

Testes Unitários

Leticia; Luiz; Matheus; Yuri.



Conceito →

São testes automatizados que verificam o funcionamento correto das partes de um código.

A **importância** de um **teste unitário** bem elaborado para um sistema web.

- Garantir qualidade
- Correções preventivas
- Garantir estabilidade
- Confiabilidade
- Segurança

A **importância** de um **teste unitário** bem elaborado em relação ao **Mercado**.

- Confiabilidade do mercado
- Fidelização de Clientes
- Competitividade

Testes Unitários & Jest & TypeScript



Jest com TypeScript

Instalação ts-jest

```
npm install typescript --save-dev
```

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

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

```
npm install @types/jest --save-dev
```

Criando o arquivo de configuração

```
npx ts-jest config:init
```

Final da instalação

JS jest.config.js ✕

jestTypescript > JS jest.config.js > ...

```
1  /** @type {import('ts-jest').JestConfigWithTsJest} */
2  module.exports = {
3    preset: 'ts-jest',
4    testEnvironment: 'node',
5  };
```

{ } package.json ✕

jestTypescript > { } package.json > ...

```
1  {
2    ▶ Debug
3    "scripts": {
4      "test": "jest"
5    },
6    "devDependencies": {
7      "@types/jest": "^29.5.0",
8      "jest": "^29.5.0",
9      "ts-jest": "^29.0.5"
10  }
```

Teste com Javascript

```
// Função 1 - doubleANumber  
const doubleANumber = (number) => number * 2;
```

```
// Teste  
const ds = require("../desafio");  
  
describe("doubleANumber", () => {  
  it("should return double of the number", () => {  
    const param = 2;  
    expect(typeof param).toBe("number");  
    const result = ds.doubleANumber(param);  
    expect(result).toBe(4);  
  });  
});
```


Mudanças

- Formato de exportação e importação das funções
- Adição das tipagens
- Estabelece o tipo de entrada e saída
- Pode evitar que sejam passados argumentos do tipo errado para a função
- Melhor para documentar o código



Teste com Typescript

```
// Função 1 - doubleANumber  
export const doubleANumber = (number: number): number => number * 2;
```

```
// Teste  
import { doubleANumber } from "../funcoes";  
  
describe("doubleANumber", () => {  
  it("should return double of the number", () => {  
    const param: number = 2;  
    const result: number = doubleANumber(param);  
    expect(result).toBe(4);  
  });  
});
```

Teste com Typescript

```
// Função 1 - doubleANumber
export const doubleANumber = (number: number): number => number * 2;
```

```
1 // Teste
2 import { doubleANumber } from "../funcoes";
3
4 describe("doubleANumber", () => {
5     it("should return double of the number", () => {
6         const result: number = doubleANumber("string");
7         expect(result).toBe(4);
8     });
9 });
10
```

PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

Filter (e.g. text, **/*.ts, !**/node_modules/**)

▼ TS funcoes.test.ts jestTypescript 1

⊗ Argument of type 'string' is not assignable to parameter of type 'number'. ts(2345) [Ln 6, Col 42]

↖ Rethink