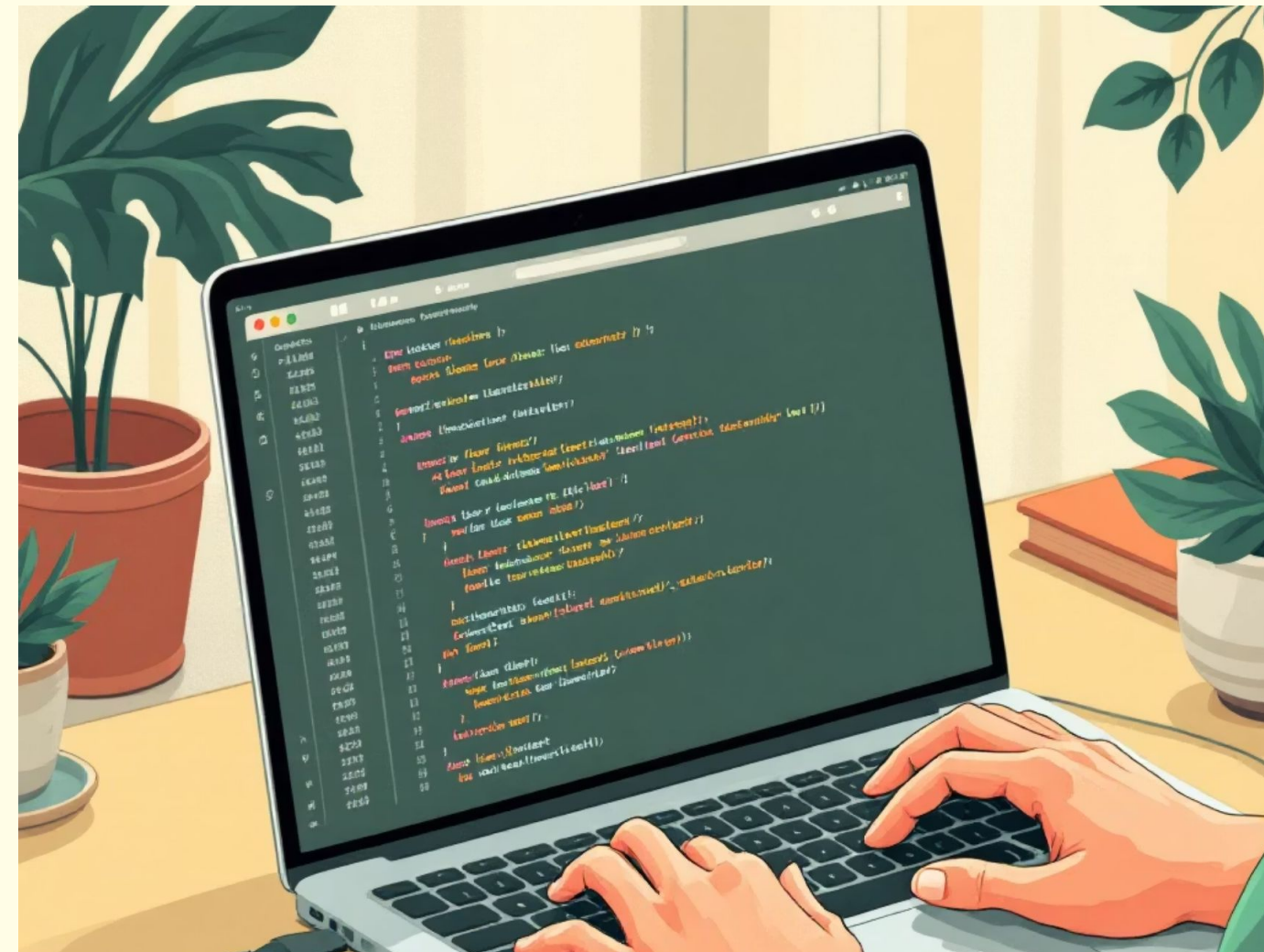
Execute seus testes e analise a cobertura de código para ver seu progresso.

### Sugestões:

**verificaldade(idade):** Retorna se a pessoa é maior de idade.

**converteCelsiusParaFahrenheit(celsius):** Converte temperaturas.

**calculaMedia(notas):** Calcula a média de um array de notas.



**"O que pode acontecer  
em um sistema que não  
tem testes?"**

