

POR QUE TESTAR?

"É IMPENSÁVEL DESENVOLVER PROFISSIONALMENTE SEM INCLUIR TESTES"

- **Confiança ao refatorar**
- **Confiança ao incluir novas funcionalidades**
- **Confiança ao atualizar dependências**
- **Facilita a compreensão da implementação**
- **Funciona como documentação**

O QUE TESTAR?

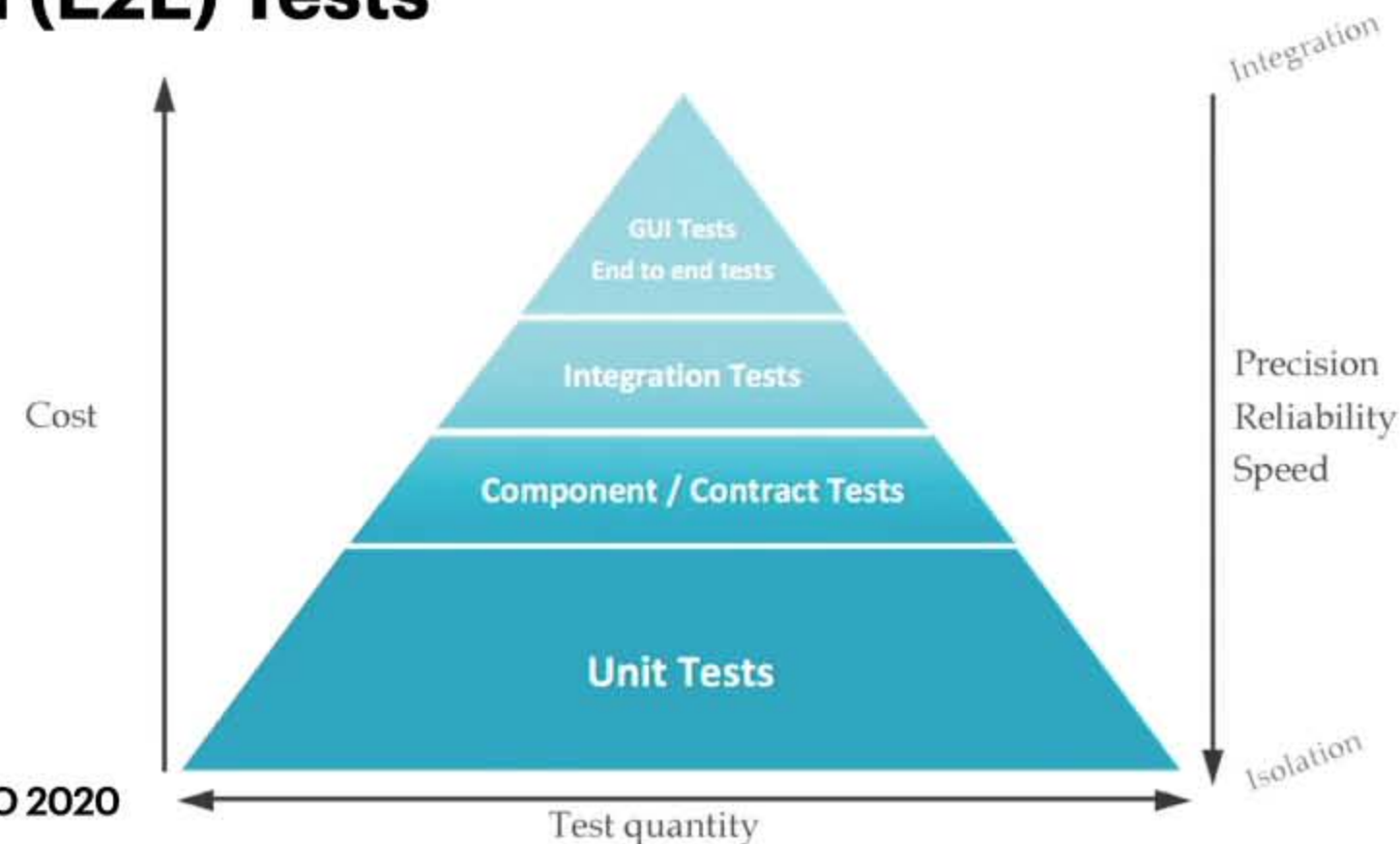
- Tudo o que fizer sentido testar!
- “Devo deixar um comentário?”
Escreva um teste no lugar! 🚀



PIRÂMIDE DE TESTES

OS TIPOS MAIS COMUNS DE TESTES

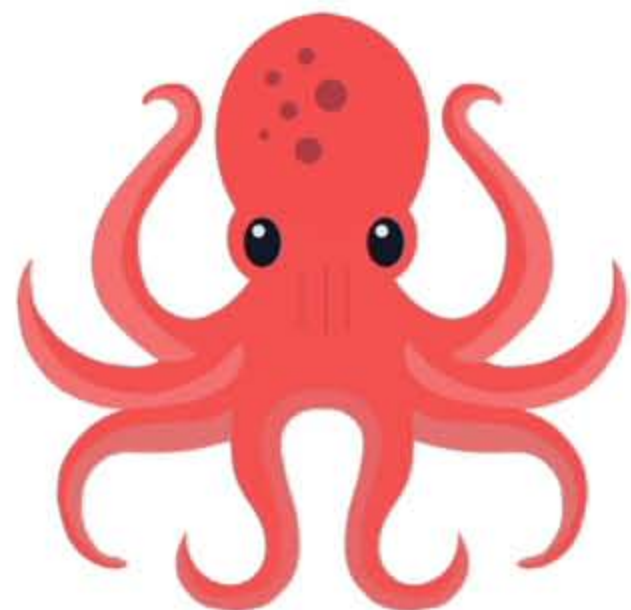
- Unit Tests
- Integration Tests
- End to End (E2E) Tests



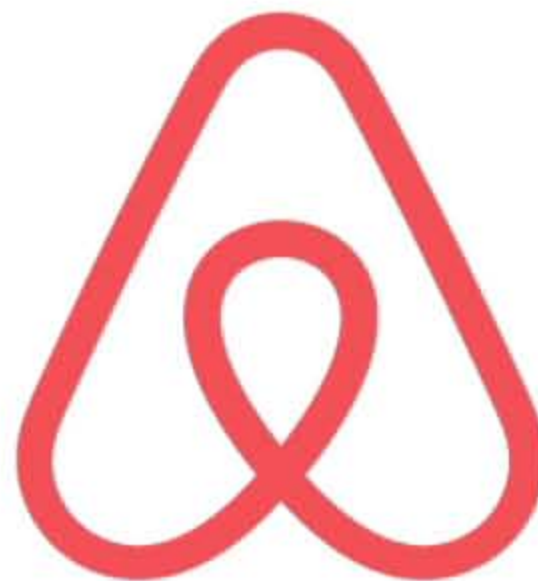
FERRAMENTAS



Jest



Testing Library



Enzyme



Vue Test Utils



Cypress

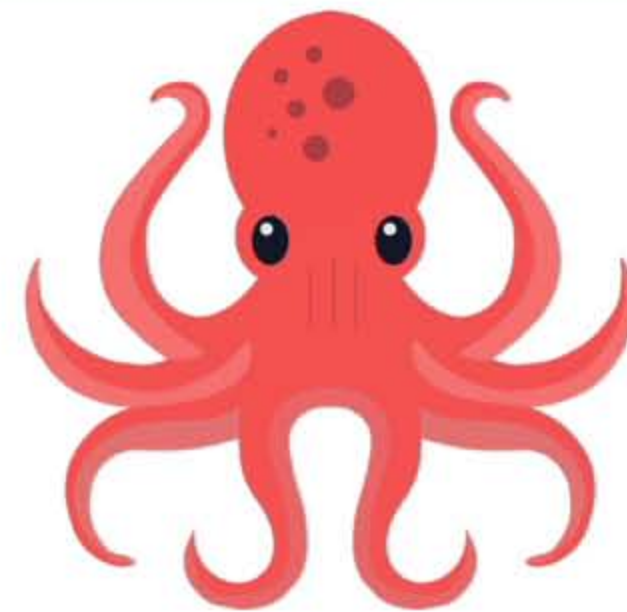
FERRAMENTAS

- É um test runner
 - Localiza os arquivos e executa os testes
 - Permite fazer mock e observar métodos em bibliotecas
 - Permite fazer assertions, tais como `expect().toBe()`



Jest

FERRAMENTAS



Testing Library

- É uma biblioteca
 - Provê utilitários para montar componentes (React, Vue, Svelte ...)
 - Oferece ferramentas de interagir com componentes como se fosse o usuário
 - Trabalha em conjunto com o Jest
 - Faz o mesmo papel do Enzyme e Vue-test-utils
 - Pode ser usada para substituí-las

FERRAMENTAS

- É uma biblioteca
 - Provê utilitários para montar componentes React
 - Oferece maneira de interagir com componentes como se fosse o usuário
 - Permite manejar state e testar detalhes de implementação
 - Trabalha em conjunto com o Jest
 - Faz o mesmo papel da Testing Library
 - Pode ser usada para substituí-la



Enzyme

FERRAMENTAS

- É uma biblioteca
 - Provê utilitários para montar componentes Vue.js
 - Oferece maneira de interagir com componentes como se fosse o usuário
 - Permite manejar state e testar detalhes de implementação
 - Trabalha em conjunto com o Jest
 - Faz o mesmo papel da Testing Library
 - Pode ser usada para substituí-la



Vue Test Utils

FERRAMENTAS

- É um framework de testes End to End
 - Executa a aplicação no browser, como se fosse o usuário
 - Permite executar no Chrome, Firefox, Edge e Electron
 - Uma ferramenta completa, não precisa ser usada com Jest



Cypress


CI/CD


O PAPEL DOS TESTES NO DEPLOY MODERNO

Os testes são integrados ao processo de Continuous Integration / Continuous Delivery e são fundamentais para que estes existam.



- **Github Actions**
- **CircleCI**








Search or jump to...


[Pull requests](#) [Issues](#) [Trending](#) [Explore](#)


 [vedovelli / classroom.vedovelli.com.br](#) Private 


 Code


 Issues


 Pull requests 2


 **Actions**

 Projects 1


 Wiki

 Security

 Insights

 **Locks curso js testes**

locks-curso-js-tes...

 79claca

▼ ci

on: pull_request

✓ build

ci / build

succeeded 8 days ago in 6m 21s

▶ ✓ Set up job

▶ ✓ Checkout repository

▶ ✓ Set up Node

▶ ✓ Get yarn cache directory path

▶ ✓ Run actions/cache@v1

▶ ✓ Install dependencies

▶ ✓ Build project

▶ ✓ Run http server with the app in the background

▶ ✓ Run unit tests

▶ ✓ Run E2E tests

▶ ✓ Post Run actions/cache@v1

▶ ✓ Post Checkout repository

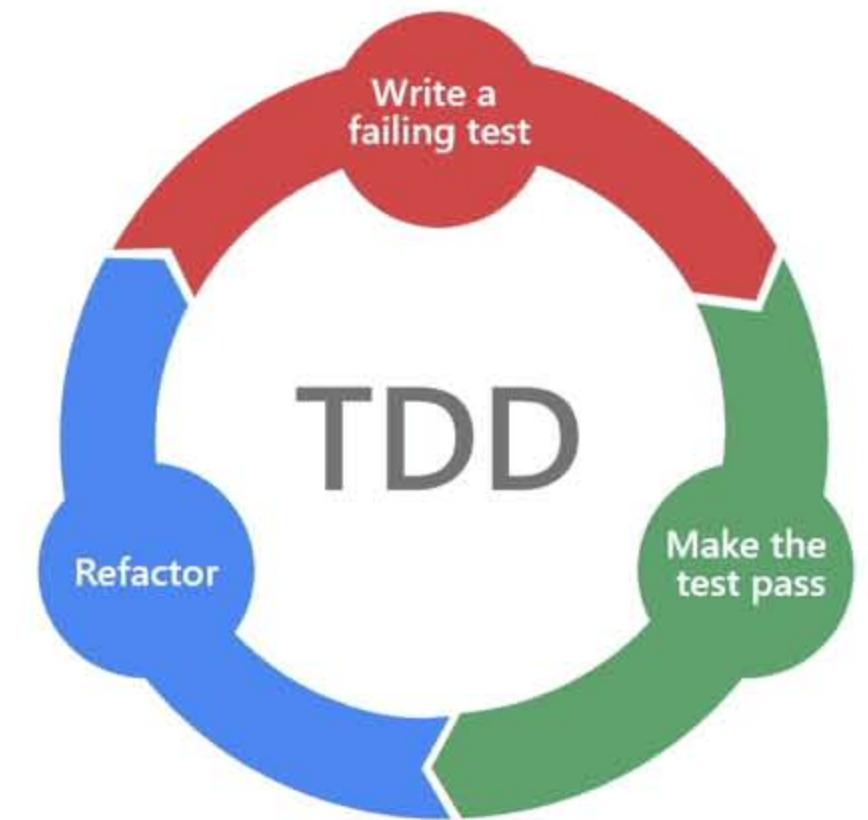
▶ ✓ Complete job

Github Actions

TDD

VANTAGENS

- **Melhora a implementação**
- **Facilita implementações mais simples**
- **Facilita a escrita dos testes**
- **Detalhes implementação frescos na mente**
- **Menor tempo dedicado a bug fixing**



TDD

DESAFIOS

- **Curva de aprendizado**
- **Maior tempo de desenvolvimento**
- **Difícil de vender ao time de produto**

