

Teste de Software

Uma implementação a reactJS

Data: 29 de Novembro de 2021

Introdução

O que é teste de Software ?



Vantagens


 Aumento de produtividade

 Diminuição dos erros:

 Reuso

Desvantagens

 Trabalho para criar automatizações

 Trabalho para criar Custos com as estruturas para suportar as ferramentas

TIPOS DE TESTE

Teste de Big Bang

Ao usar esta técnica, os módulos são testados separadamente e depois integrados de uma só vez.

Teste Regressão

que consiste na aplicação de versões mais recente do software,

Teste de Unidade

teste básico em um software, fornecendo valores válidos ou inválidos e verificando se o retorno foi de acordo com o esperado.



Teste de Segurança

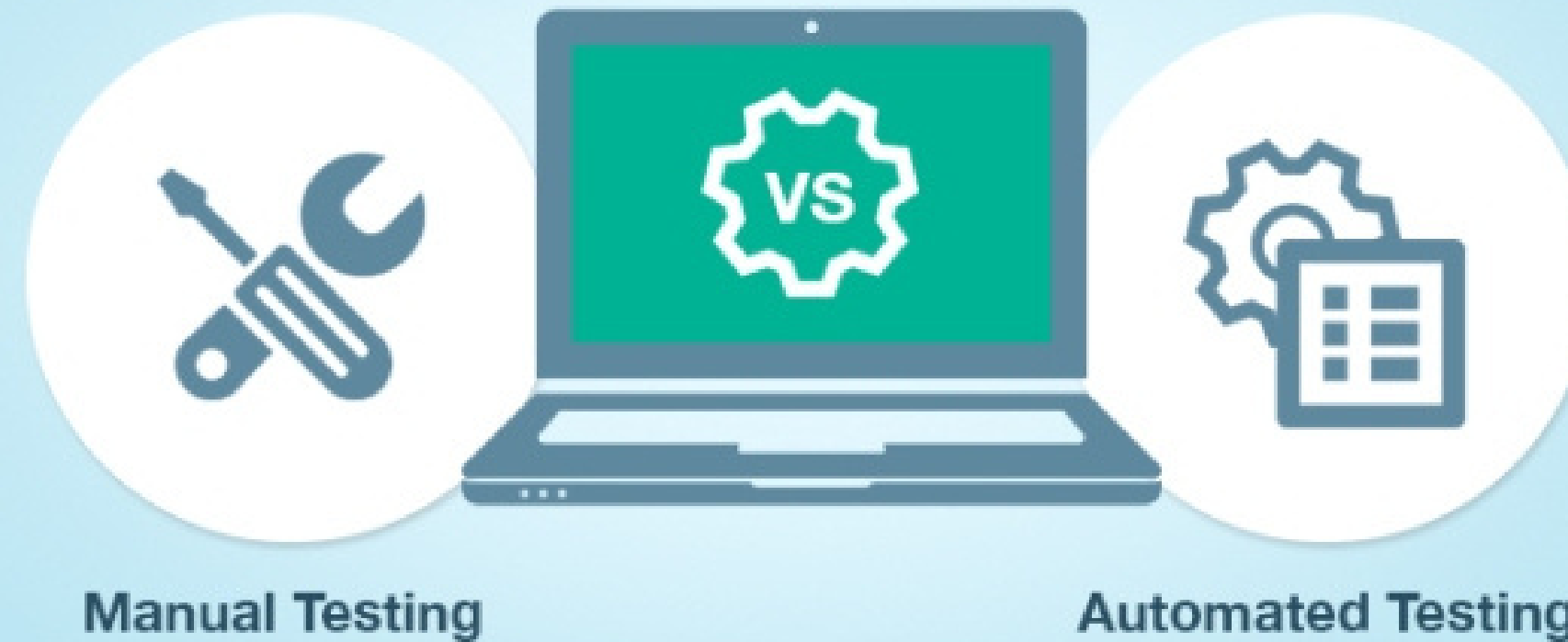
e Teste de Invasão visam identificar as falhas de segurança de um software ou ambiente em que está sendo executado

Teste de Integração

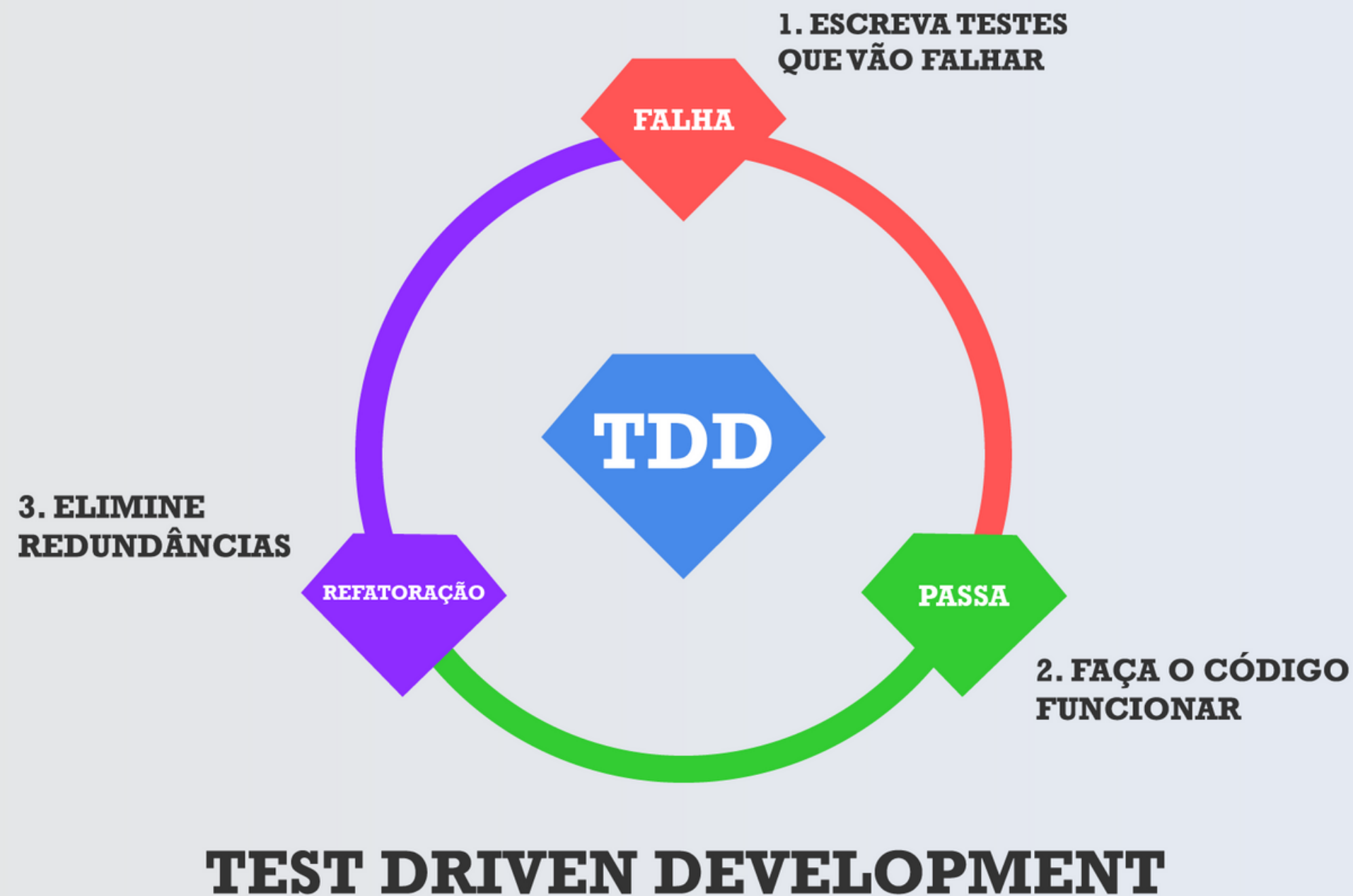
é quando os módulos (unidades de código) são testados em grupo para garantir que não haja nenhuma quebra naquilo que foi feito unitariamente e naquilo que está sendo integrado junto

Modelos de Desenvolvimento orientado a testes

O que são ?



TDD - Test Driven Development



Vantagens



Feedbacks rápidos



Teste objetivo



Códigos simples

Desvantagens



Dificuldade em começar, pois não tem o código base



Tempo de prática do TDD é maior

BDD - Behavior Driven Development

1. Descoberta

Visão de negócio, Descoberta de funcionalidades, Criação de fluxos de negócio

2. Definição

Regras de negócio, Exemplos de funcionamento, Entendimento compartilhado

3. Formalização

Critérios de aceitação (Gherkin), Protótipos funcionais, Wireframes BDD

4. Entrega

Software de valor para o negócio, Monitoramento e feedback



Vantagens



Compartilhamento de conhecimento



Documentação dinâmica:



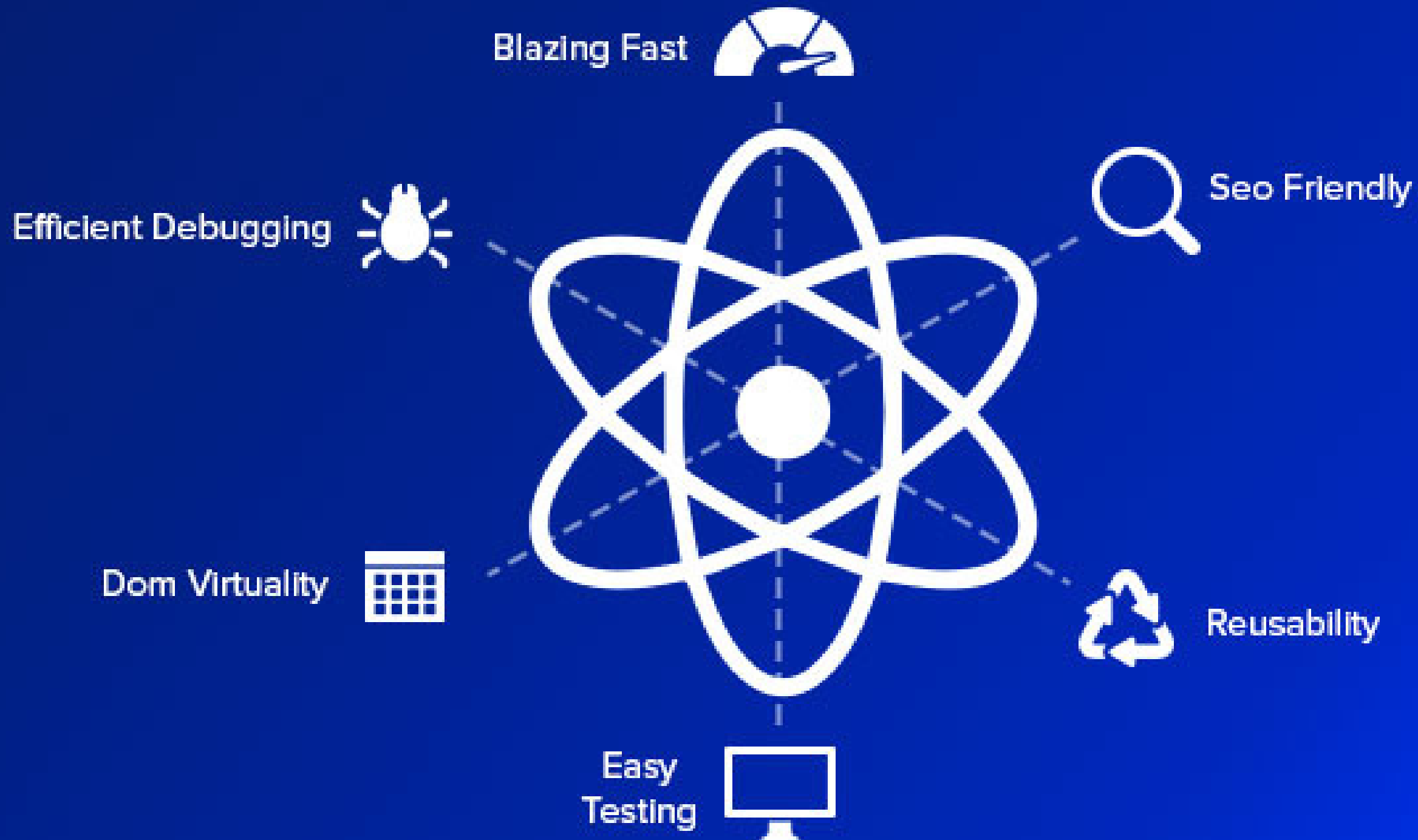
Comunicação entre equipes

Desvantagens



Torna difícil empregar com programadores menos experientes, especialmente se eles não têm experiência em TDD.

Testing ReactJS



TESTE DA APLICAÇÃO

ReactJS

Teste De Interface
serve para garantir que todos os itens de uma tela funcionem adequadamente, analisando a interação de um aplicativo, site ou programa com o usuário final.

Teste Unitário

é também conhecido como teste de unidade, ele testa os pedaços pequenos e isolados de código.

Teste Render

O React test renderer é um pacote que fornece um renderizador de componentes que não depende da DOM.

Teste Unitário

Componete

JS App.js M X

pwebtest > src > JS App.js >  sum

```
1 | function sum(a, b) {  
2 |   return a + b;  
3 | }  
4 | module.exports = sum;
```

Test

JS App.test.js M X

pwebtest > src > JS App.test.js >  test('adds 1 + 2 to equal 3') cal

```
1 | const sum = require('./App');  
2 |  
3 | test('adds 1 + 2 to equal 3', () => {  
4 |   expect(sum(1, 2)).toBe(3);  
5 | });
```

Teste Render

```
const valueTesting = 'testing'

const { getByTestId, getByText } = render(<Lista/>)

const checkInput = await waitForElement(
  () => getByTestId('form-field')
)
fireEvent.change(
  checkInput,
  { target: { value: valueTesting } }
)
expect(checkInput.value).toEqual(valueTesting)
```

Teste Interface

```
const { getByTestId, getByText } = render(<Lista/>)

const checkInput = await waitForElement(
  () => getByTestId('form-field')
)
fireEvent.change(
  checkInput,
  { target: { value: valueTesting } }
)
expect(checkInput.value).toEqual(valueTesting)

//Bela
const checkButton = await waitForElement(
  () => getByTestId('form-btn')
)
fireEvent.click(checkButton)

//vivi
const checkTable = await waitForElement(
  () => getByText(valueTesting)
)
expect(checkTable).toBeDefined()
})
})
```


A Equipe



DÉBORA FERRO



ISABELE YASMIN



ITALO SANTOS



VITORIA DANTAS



VIVIANE SILVA

Referencias:

Introdução:

- <https://blog.cedrotech.com/entenda-importancia-de-testes-de-uiux-para-o-desenvolvimento-de-um-projeto>
- <https://blog.betrybe.com/tecnologia/testes-unitarios/>
- <https://pt-br.reactjs.org/docs/test-renderer.html>

Referencias:

Desenvolvimento:

- <https://www.eduardopires.net.br/2012/06/ddd-tdd-bdd/>
- <https://jestjs.io/pt-BR/>
- <https://testing-library.com/docs/react-testing-library/intro/>

Referencias:

Video:

- <https://www.youtube.com/watch?v=sdkgUu5hr6g>

Playlist youtube:

- <https://www.youtube.com/playlist?list=PLQ8doQ5hhsohpHIFm8-fYWY78p9ASDU0g>