

Iniciando com testes em React

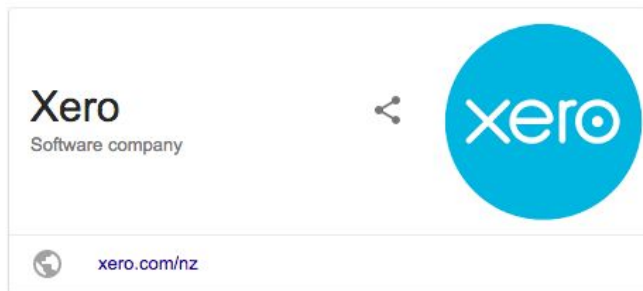
React Nanodegree - Udacity Brasil





Eduardo Rabelo

@oieduardorabelo - twitter, github, etc



<- Atualmente aqui
Em Wellington,
Na Nova Zelândia



Agenda e Expectativas

- Valores e importância de Testes
- Introdução as metodologias BDD e TDD
- Prática, prática e prática! (Matheus Marsiglio + Exemplo)



Valores e importância de Testes

O porquê devemos ou não escrevê-los!





Willian Justen entrou na sala...



@willian_justen

"Wes Bos" Brasileiro

willianjusten.com.br/cursos

Teste seu software, ou seus usuários irão testar por você!

É qualidade para você, para o seu chefe e também, seu cliente!



willianjusten.com.br/entendendo-testes-de-software

Sibelius Seraphini entrou na sala...



@sseraphini

React Conf Brasil / Entria

entria.com.br



Testes te ajudam a refatorar o código com segurança.

Tipos de ajudam a refatorar o código com **mais** segurança.

Etudo isso, é para encontrar bugs antes deles irem para produção

Jean Carlo Emer entrou na sala...



@jcemer

Mago JS na Work & Co

jcemer.com

O que melhor funcionou nos projetos em que já trabalhei foi enxergar as porções de código que não são declarativas e garantir que estas sejam testadas.



medium.com/@jcemer <- VÁRIOS artigos maneiros!

Jean Carlo Emer entrou na sala...



@jcemer

Mago JS na Work & Co

jcemer.com

Código Front-end sempre terá muitas porções declarativas e não costumo escrever testes unitários para estas porções. Nunca tive pretensão de seguir TDD a risca.



medium.com/@jcemer <- VÁRIOS artigos maneiros!

Fernando Daciuk entrou na sala...



@fdaciuk

Mentor no React Ninja
blog.da2k.com.br/cursos



TDD é uma forma de se pensar.

É como um paradigma de programação:

Se você não estiver alinhado com esse pensamento, com essa forma de escrever software, você vai desistir de escrever testes no meio do caminho.

Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter



Escriver testes pode parecer uma fase extra para o projeto. Na verdade ela elimina várias fases de manutenção em um futuro não muito distante.

Trabalhando no TC39/Test262 - O Leo é "O Cara" dos testes
A suite de testes da especificação do ECMAScript

Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

O custo do teste se paga com sobras na hora do jogo da quarta-feira.

Combine isso com um sistema automatizado de integração contínua e fique sem medo de fazer deploy no final da tarde de uma sexta feira.



Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

Se alguma coisa pode ser documentada para ser transformada em código, ela com certeza também pode ser testada.



Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

A dica é simples: Se você executa uma operação, você espera um resultado.

Os testes não fazem nada mais do que isso, testes unitários se fragmentam nas menores partes para testar cada função.

Chamou a função com esse parâmetro, verificou o que ela retornou? Pronto, próximo teste!



Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

O JavaScript, a linguagem, é documentado via ECMAScript e o **Test262** (no qual o Leo trabalha) contém todos os testes oficiais que são consumidos diretamente por todos os navegadores (Firefox, Chrome, Safari etc).



Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

Se lembra quando reclamávamos de inconsistência entre navegadores diferentes?

Essa história está cada vez mais no passado porque há uma série de testes prevenindo surpresas.



Leo Balter entrou na sala...



@leobalter

Bocoup/Vice Chair of TC39

github.com/leobalter

Escriver testes é programar
sem surpresas!



Conclusão?... 🤨🤨🤨

1. **Identifique** a necessidade ou problema
2. **Imagine** ou esboce uma solução (pode ser em papel mesmo)
3. **Escreva** um código para implementar sua ideia
4. **Escreva** um teste para validar seu código



-> Ao inverter 3 e 4, é o que chamamos de TDD!

Introdução as metodologias BDD e TDD

Suas principais diferenças e "formas de se pensar"



BDD

Behavior Driven Development



Behavior Driven Development (BDD)

- Tem foco em um cenário/processo específico
- Envolve todas as partes do cenário/processo
- Usa uma linguagem ubíqua para descrever um comportamento



Behavior Driven Development (BDD)

História: Usuário utilizando o estoque

Cenário 1: Quantidade de items no estoque

-> JIRA, Trello, etc

Dado que **há 5 items no estoque**

E um cliente **comprou 2 items do estoque**

Então quando **contar os items restantes no estoque**

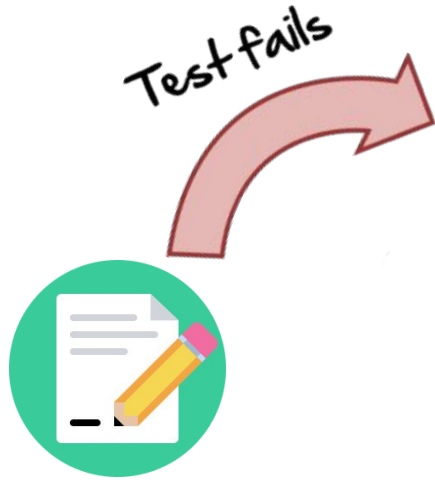
Terei **3 items no estoque**



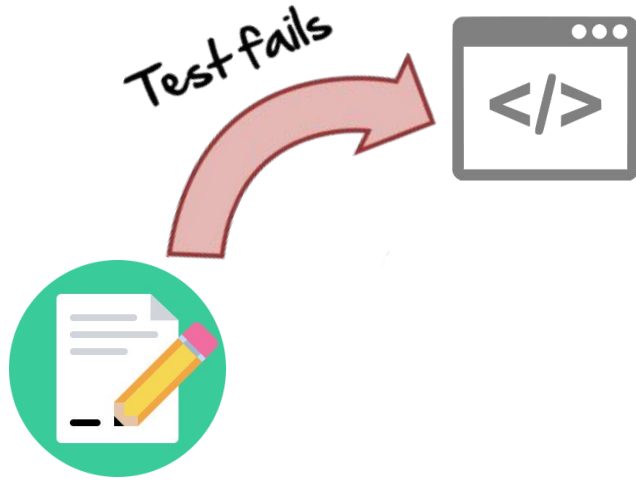
Behavior Driven Development (BDD)



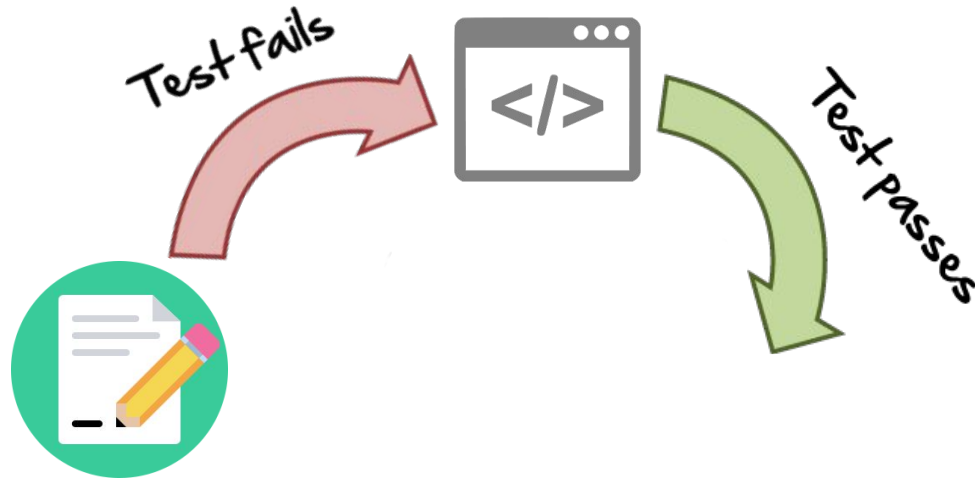
Behavior Driven Development (BDD)



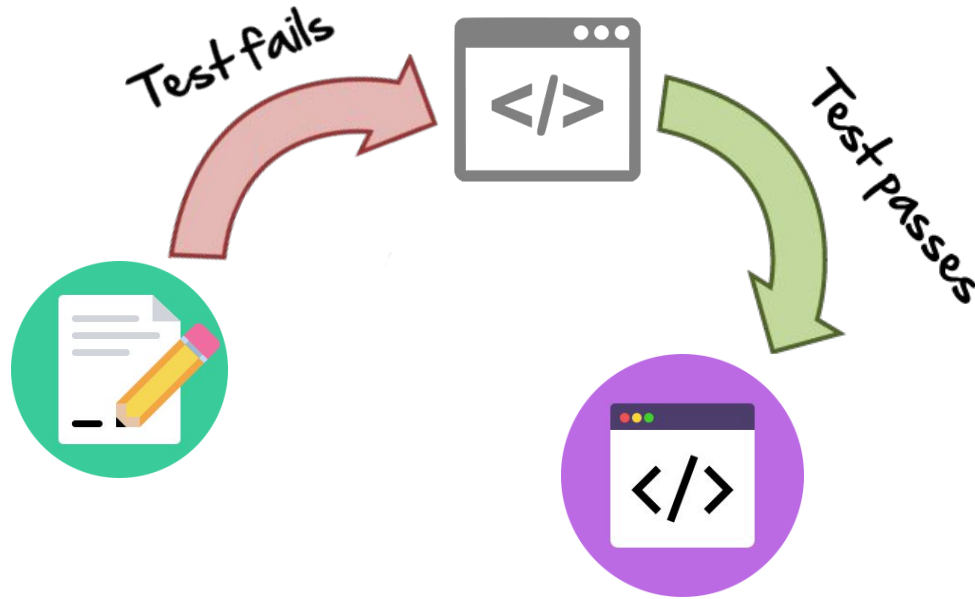
Behavior Driven Development (BDD)



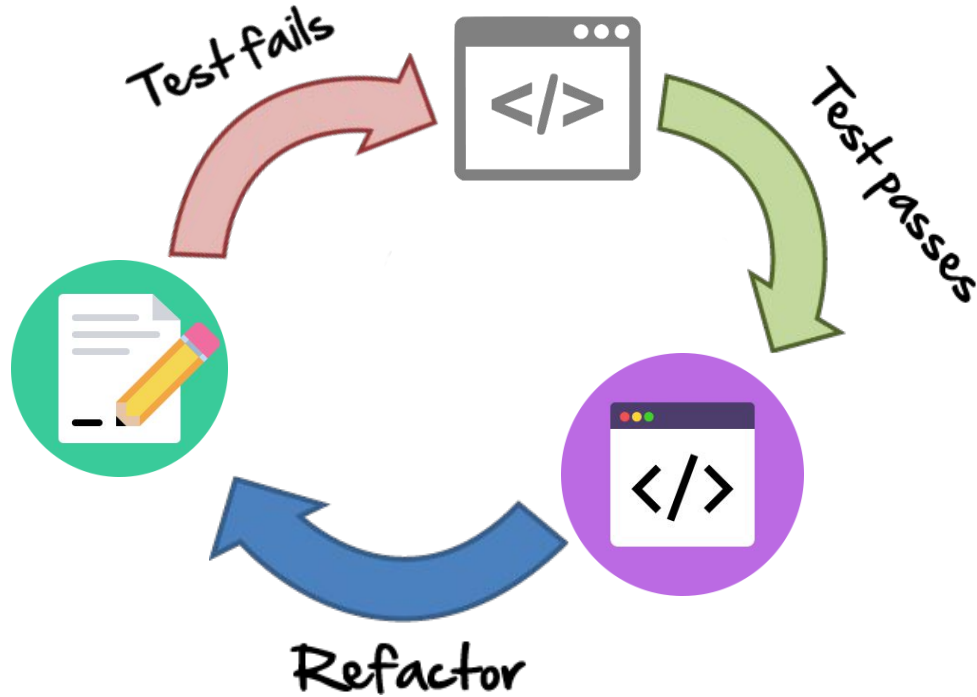
Behavior Driven Development (BDD)



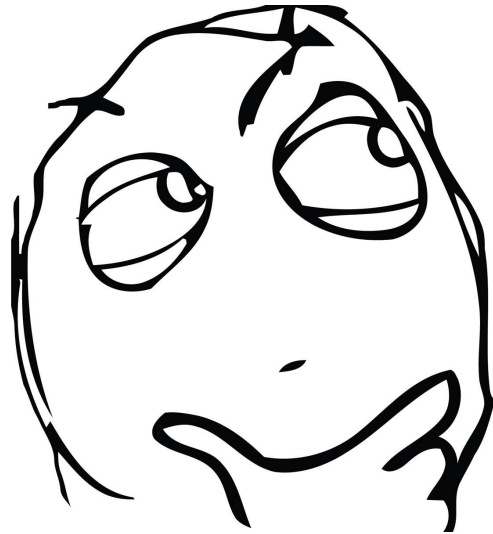
Behavior Driven Development (BDD)



Behavior Driven Development (BDD)



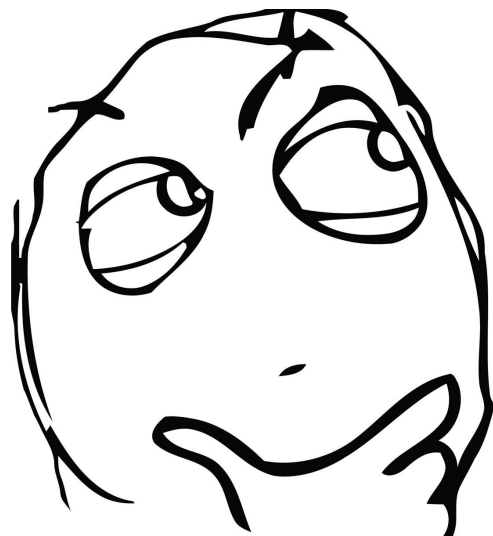
Behavior Driven Development (BDD)



Behavior Driven Development (BDD)

Isso não é TDD?

Por que ele está falando que é BDD?



Behavior Driven Development (BDD)

- BDD é derivado do TDD
- O foco é no cenário, não na unidade
- Você testa a funcionalidade como um todo,
não pedaço por pedaço



TDD

Test Driven Development

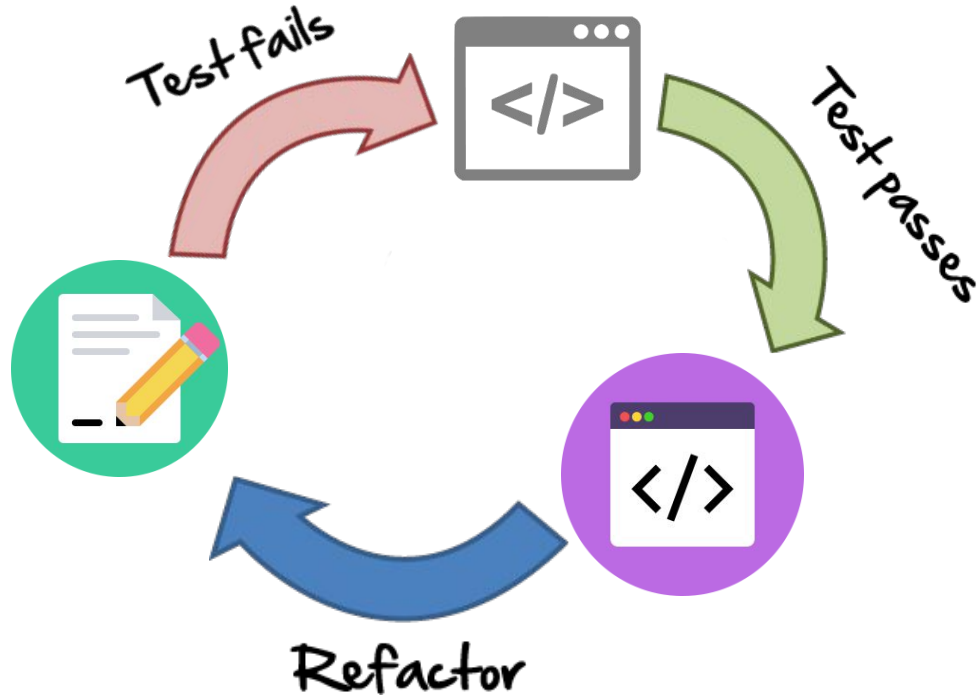


Test Driven Development (TDD)

- Tem foco em testar a unidade/função
- Ao pensar no teste primeiro, você melhora seu entendimento da implementação
- Escreva código **apenas** para fazer o teste passar



Test Driven Development (TDD)



Test Driven Development (TDD)

- Qualidade no código final
- Auto-documentação
- Refatoração/Arquitetura



Prática, prática e prática!

Usaremos o projeto do live do Matheus Marsiglio



Jest



Delightful JavaScript Testing

[TRY OUT JEST](#)[GET STARTED](#)[WATCH TALKS](#)[LEARN MORE](#) Star

11,808



- Já vem configurado no **create-react-app**
- **Convenções** e estrutura (`__tests__`, `.test`, etc)
- **Tudo** que você precisa! (asserções, stubs e task runner!)





Eduardo Rabelo

Dec 2, 2016 · 5 min read

Painless JavaScript Testing

Jest is a JavaScript testing framework, used by Facebook to test all JavaScript code including React applications.
News: Jest 16.0, Turbocharged CLI & Community Update

[GET STARTED](#)[REACT TESTING](#)[REACT NATIVE TESTING](#)[API REFERENCE](#)[★ Star](#)[6,518](#)

Jest: Escrever testes nunca foi tão divertido

Nessa longa estrada, já trabalhei com muitos frameworks para testes (escolha a...

[Read more...](#)



102

5 responses



medium.com/@oieduardorabelo



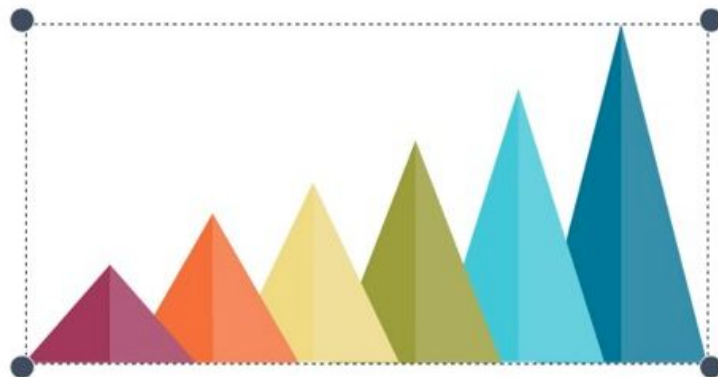
Eduardo Rabelo

That's one small step for a man - <https://eduardorabelo.me>

May 5 · 17 min read

O ecossistema de Testes em JavaScript em 2017

Uma visão geral sobre ferramentas, estratégias e o que escolher!



Diferentes escolhas para cada tamanho de projeto e problema!

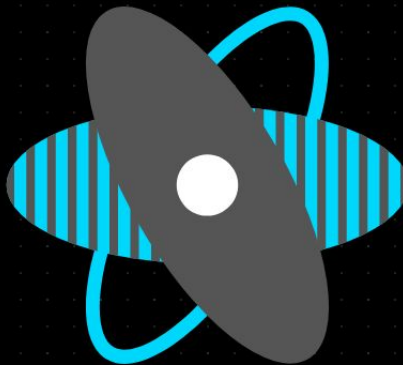


medium.com/@oieduardorabelo

REACTCONF

2017

INGRESSOS



DATA

07.10.2017

SÁBADO, DAS 8H ÀS 18H

LOCAL

SÃO PAULO, BRASIL

TEATRO AUGUSTA

RUA AUGUSTA, 943



reactconfbr.com.br



Valeu galera! Bons estudos!

@oieduardorabelo - twitter, github, etc



UDACITY