

# Ruby on Rails

## Além do básico

by Jackson Pires



# Testes com Rspec - Ruby

**O que vamos aprender?**

# O que vamos aprender?

- Testes com Rspec
- ~~— Minitest~~

# O que vamos aprender?

- Testes com Rspec
- ~~— Minitest~~
- Aprender a testar!

# O que vamos aprender?

- Testes com Rspec
- ~~— Minitest~~
- Aprender a testar!
- FactoryBot
- Shoulda Matchers
- Timecop
- VCR

# O que vamos aprender?

- Testes com Rspec
- ~~— Minitest~~
- Aprender a testar!
- FactoryBot
- Shoulda Matchers
- Timecop
- VCR
- ~~— Capybara~~
- ~~— Cucumber~~

**Por que testar?**



# Por que testar?

- Garantir a qualidade do software
- Segurança e facilidade na manutenção
- Melhor design de software
- Documentação técnica

# Por que testar?

- Garantir a qualidade do software
  - Segurança e facilidade na manutenção
  - Melhor design de software
  - Documentação técnica
- 
- **Para permitir que o seu software cresça sem dores de cabeça em relação às funcionalidades existentes**

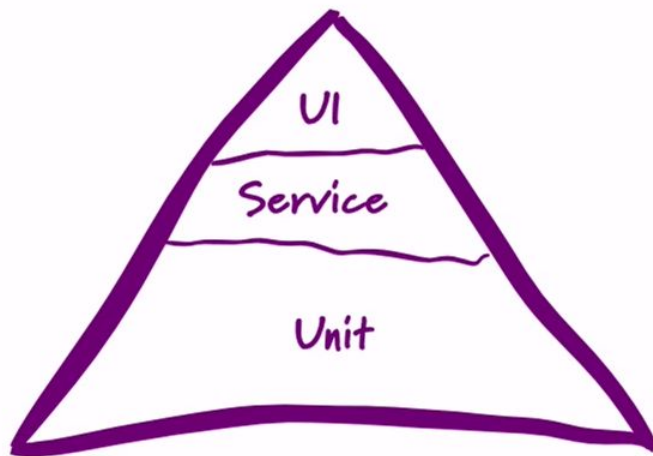
# Tipos de teste

# Tipos de teste

- Unitário
- Integração
- Sistema (código legado)
  
- [https://pt.wikipedia.org/wiki/Teste\\_de\\_software](https://pt.wikipedia.org/wiki/Teste_de_software)

# Tipos de teste

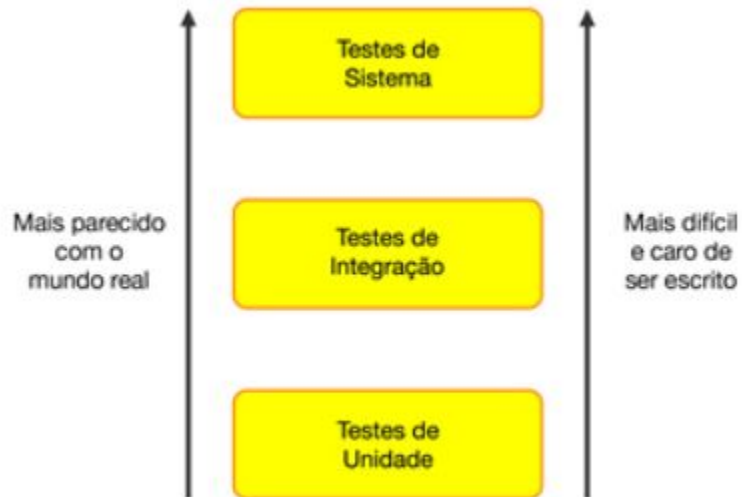
## Test Pyramid



<http://martinfowler.com/bliki/TestPyramid.html>



# Tipos de teste





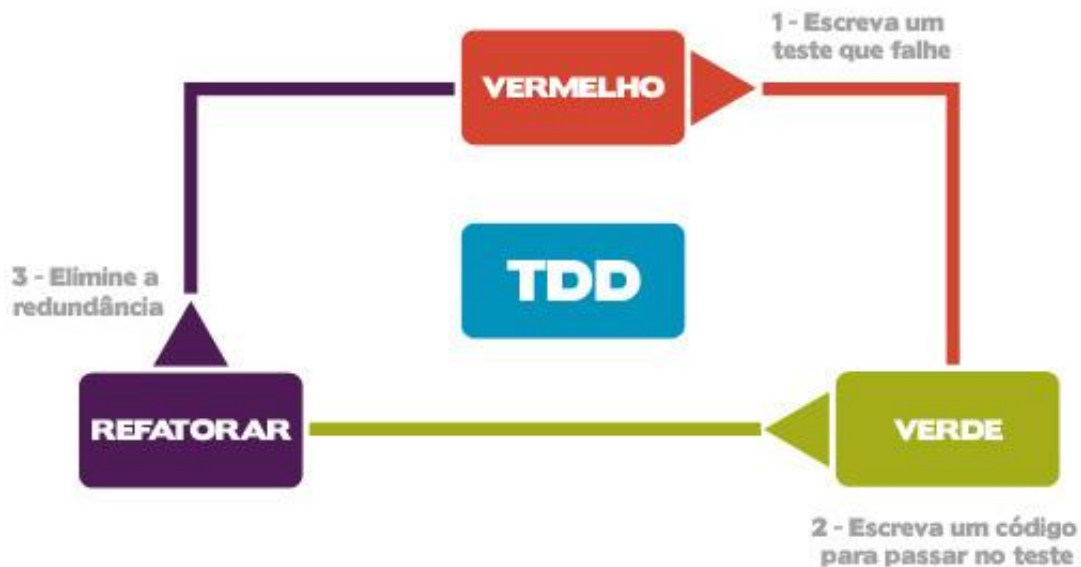
**TDD**

# TDD

- TDD: Test Driven Development
- Desenvolvimento Guiado a Testes
- [https://pt.wikipedia.org/wiki/Test-driven\\_development](https://pt.wikipedia.org/wiki/Test-driven_development)



# TDD



**Devo usar TDD sempre?**

# Devo usar TDD sempre?

- Na minha opinião, NÃO!

# Devo usar TDD sempre?

- Na minha opinião, NÃO!
- Você deve TESTAR sempre!

**Conhecendo o Rspec**

# Conhecendo o Rspec

- <http://rspec.info/>
- <https://github.com/rspec/rspec>
  
- #Gemfile
- source "<https://rubygems.org>"
- gem 'rspec'
  
- \$> bundle install
- \$> rspec --init
- \$> rspec

# Conhecendo o Rspec

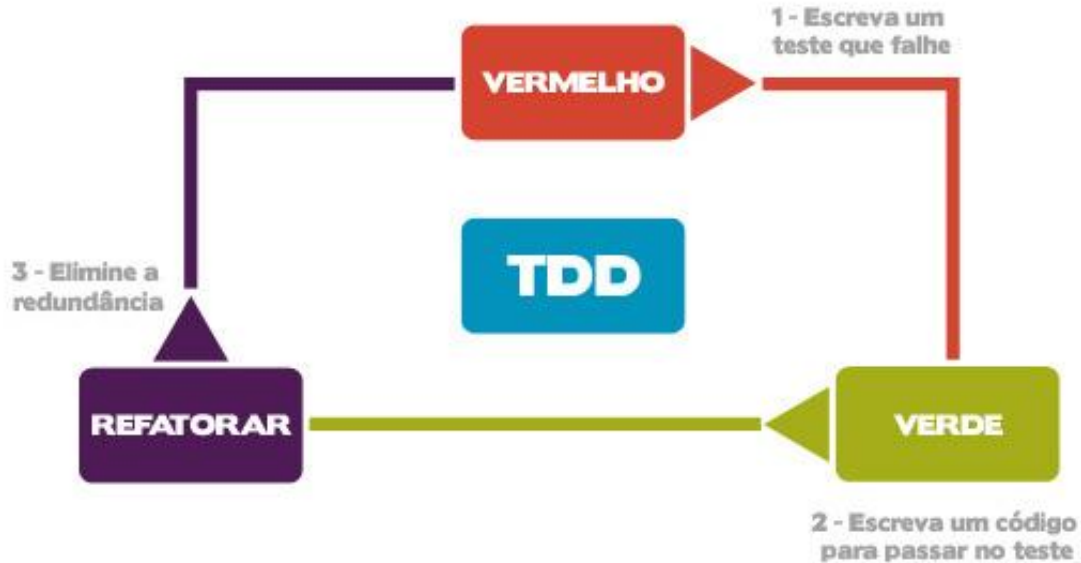
- Dica:
  - A pasta onde ficam os testes é a `/spec`
  - O comando para executar os testes é o `rspec`
  - Os testes do rspec possuem nome do arquivo com o sufixo `_spec.rb`
  - O Rspec adiciona `./lib` no `$LOAD_PATH`

# Conhecendo o Rspec

- `/spec`
  - `calculator_spec.rb`



# Conhecendo o Rspec



# Conhecendo o Rspec

- #calculator\_spec.rb

```
require 'calculator'
describe Calculator do
  end
```
- `$> rspec`

# Conhecendo o Rspec

- `/lib`
  - `calculator.rb`
    - `class Calculator`
    - `end`
- `$> rspec`

# Conhecendo o Rspec

- `/lib`
  - `calculator.rb`
    - `class Calculator`
      - `it 'add two numbers'`
    - `end`
- `$> rspec`

# Conhecendo o Rspec

```
- /lib
  - calculator.rb
    - class Calculator
      - it 'add two numbers' do
        - calc = Calculator.new
        - result = calc.sum(5,7)
        -
        - expect(result).to eq(12)
      - end
    - end

- $> rspec
```

# Conhecendo o Rspec

```
- /lib
  - calculator.rb
    - class Calculator
      - def sum
      - end
    - end

- $> rspec
```

# Conhecendo o Rspec

```
- /lib
  - calculator.rb
    - class Calculator
      - def sum(a, b)
        - a + b
      - end
    - end

- $> rspec
```

# Conhecendo o Rspec

- Organize as specs com
  - describe
  - context
  - it



# **As quatro fases de um teste**

# As quatro fases de um teste

- **Setup:** Quando você coloca o objeto sendo testado no estado necessário para o teste;
- **Exercise:** Quando você interage com o objeto que está sendo testado;
- **Verify:** Quando você verifica o comportamento esperado;
- **Teardown\*:** Quando você coloca o sistema no estado em que ele estava antes do teste ser executado;

\* Nem sempre teremos essa fase

**Como escrever bons testes?**

# Como escrever bons testes?

- Testes devem ser confiáveis
  - Testes devem ser fáceis de escrever
  - Testes devem ser fáceis de entender hoje e no futuro
- 
- Better Specs
  - <http://www.betterspecs.org/br/>

# Exercício

# Exercício

- Complemente a calculadora adicionando as operações de subtração, multiplicação e divisão, bem como implemente os testes para cada uma das operações.