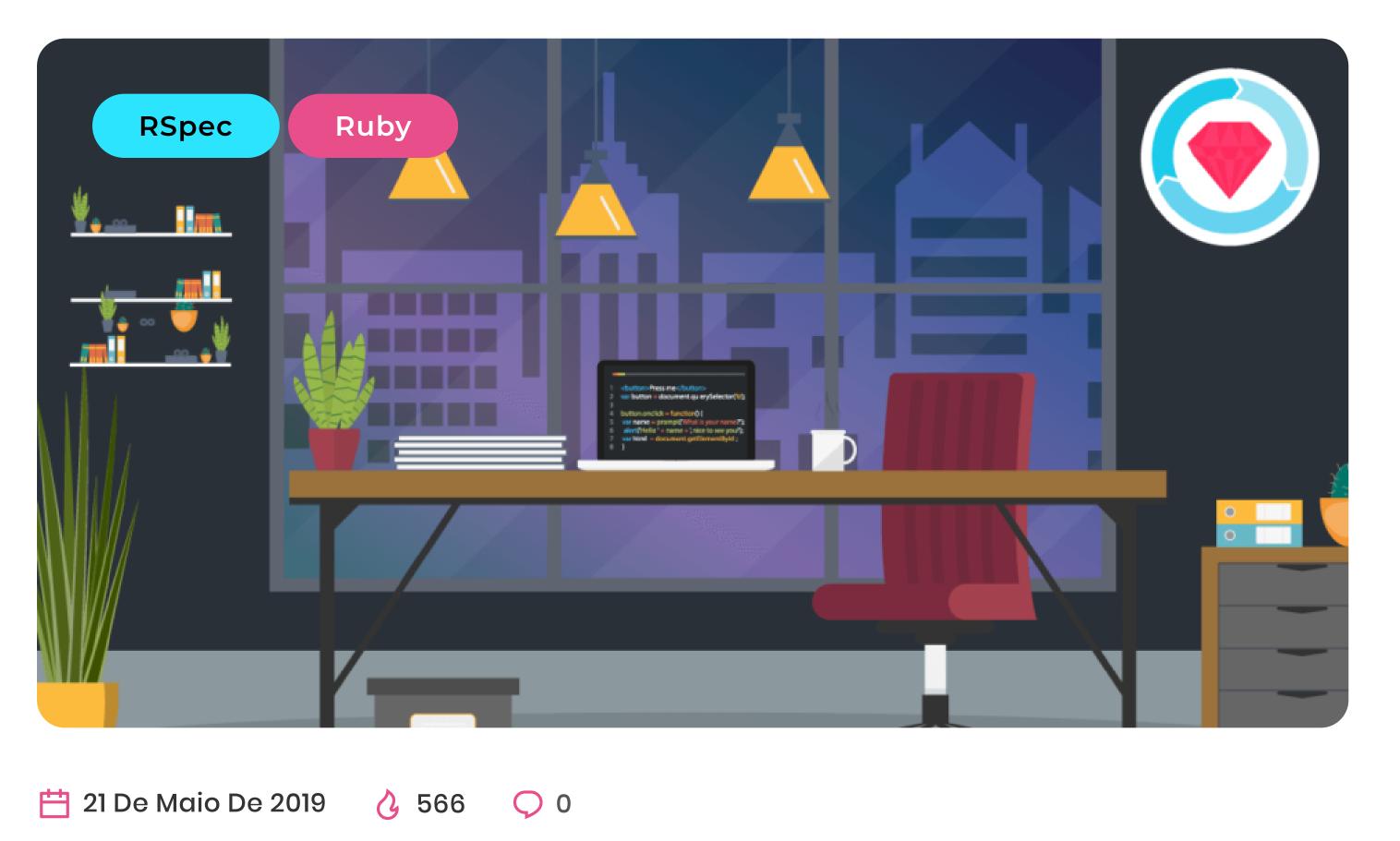
**Assuntos** 



Guia de referência para validações do

in

P

1

## RSpec O rspec é uma gem (library ou biblioteca) entre os mais

populares frameworks para a realização de testes automatizados em ruby.

Diferente de outras abordagens de teste, o rspec foca em testar comportamentos específicos ao invés de testar somente métodos, ou seja, utiliza a abordagem BDD ou Behaviour Driven Development, em tradução literal, desenvolvimento guiado a comportamentos.



expect(actual).to matcher(expected) expect(actual).to\_not matcher(expected)

```
Um exemplo para cada um dos modos de expectation pode ser:
      expect(1).to eq(1)
```

expect(1).to\_not eq(2)

Como mencionado na seção anterior, o RSpec vem com muitos built-in

```
matchers para nos ajudar a escrever testes expressivos. Vamos dar uma olhada
neles, começando pelos mais básicos, os "be matchers".
```

false. Você pode usá-los do seguinte modo:

expect(obj).to be\_false

Matchers relacionados a truthy e falsy

dois modos diferentes:

# passa se obj é truthy (não nil ou não false) expect(obj).to be\_true # passa se obj é falsy (nil ou false)

Os be matchers servem para você testar se um objeto é avaliado como true ou

```
# passa se obj é nil
      expect(obj).to be_nil
      # passa se obj é truthy (não nil e não false)
      expect(obj).to be_truthy
Os matchers de equidade
Os equality matchers servem para verificar se um objeto é igual a outro objeto.
```

Existe mais de um matcher para esse tipo de verificação porque no Ruby existe

mais de um modo de checar a equidade entre dois objetos, que são os

## seguintes

expect(a).to equal(b) # passa se a.equal?(b) expect(a).to be(b) # passa se a.equal?(b) expect(a).to eql(b) # passa se a.eql?(b) expect(a).to eq(b) # passa se a == b

```
Matchers relacionados a arrays
O RSpec nos oferece alguns matchers específicos para verificação de arrays. O
primeiro é o MatchArray. Esse matcher é usado para verificar se um array é
```

"igual" a outro, independente da ordem dos seus elementos. Segue um exemplo

## de uso desse matcher:

expect(array).to include(1, 2, 3, 4)

expect(array).not\_to include(5, 6, 7, 8)

expect(array).not\_to include([1, 2, 3, 4])

expect(array).not\_to include(0)

expect(array).not\_to include(5)

array = [1, 2, 3, 4]

array = [1, 2, 3, 4]expect(array).to match\_array([1, 2, 3, 4]) expect(array).to match\_array([4, 3, 2, 1]) expect(array).not\_to match\_array([1, 2, 3]) expect(array).not\_to match\_array([1, 2, 3, 4, 5])

```
Outro matcher relacionado à verificação de arrays é o Include Matcher. Você
pode usá-lo para verificar a relação de pertinência entre um ou mais elementos
e um determinado array. Segue um exemplo de uso:
      array = [1, 2, 3, 4]
      expect(array).to include(1)
      expect(array).to include(1, 2, 3)
```

Por fim, existem também os matchers start\_with e end\_with que servem para verificar se um array começa ou termina com uma sequência de elementos. Segue um exemplo de uso desses array = [1, 2, 3, 4]:

```
expect(array).to start_with(1)
      expect(array).to start_with(1, 2)
      expect(array).not_to start_with(2)
      expect(array).to end_with(4)
      expect(array).to end_with(3, 4)
      expect(array).not_to end_with(3)
Matchers relacionados a strings
```

O RSpec oferece alguns matchers relacionados a strings, sendo que alguns

deles são os mesmos usados para verificar arrays. Vamos começar olhando o

Match Matcher, que serve para verificar o valor de uma string de acordo com

## string = "qa ninja" expect(string).to match(/qa/) expect(string).to match(/inja/) expect(string).not\_to match(/ka/)

string = "qa ninja"

expect(string).to include("q")

expect(string).to include("qa")

expect(string).to start\_with("qa")

expect(string).to end\_with("ninja")

Referência: https://relishapp.com/rspec

expect(string).not\_to end\_with("qa")

expect(string).not\_to start\_with("ninja")

expect(string).not\_to match(/qaninja/)

uma expressão regular:

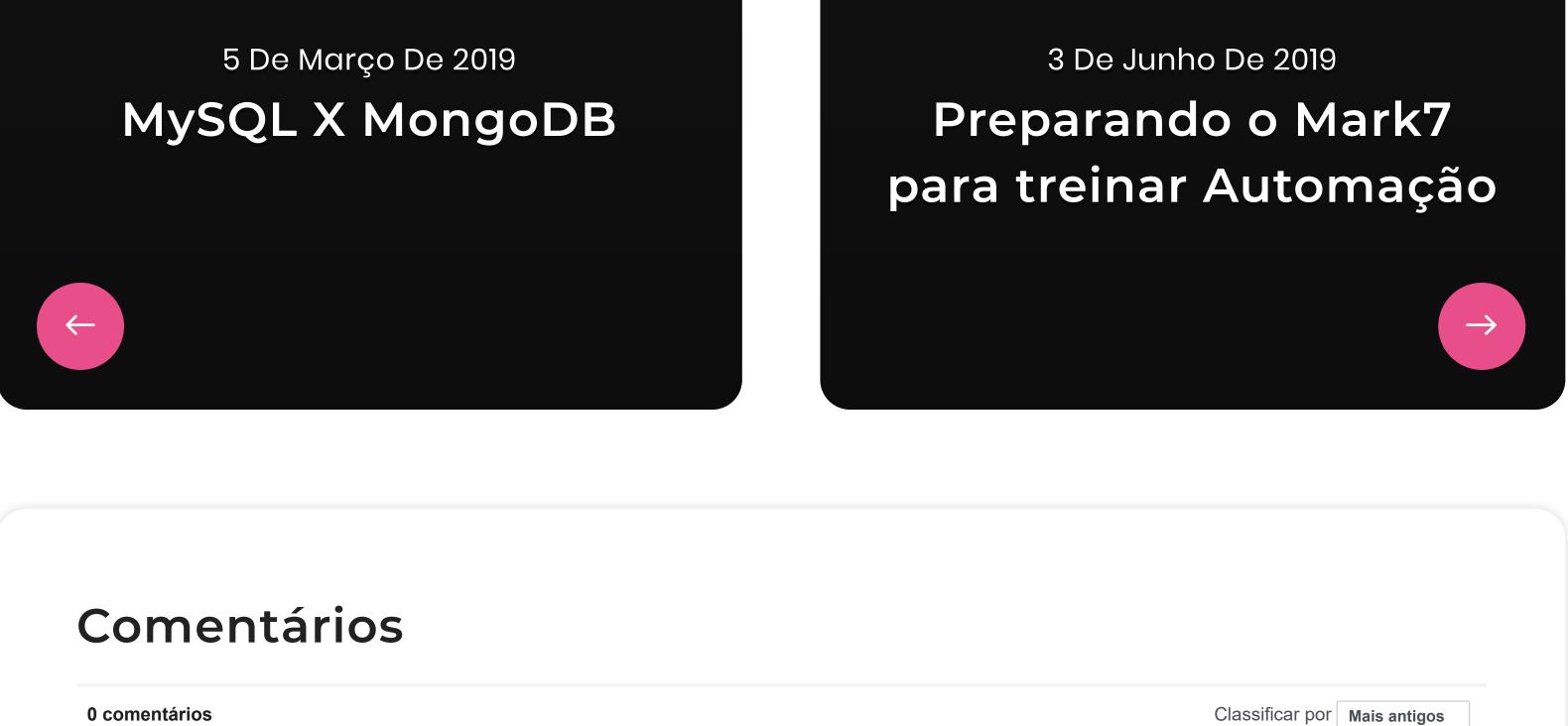
Agora seguem exemplos dos matchers usados por strings e arrays, começando pelo include:

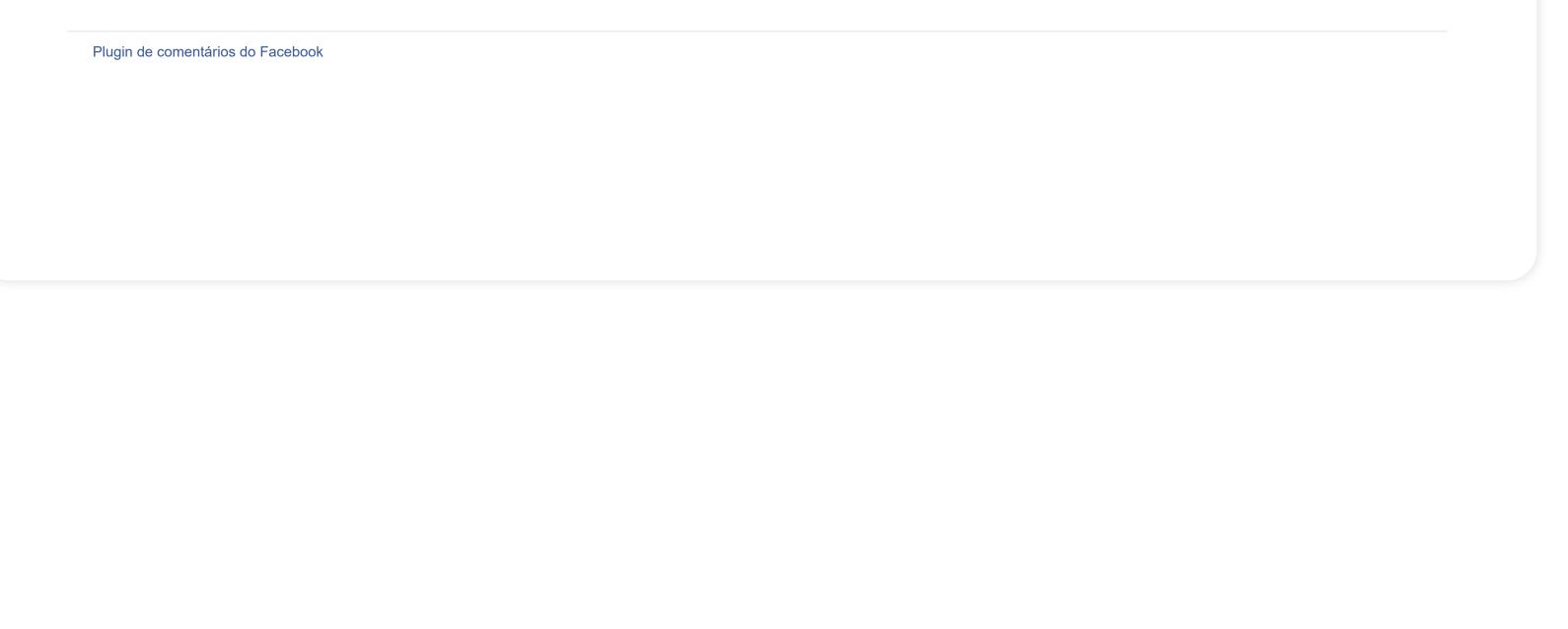
```
expect(string).to include("qa", "ninja")
      expect(string).not_to include("ka")
      expect(string).not_to include("ninjas")
Por fim, você também pode usar os matchers start_with e end_with com strings:
      string = "qa ninja"
```

Enjoy!

```
Compartilhar artigo:
               https://qaninja.io/guia-de-referencia-para-vali
automacao, rspec, ruby
```

Papito





**Android** 

Capybara

Gitlab

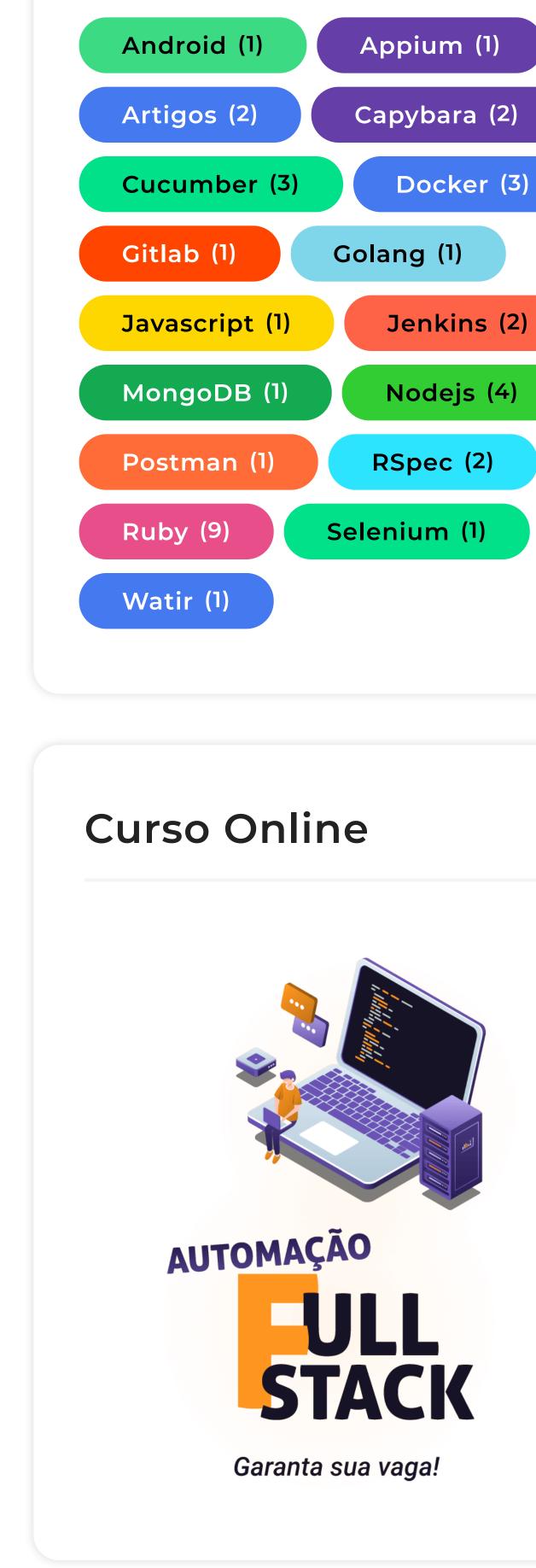
**Jenkins** 

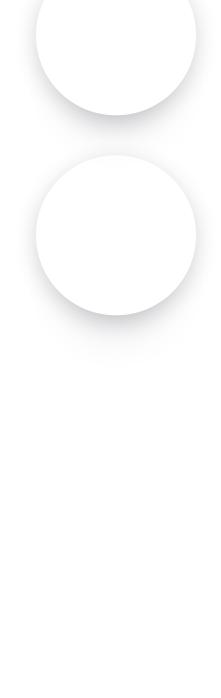
Postman

Selenium









Adicione um comentário...



Appium

MongoDB

RSpec

Watir

Golang

Cucumber

Artigos

Javascript

Ruby

Nodejs

Docker