



Engenharia de Software

# TEST-DRIVEN DEVELOPMENT



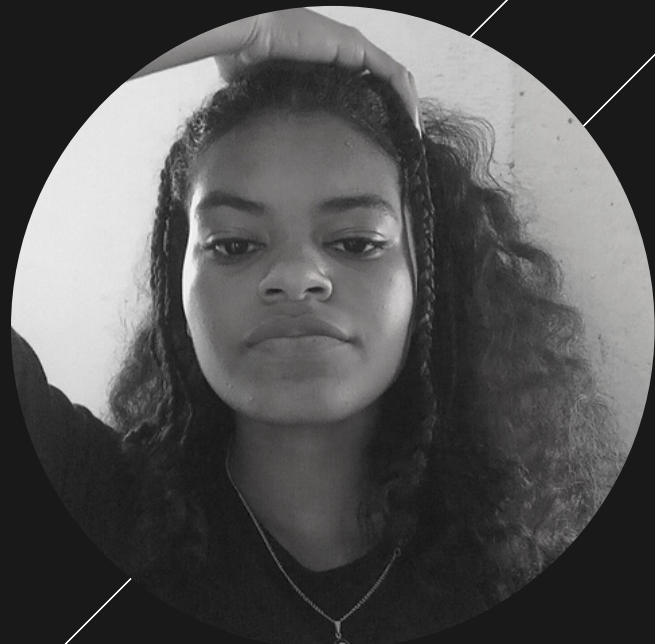
TIME



 **ALDO**



 **DOUGLAS**



 **JACKELINE**



 **ANDREY**



 **PEDRO**

# TEST-DRIVEN DEVELOPMENT

▶ Escreva seus testes antes dos códigos de produção de software



## Ciclo de testes

Utiliza uma técnica de ciclo de testes.



## Metodologia Ágil

Metodologia de desenvolvimento focada na agilidade de equipes e qualidade do produto.



## Objetivo

Códigos simples, limpos e funcionais.



# KENT BECK

Engenheiro de Software

**"TDD encoraja projetos simples e inspira confiança"**

## Agile Manifesto

Um dos 17 signatários do documento fundador do desenvolvimento ágil.

## TDD e Extreme Programming

Criou o TDD como parte do Extreme Programming, metodologia que faz o acompanhamento constante com testes e pequenos ajustes em projetos que precisam de agilidade

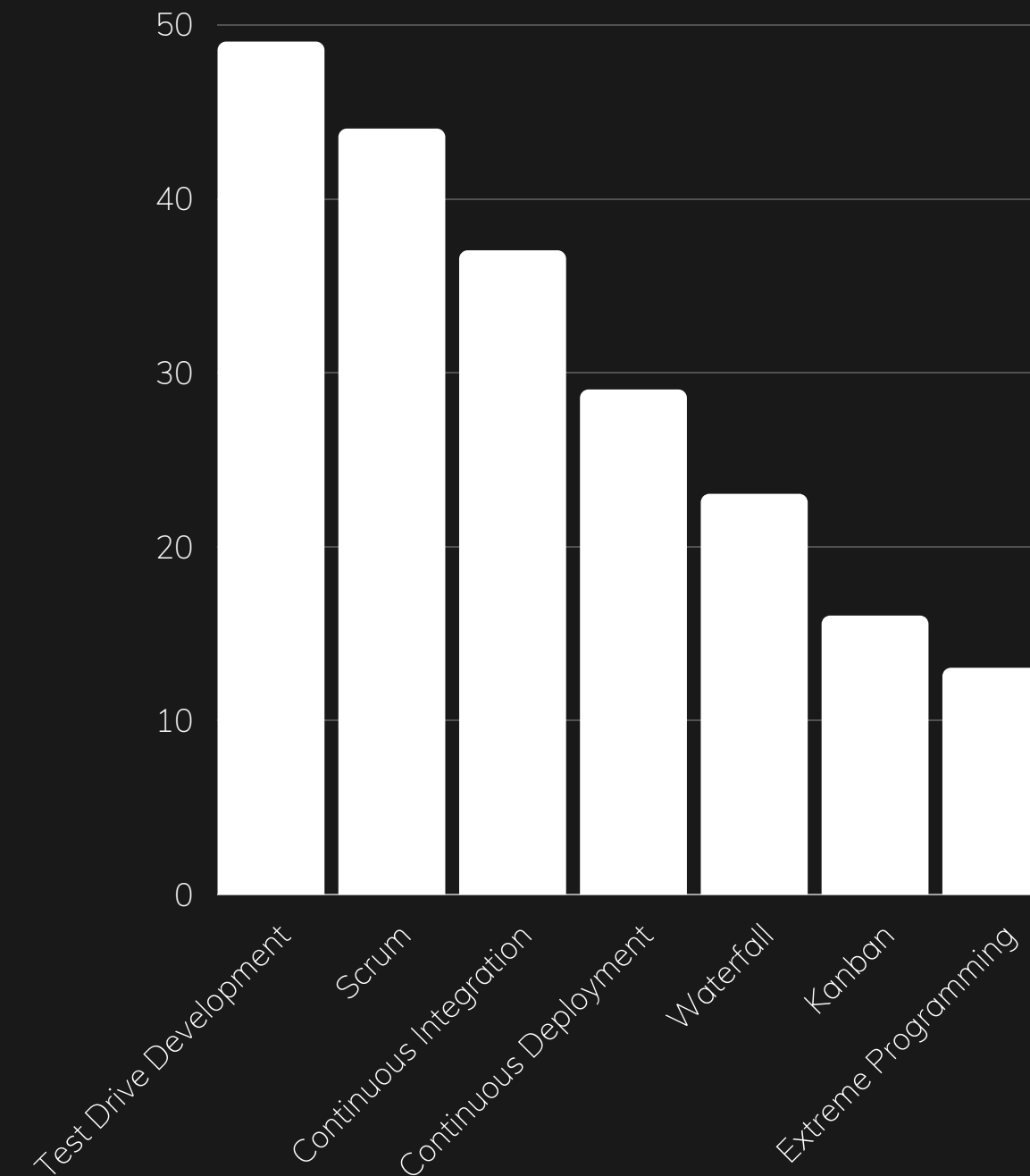
# POPULARIDADE

▶ TDD não é uma tecnologia nova, ela tem sua popularidade

X

## Estudo Open Source Language 2016

Quase metade de todos os entrevistados mencionou o TDD como tecnologia mais usada por eles



# CICLO RED-GREEN-REFACTOR

▶▶ O passo a passo para eficácia da metodologia:

**1 RED**

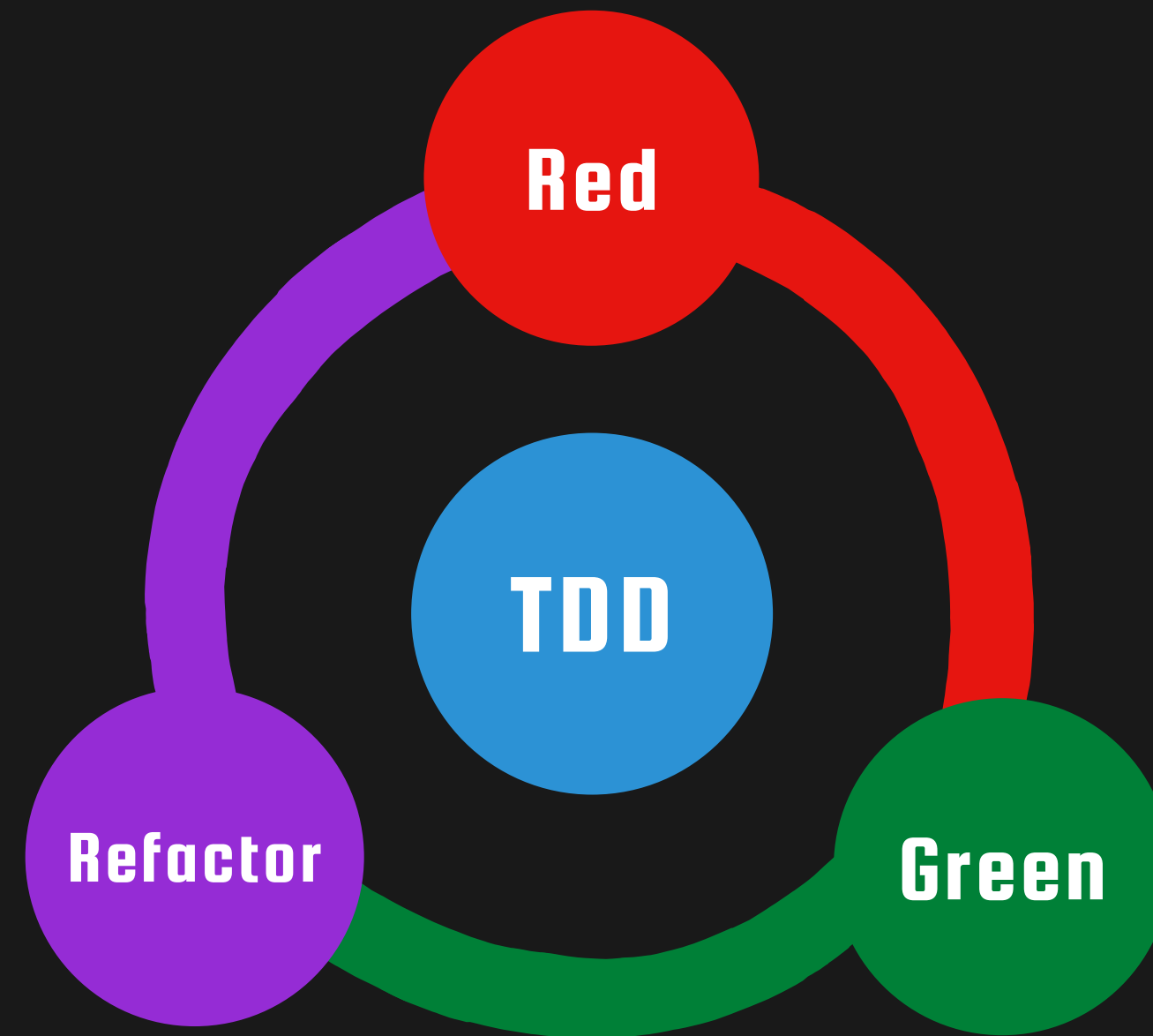
Escreva um teste que inicialmente falhe

**2 GREEN**

Implemente o código com a funcionalidade.

**3 REFACTOR**

Melhore o código, torne-o limpo.



# EXEMPLO PRÁTICO COM TDD

- ▶ Como desenvolvedor, quero implementar o código que faça uma soma

## I CRIAÇÃO DO TESTE

(RED)

```
teste.py
1 from calculadora import soma
2
3 def test_soma();
4     assert soma(2,3) == 5
```

```
def test_soma():
>     assert soma(2,3) == 5
E     NameError: global name 'soma' is not defined

test_calculadora.py:2: NameError
```

X

X

# EXEMPLO PRÁTICO COM TDD

- ▶ Como desenvolvedor, quero implementar o código que faça uma soma

## 2 IMPLEMENTAÇÃO DO CÓDIGO (GREEN)

```
calculadora.py  
1  def soma(a, b);  
2      int total = a+b  
3      return total
```

```
.calculadora.py .
```

```
———— 1 passed in 0.02 seconds ————
```

X

X



# EXEMPLO PRÁTICO COM TDD

- ▶ Como desenvolvedor, quero implementar o código que faça uma soma

3

REFATORAÇÃO

(REFACTOR)

```
calculadora.py  
1  def soma(a, b);  
2  return a+b
```

# VANTAGENS

Segue as vantagens de utilizar o TDD.

## Produtividade

Desenvolvimento apenas do necessário, com qualidade;

Alta cobertura de testes para cada funcionalidade.

## Clean Code

Refatoração dos códigos incentiva a criação de códigos limpos;

## Feedback

Maior retorno do funcionamento do software;

Divisão do projeto em etapas.

# DESVANTAGENS

Segue as desvantagens de utilizar o TDD.

## Influencia o projeto

Toda a equipe adota a metodologia ou ninguém usa

## Manter os testes

Caso haja mudança nos requisitos, será necessário refazer os testes antes dos códigos

## Processo lento

Sensação de mais trabalho pela necessidade de criação de testes.

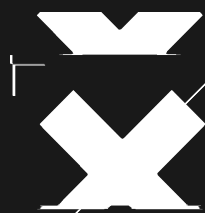


# TDD

Concluindo

Escrevendo os testes na etapa inicial, pode até parecer que acabamos tendo uma tarefa a mais a ser desempenhada; porém, no fim, seu software terá menos possibilidade de falhas, além de você acabar desenvolvendo código com mais qualidade.





# OBRIGADO!

"TDD encoraja projetos simples e inspira confiança" - Kent Beck

