

Tutorial para aplicar TDD em um projeto React.js

Passo 1: Configuração inicial

1. Crie um novo projeto React.js usando o comando `npx create-react-app nome-do-projeto` ou navegue até o diretório do projeto existente.
2. Abra um terminal na raiz do projeto e execute `npm install --save-dev jest enzyme enzyme-adapter-react-16` (ou use o Yarn) para instalar as bibliotecas Jest e Enzyme, que serão usadas para escrever e executar os testes.
3. Crie uma pasta chamada `__tests__` na raiz do seu projeto para armazenar os arquivos de teste.

Passo 2: Escrevendo o primeiro teste

4. Dentro da pasta `__tests__`, crie um arquivo chamado `App.test.js`.
5. Dentro do arquivo `App.test.js`, importe as bibliotecas necessárias e escreva o primeiro teste para verificar se o componente `App` renderiza corretamente:

```
1 import React from 'react';
2 import {render, screen} from '@testing-library/react'
3 import App from './App';
4
5 test('renderizando a mensagem Ola', () => {
6   render(<App />);
7   const helloElement = screen.getByText(/Ola/i);
8   expect(helloElement).toBeInTheDocument();
9 });
```

Passo 3: Executando o primeiro teste

6. Abra um terminal na raiz do projeto e execute `npm test` (ou use o Yarn) para executar os testes.
7. O teste que você acabou de escrever deve com erros.

```
PASS src/utils/sum.test.jsx
FAIL src/App.test.jsx
  ● renderizando a mensagem Ola
```

Passo 4: Implementando o código de produção

8. Abra o arquivo `App.js` na raiz do seu projeto.
9. Implemente um componente básico `App` que renderiza um texto simples:

```
1 import React from "react";
2 import { BrowserRouter as Router, Routes, Route }
  from "react-router-dom";
3 import LoginForm from "../pages/loginform";
4 import Home from "../pages/home";
5
6 const App = () => {
7   return (
8     <div>
9       <h1>Ola</h1>
10    </div>
11  );
12 };
13
14 export default App;
15
```

Passo 5: Executando o teste novamente

10. Abra um terminal na raiz do projeto e execute `npm test` (ou use o Yarn) para executar os testes.
11. O teste deve executar sem erros.

```
PASS src/utils/sum.test.jsx
PASS src/App.test.jsx
A worker process has failed to exit gracefully and has been
find leaks. Active timers can also cause this, ensure that .

Test Suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        2.962 s
Ran all test suites.
```