

TUTORIAL MOCHA E CHAI

1) Instalando Mocha, Chai e Dependências

No terminal, execute:

```
npm install --save-dev mocha chai ts-node @types/mocha @types/chai
```

Explicação:

- mocha: Framework de testes.
- chai: Biblioteca de asserções.
- ts-node: Permite rodar TypeScript diretamente no Node.js.
- @types/mocha e @types/chai: Tipagens para melhor compatibilidade com TypeScript.

2) Configurando o Mocha

Edite o package.json com a seguinte configuração:

```
"mocha": {  
  "require": "ts-node/register",  
  "recursive": true,  
  "timeout": 5000  
}
```

Isso configura:

- Uso do ts-node para interpretar TypeScript.
- Execução recursiva de testes em subdiretórios.
- Timeout padrão de 5000ms.
- Uso do **reporter** padrão spec.

3) Estruturando os Arquivos de Teste

Por convenção, os testes ficam em test/ ou ao lado dos arquivos testados, com o sufixo .spec.ts ou .test.ts. Sua estrutura do projeto deve estar parecido com algo assim:

```
/my-project
├── /src
│   ├── calculator.ts
│   └── index.ts
├── /test
│   └── calculator.spec.ts
├── package.json
└── tsconfig.json
```

4) Criando um código para testar

Arquivo: src/calculator.ts

```
export class Calculator {
  add(a: number, b: number): number {
    return a + b;
  }

  subtract(a: number, b: number): number {
    return a - b;
  }
}
```

5) Criando os Testes com Mocha e Chai

Arquivo: test/calculator.spec.ts

```
import { expect } from "chai";
import { Calculator } from "../src/calculator";

describe("Calculator", () => {
  let calculator: Calculator;

  beforeEach(() => {
    calculator = new Calculator();
  });

  it("should add two numbers correctly", () => {
```

```
const result = calculator.add(2, 3);
expect(result).to.equal(5);
});

it("should subtract two numbers correctly", () => {
  const result = calculator.subtract(5, 3);
  expect(result).to.equal(2);
});

it("should return a number when adding", () => {
  const result = calculator.add(10, 15);
  expect(result).to.be.a("number");
});
});
```

6) Rodando os Testes

Adicione um script no package.json:

```
"scripts": {
  "test": "mocha"
}
```

Execute os testes:

```
npm test
```

Saída esperada:

```
Calculator
  ✓ should add two numbers correctly
  ✓ should subtract two numbers correctly
  ✓ should return a number when adding

3 passing (10ms)
```

7) Cobertura de código com NYC

```
npm install --save-dev nyc
```

Adicione ao package.json:

```
"scripts": {  
  "test": "mocha",  
  "coverage": "nyc npm test"  
}
```

Execute:

```
npm run coverage
```

Saída esperada:

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line
All files	100	100	100	100	