

RSpec

O que é um teste unitário?

Teste é um método que contém asserções — segundo o dicionário Houaiss, asserção significa “afirmação categórica” — e que representam um cenário de testes em particular. Um teste só passará caso todas as asserções sejam verdadeiras.

Assertivas em Ruby

```
it 'is initially empty' do
  expect(@stack).to be_empty
end
```

Inicializando a estrutura do projeto

```
$ rspec --init
```

Estrutura dos Testes

```
> .  
>> /lib  
>> /spec  
>>> /spec/spec_helper.rb
```

Criamos o arquivo: spec/calculator_spec.rb

Exemplo da Calculadora

```
RSpec.describe Calculator do  
  # testes para cada método da classe  
end
```

Descrevendo contextos

```
describe '#add' do  
  # assetivas vão aqui  
end
```

E a assetiva

```
it 'returns the sum of its arguments' do
  expect(Calculator.new.add(1, 2)).to eq(3)
end
```


resultando

```
RSpec.describe Calculator do
  describe '#add' do
    it 'returns the sum of its arguments' do
      expect(Calculator.new.add(1, 2)).to eq(3)
    end
  end
end
```

Criamos a Classe

Criamos a classe `lib/calculator.rb`

```
class Calculator  
end
```

Adicionamos o método

```
class Calculator
  def add
  end
end
```

Fazermos o método

```
class Calculator
  def add a,b
    a + b
  end
end
```

Exercício

1. Fazer o teste e o método da subtração
2. Fazer o teste e o método da multiplicação
3. Fazer o teste e o método da divisão

Stubs

Exemplo da Pilha

```
RSpec.describe Factory do
  before(:example) do
    @factory = Factory.new
  end

  it 'eh criada com a pilha vazia' do
    expect(@factory.stack).to be_empty
  end

  ...
end
```

Exercício

1. Criar uma classe usando testes de uma conta bancária. Ela deve ter um método para:
 - ▶ criar uma nova conta (`#initialize`)
 - ▶ depositar dinheiro
 - ▶ sacar dinheiro
 - ▶ rendimento mensal de 3%
2. Faça testes que:
 - ▶ usar um `before` para criar um objeto de testes
 - ▶ teste que eu não possa sacar mais dinheiro do que tem na minha conta
 - ▶ o rendimento está correto depois de 3, 6 e 12 meses