

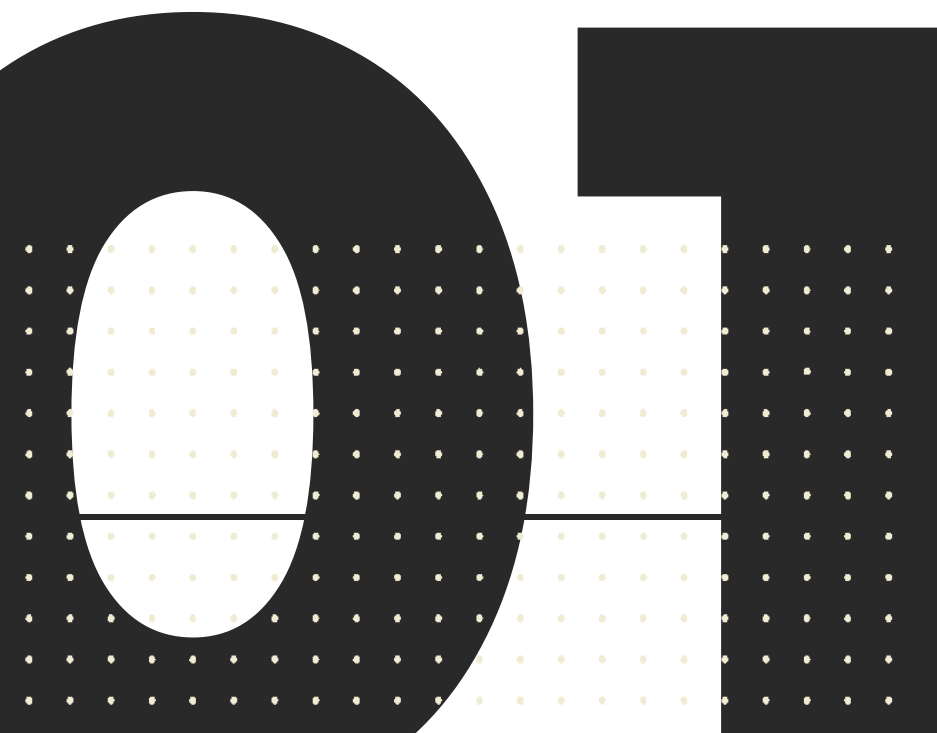
Testes unitários com Jest

Grupo Meio a Meio 2022.1



SUMÁRIO

- + Testes unitários
- + Utilizando Jest
- + Comandos e funções
- + Dojo
- + Perguntas



02

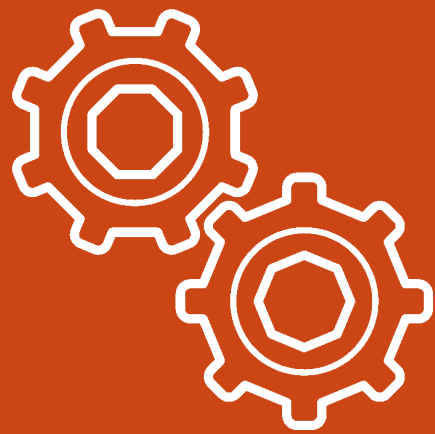
×

O que são Testes Unitários?

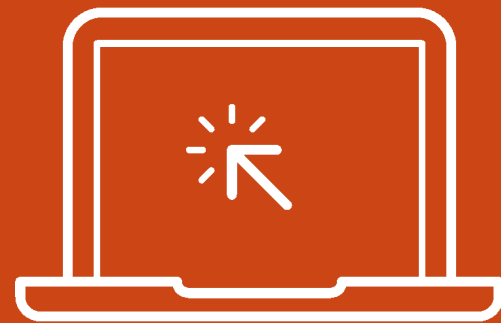
Consiste em verificar o comportamento das menores unidades de uma aplicação.

02

Por que utilizar testes unitários ?



**Feedback
instantâneo da
implementação**



**Diz o quão
testável é seu
código**



**Aumenta a
produtividade
da equipe**

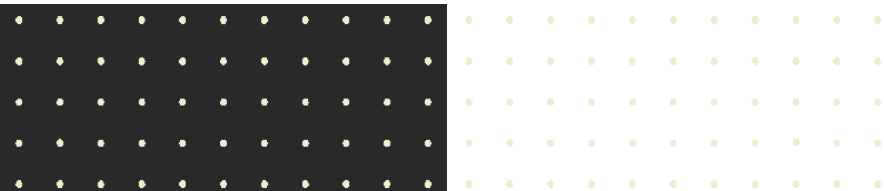


03

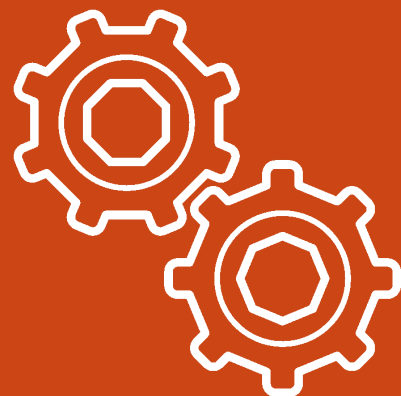


- O que é
- Comandos básicos

Começando com Jest



Comandos



Configurando o ambiente

NPM

|

YARN

```
npm install --save-dev jest
```

```
yarn add --dev jest
```

```
"scripts": {  
  "test": "jest"
```



Executando testes

```
yarn test ou npm run test
```

05



Usando Matchers



Matchers Comuns

```
expect(2 + 2).toBe(4);  
expect(a + b).not.toBe(0);  
expect(data).toEqual({one: 1, two: 2});
```



Verdade

```
expect(n).toBeNull();  
expect(n).toBeDefined();  
expect(n).not.toBeUndefined();  
expect(n).not.toBeTruthy();  
expect(n).toBeFalsy();
```



Números

```
expect(value).toBeGreaterThan(3);  
expect(value).toBeGreaterThanOrEqual(3.5);  
expect(value).toBeLessThan(5);  
expect(value).toBeLessThanOrEqual(4.5);
```



Strings

```
test('não existe I em team', () => {  
  expect('team').not.toMatch(/I/);  
});  
  
test('mas existe "stop" em Christoph', () => {  
  expect('Christoph').toMatch(/stop/);  
});
```



Arrays e iteráveis

```
test('a lista de compras tem leite nela', () => {  
  expect(shoppingList).toContain('leite');  
  expect(new Set(shoppingList)).toContain('leite');  
});
```



Exceções

```
expect(() => compileAndroidCode()).toThrow();  
expect(() => compileAndroidCode()).toThrow(Error);
```

Testando Códigos Assíncronos



Promises

```
test('the data is peanut butter', () => {  
  return fetchData().then(data => {  
    expect(data).toBe('peanut butter');  
  });  
});
```



Callbacks

```
test('the data is peanut butter', done => {  
  function callback(error, data) {  
    if (error) {  
      done(error);  
      return;  
    }  
    fetchData(callback);  
  }  
});
```



Async/Await

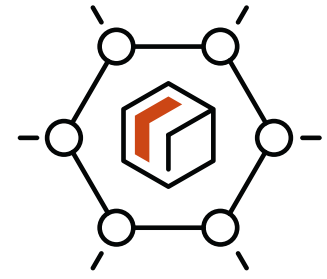
```
test('the fetch fails with an error', async () =>  
  expect.assertions(1);  
  try {  
    await fetchData();  
  } catch (e) {  
    expect(e).toMatch('error');  
  }  
});
```



.resolves/.rejects

```
test('the fetch fails with an error', () => {  
  return expect(fetchData()).rejects.toMatch('error');  
});
```

```
test('the data is peanut butter', () => {  
  return expect(fetchData()).resolves.toBe('peanut butter');  
});
```

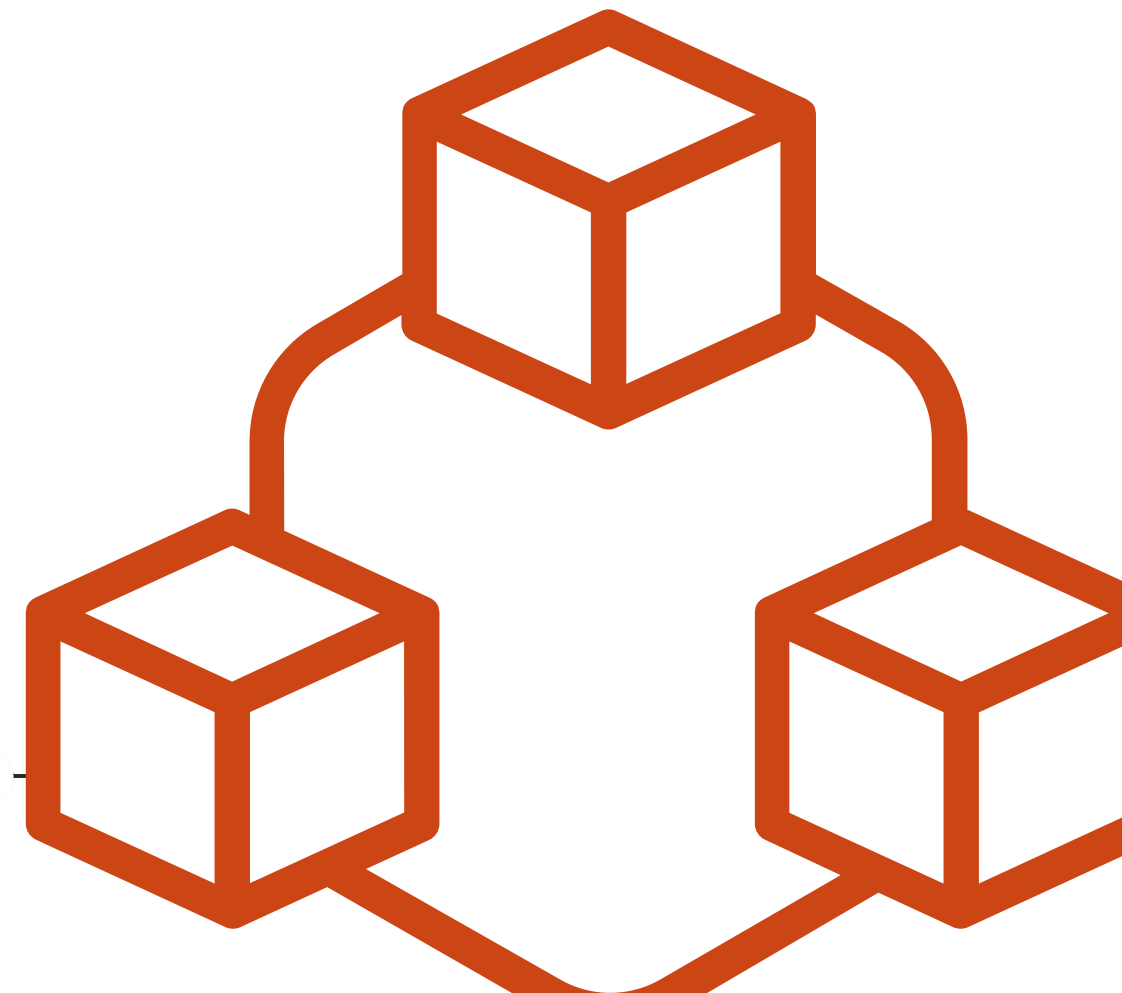



Mocks

08

+ O que são mocks ?

+ Use Cases



DOJO



Perguntas



Perguntas

1. O que são Testes Unitários?

Perguntas

2. Cite um dos benefícios de fazer Testes Unitários.

Perguntas

3. Qual é o símbolo do Jest?

Perguntas

4. Qual é o Matcher utilizado para encontrar expressões regulares em strings?

Perguntas

5. O que são Mocks?

Perguntas

6. O que faz o beforeAll?

Perguntas

**7. O que o Matcher
toEqual faz?**

Perguntas

**8. O que o Matcher
toContain faz?**

Perguntas

9. Quais são as assinaturas de uma função assíncrona?

Perguntas

10. Cite dois matchers de comparação de números.



Obrigado!!