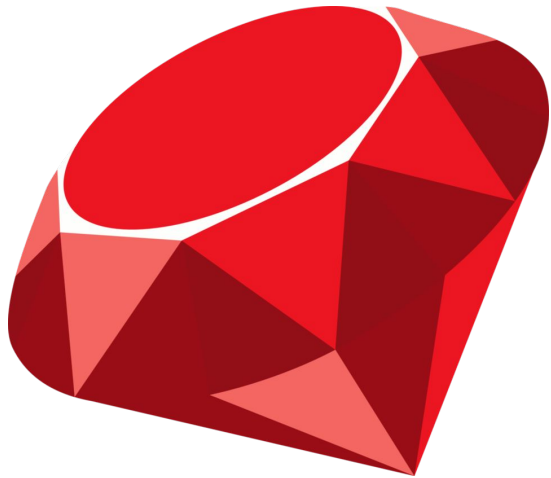


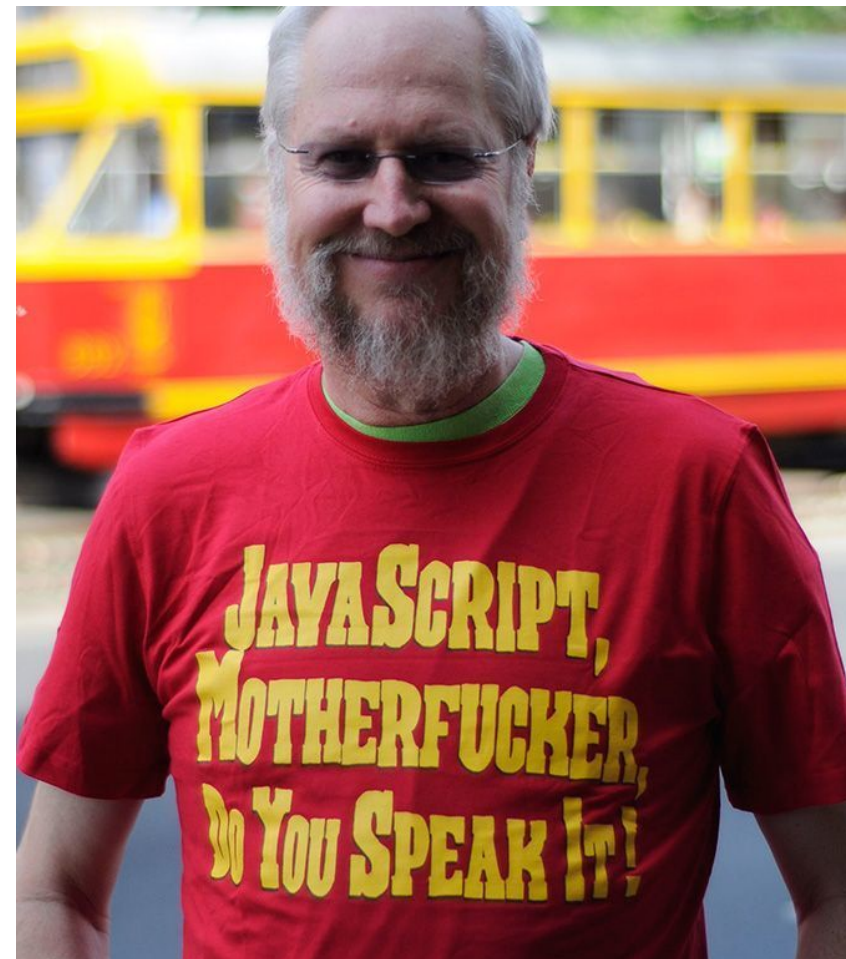
Automação de Testes API - HTTParty com CircleCI



```
{  
  "id" : 123  
  "avatar" : "data:image/png;base64,iVB...",  
  "nome" : "Rafael Berçam Medeiros",  
  "idade" : 32,  
  "profissao" : {  
    "cargo" : "Q.A.",  
    "experiencia" : 7  
  },  
  "foo" : "lêiro"  
}
```

JSON

- JavaScript Object Notation
- Douglas Crockford (2000)
- Modelo de transmissão de informações no formato **texto** bastante legível a humanos, no formato **atributo-valor**.
- Natureza auto-descritiva



O que é uma API?

{ api }

- Application Programming Interface, ou em português, Interface de programação de aplicações.



O que é uma API?

{ api }





SoapUI



POSTMAN



Insomnia



ARC



 HTTParty



Pré requisitos



- **ruby > 2.0.0**
- **gem cucumber**
- **gem multi_xml**
- **gem httparty**
- **gem rspec**
- **gem json_matchers**

Exemplo de uso HTTParty



#Acessando o método de forma rápida

```
reponse = HTTParty.get('http://end.com/end')  
puts response.body  
puts response.code  
puts response.message  
puts response.headers
```


Exemplo de uso HTTParty

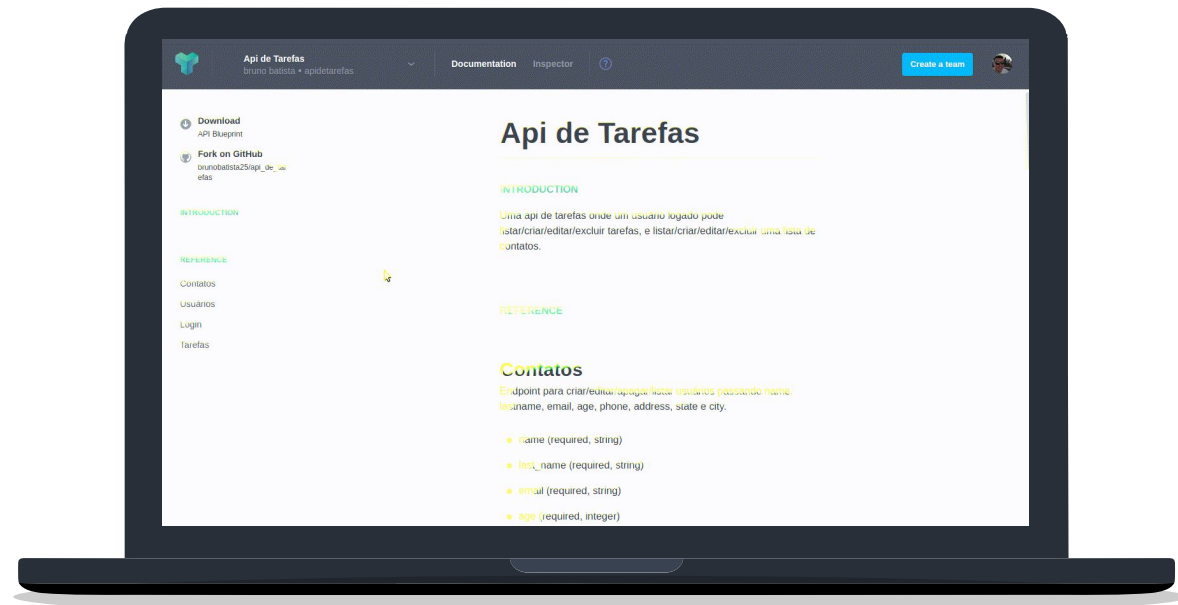


```
#Abstraindo métodos da classe
class Exemplo
  include HTTParty
  base_uri 'http://end.com'
  def initialize (headers)
    @options = {:headers => headers}
  end
  def getRequest
    self.class.get("/end",@options)
  end
end
teste = Exemplo.new("accept" => "*/*")
puts teste.getRequest
```

API de Tarefas



- Autor: Bruno Batista
- <https://apidetarefas.docs.apiary.io/#reference>
- Login
- Tarefas



Login

Endpoint para login e logout do usuário

- Ao fazer login passando o token do usuário, A API vai gerar um novo token e será com esse token que irá conseguir listar/criar/editar/apagar tarefas.
- Para deslogar o usuário so passar o token na url.
- <https://api-de-tarefas.herokuapp.com/sessions/2mDe7dXZRAkH>

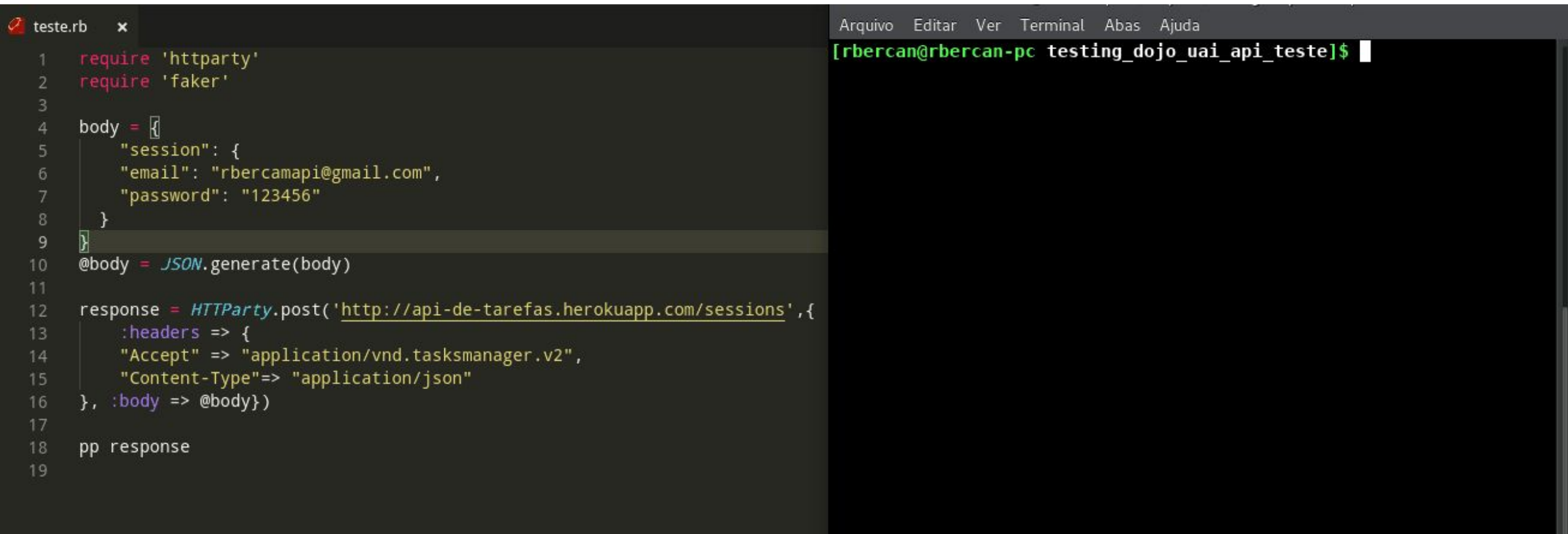
Login



Logout



Exemplo de uso HTTParty na API



The image shows a code editor window with a file named 'teste.rb' and a terminal window. The code in the editor is a Ruby script that uses the HTTParty library to perform a POST request to an API. The terminal shows the command to run the script.

```
1 require 'httparty'
2 require 'faker'
3
4 body = {
5   "session": {
6     "email": "rbercamapi@gmail.com",
7     "password": "123456"
8   }
9 }
10 @body = JSON.generate(body)
11
12 response = HTTParty.post('http://api-de-tarefas.herokuapp.com/sessions', {
13   :headers => {
14     "Accept" => "application/vnd.tasksmanager.v2",
15     "Content-Type" => "application/json"
16   }, :body => @body})
17
18 pp response
19
```

Arquivo Editar Ver Terminal Abas Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api_teste]\$

Estrutura do Projeto

- **RubyGene**

O intuito da RubyGene é conseguir gerar projetos de automação de testes já configurados para um projeto Web, Mobile ou Api. Todos eles baseados na estrutura de um projeto em cucumber, page objects e relatórios.

Instalação: `gem install rubygene`

Link: <https://github.com/brunobatista25/RubyGene>

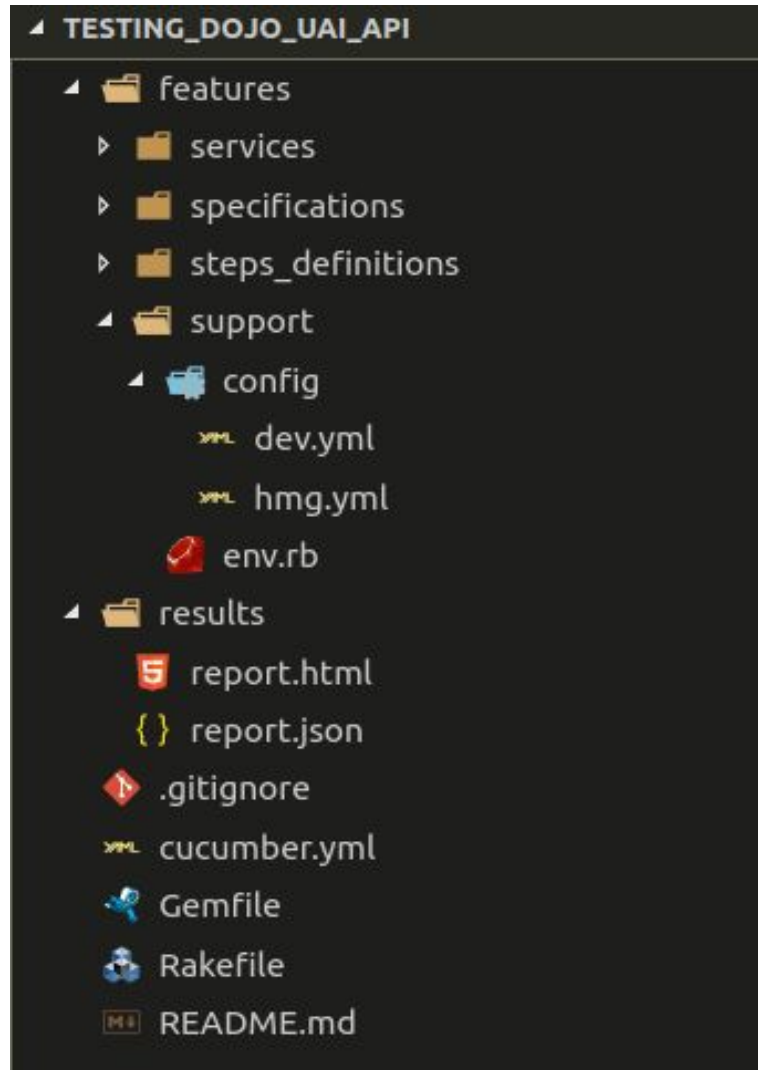
Autor: Bruno Batista

Estrutura do Projeto

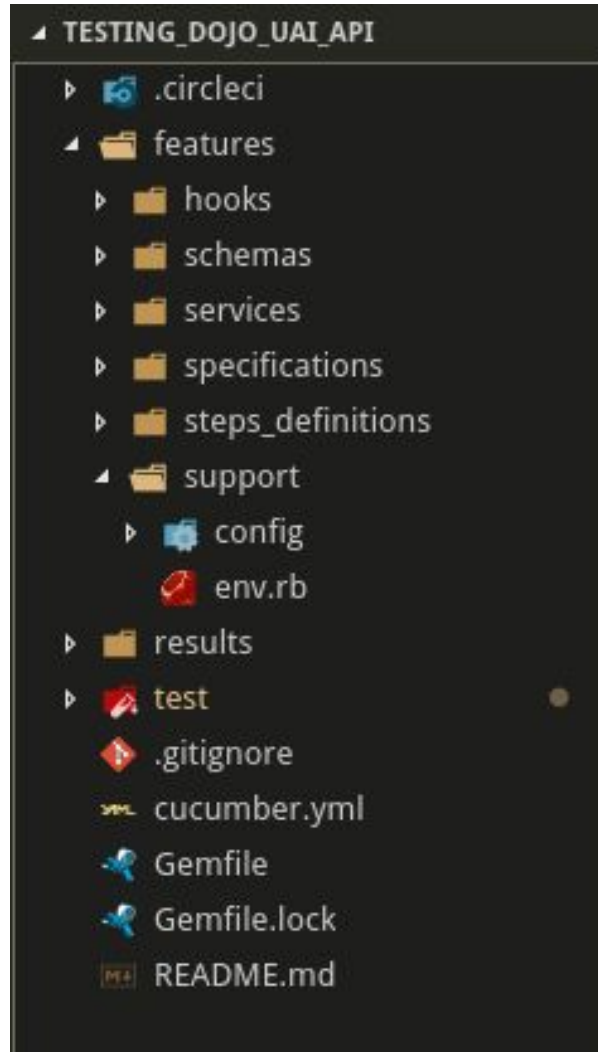


```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
rafaelbercam@rafaelbercam-Arquimedes-Corporativo-B:~/Projetos$ rubygene new_api
testing_dojo_uai_api
  create  testing_dojo_uai_api
  create  testing_dojo_uai_api/.gitignore
  create  testing_dojo_uai_api/Gemfile
  create  testing_dojo_uai_api/README.md
  create  testing_dojo_uai_api/Rakefile
  create  testing_dojo_uai_api/cucumber.yml
  create  testing_dojo_uai_api/features/services/.gitkeep
  create  testing_dojo_uai_api/features/specifications/.gitkeep
  create  testing_dojo_uai_api/features/steps_definitions/.gitkeep
  create  testing_dojo_uai_api/features/support/config/dev.yml
  create  testing_dojo_uai_api/features/support/config/hmg.yml
  create  testing_dojo_uai_api/features/support/env.rb
rafaelbercam@rafaelbercam-Arquimedes-Corporativo-B:~/Projetos$
```


Estrutura do Projeto



Estrutura do Projeto



- .circleci
- features
 - hooks
 - schemas
 - services
 - specifications
 - step_definitions
 - support
 - config
- results

Configurações

- Configurações de ambiente (support/config)

```

YAML dev.yml  x
1  base_uri: "https://api-de-tarefas.herokuapp.com"
2  email: "rbercamapi@gmail.com"
3  token: "1wpZQUuWREy_VsYaRH1Q"
4  password: "123456"

YAML hmg.yml  x
1  base_uri: "https://other"
2  email: "other@gmail.com"
3  token: "other"
4  password: "other"
5

```

Configurações

- Configurações de ambiente (support)

```
env.rb x
6 require 'date_format'
7 require 'faker'
8 require "json_matchers/rspec"
9
10 JsonMatchers.schema_root = "features/schemas/"
11
12 ENVIRONMENT = ENV['ENVIRONMENT']
13
14 CONFIG = YAML.load_file(File.dirname(__FILE__) + "/config/#{ENVIRONMENT}.yaml")
```

Configurações

- Configurações de ambiente cucumber.yml

```

cucumber.yml x
1  ## YAML Templatessss
2  ---
3  default: -p pretty -p json -p dev -p ns
4  ns: --no-source
5  pretty: --format pretty
6  ci: --format progress
7
8  html: --format html --out=results/report.html
9  json: --format json --out=results/report.json
10
11 dev: ENVIRONMENT=dev
12 hmg: ENVIRONMENT=hmg
13

```

Escrevendo uma feature



```
01-login.feature - testing_dojo
Arquivo  Editar  Seleção  Visualizar  Ir  Depurar  Terminal  Ajuda

1  #language: pt
2  @login
3  Funcionalidade: Autenticação API
4
5  Cenário: Efetuando login com sucesso
6      Quando realizar a requisição para logar na API
7      Então o sistema retorna o código 200
8      E o token da url é gerado criando a sessão
9
10 Cenário: Efetuando logout
11     Quando realizar a requisição para fazer logout na API
12     Então o sistema retorna o código 204

Terminal - rbercan@rbercan-pc:~/Projetos/testing_dojo_uai_api_teste
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$
```

Criando 'steps'



Can't see shit

```
login_steps.rb x
1 Quando("realizar a requisição para logar na API") do
2   $payload = HTTParty.post(
3     "https://api-de-tarefas.herokuapp.com/sessions",
4     {
5       :headers => {"Content-Type" => "application/json",
6                   "Accept" => "application/vnd.taskmanager.v2"},
7       :body => JSON.generate({
8         "session": {
9           "email": CONFIG["email"],
10          "password": CONFIG["password"]
11        }
12      })
13    }
14  )
15 end
16
17 Então("o sistema retorna o código {int}") do |int|
18   expect($payload.code).to eq(int)
19 end
20
21 Então("o token da url é gerado criando a sessão") do
22   puts $payload['data']['attributes']['auth-token']
23 end
24
25 Quando("realizar a requisição para fazer logout na API") do
26   $token = $payload['data']['attributes']['auth-token']
27   $payload = HTTParty.delete(
28     "https://api-de-tarefas.herokuapp.com/sessions/#{$token}",
29     {
30       :headers => {"Content-Type" => "application/json",
31                   "Accept" => "application/vnd.taskmanager.v2"}
32     }
33   )
34 end
```



Criando 'steps'



```
login_steps.rb x
1 Quando("realizar a requisição para logar na API") do
2     $payload = HTTParty.post(
3         "https://api-de-tarefas.herokuapp.com/sessions",
4         {
5             :headers => {"Content-Type" => "application/json",
6                           "Accept" => "application/vnd.taskmanager.v2"},
7             :body => JSON.generate({
8                 "session": {
9                     "email": CONFIG["email"],
10                    "password": CONFIG["password"]
11                }
12            })
13        }
14    )
15 end
16
```

Criando 'steps'



```
17 Então("o sistema retorna o código {int}") do |int|
18   expect($payload.code).to eq(int)
19 end
20
21 Então("o token da url é gerado criando a sessão") do
22   puts $token = $payload['data']['attributes']['auth-token']
23 end
24
25 Quando("realizar a requisição para fazer logout na API") dos
26   $payload = HTTParty.delete(
27     "https://api-de-tarefas.herokuapp.com/sessions/#{$token}",
28     :headers => {"Content-Type" => "application/json",
29                 "Accept" => "application/vnd.taskmanager.v2"}
30   )
31 end
```

Rodando o protótipo



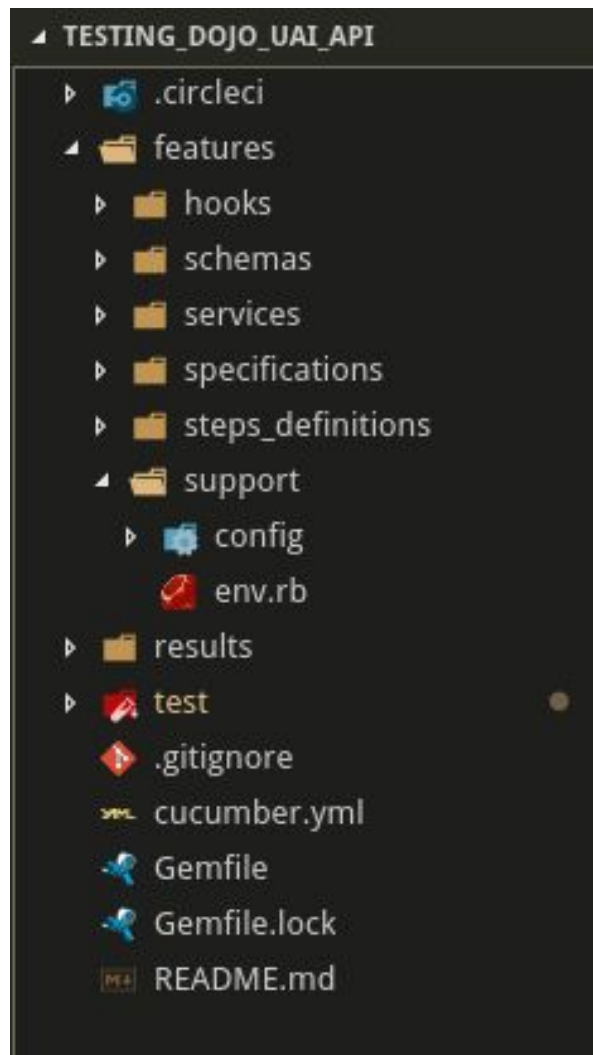
```
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$ cucumber
Using the default, pretty, json, dev and ns profiles...
# language: pt
@login
Funcionalidade: Autenticação API

  Cenário: Efetuando login com sucesso
    Quando realizar a requisição para logar na API
    Então o sistema retorna o código 200
    E o token da url é gerado criando a sessão
      gZJArA1RTybFezmUXarR

  Cenário: Efetuando logout
    Quando realizar a requisição para fazer logout na API
    Então o sistema retorna o código 204

2 scenarios (2 passed)
5 steps (5 passed)
0m2.070s
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$
```


Estrutura do Projeto



- .circleci
- features
 - hooks
 - schemas
 - services
 - specifications
 - step_definitions
 - support
 - config
- results

Hooks

- Vamos criar um arquivo 'login_hooks.rb',

```
01-login_hooks.rb •
1  Before '@login' do
2      body = {
3          "session": {
4              "email": CONFIG["email"],
5              "password": CONFIG["password"]
6          }
7      }
8      @body = JSON.generate(body)
9
10     @headers = {
11         "Content-Type" => "application/json",
12         "Accept" => "application/vnd.taskmanager.v2"
13     }
14 end
```

Ajustando os Steps



```
login_steps.rb x
1 Quando("realizar a requisição para logar na API") do
2   $payload = HTTParty.post(
3     "https://api-de-tarefas.herokuapp.com/sessions",{
4       :headers => @headers,
5       :body => @body
6     }
7   )
8 end
9
10 Então("o sistema retorna o código {int}") do |int|
11   expect($payload.code).to eq(int)
12 end
13
14 Então("o token da url é gerado criando a sessão") do
15   puts $token = $payload['data']['attributes']['auth-token']
16 end
17
18 Quando("realizar a requisição para fazer logout na API") do
19   $payload = HTTParty.delete(
20     "https://api-de-tarefas.herokuapp.com/sessions/#{$token}",
21     :headers => @headers
22   )
23 end
```

Rodando o protótipo



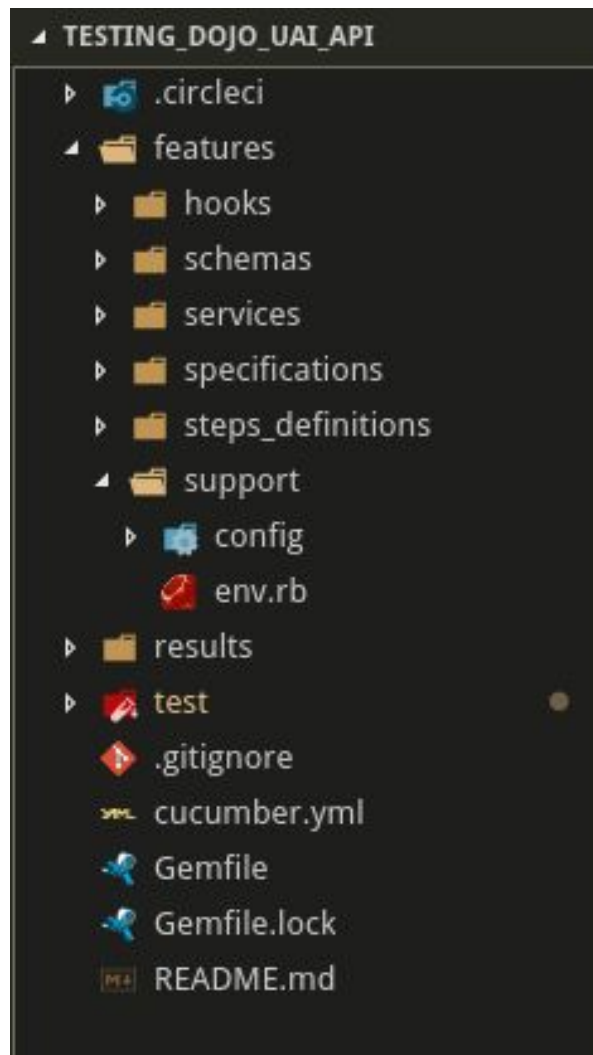
```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$ cucumber
Using the default, pretty, json, dev and ns profiles...
# language: pt
@login
Funcionalidade: Autenticação API

  Cenário: Efetuando login com sucesso
    Quando realizar a requisição para logar na API
    Então o sistema retorna o código 200
    E o token da url é gerado criando a sessão
      Ubbm1cLCTKtjGkuswyRH

  Cenário: Efetuando logout
    Quando realizar a requisição para fazer logout na API
    Então o sistema retorna o código 204

2 scenarios (2 passed)
5 steps (5 passed)
0m2.514s
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$
```

Estrutura do Projeto



- .circleci
- features
 - hooks
 - schemas
 - services
 - specifications
 - step_definitions
 - support
 - config
- results

Services

- Vamos criar o arquivo 'login_services.rb'

```
01-login_services.rb x
1  class Login
2    include HTTParty
3    base_uri CONFIG['base_uri']
4    def initialize(headers, body)
5      | @options = {:headers => headers, :body => body}
6    end
7
8    def post_login
9      | self.class.post("/sessions", @options)
10   end
11
12   def delete_logout(token)
13     | self.class.delete("/sessions/#{token}", @options)
14   end
15 end
16
```


Instanciar variável da classe no Hooks

```
01-login_hooks.rb x
1  Before '@login' do
2      body = {
3          "session": {
4              "email": CONFIG["email"],
5              "password": CONFIG["password"]
6          }
7      }
8      @body = JSON.generate(body)
9
10     @headers = {
11         "Content-Type" => "application/json",
12         "Accept" => "application/vnd.taskmanager.v2"
13     }
14
15     @login = Login.new(@headers, @body)
16 end
```

Ajustando os Steps

```
login_steps.rb x
1 Quando("realizar a requisição para logar na API") do
2   $payload = @login.post_login
3 end
4
5 Então("o sistema retorna o código {int}") do
6   expect($payload.code).to eq(int)
7 end
8
9 Então("o token da url é gerado criando a ses
10   puts $token = $payload['data']['attribu
11 end
12
13 Quando("realizar a requisição para fazer log
14   $payload = @login.delete_logout($token)
15 end
```



Rodando o protótipo



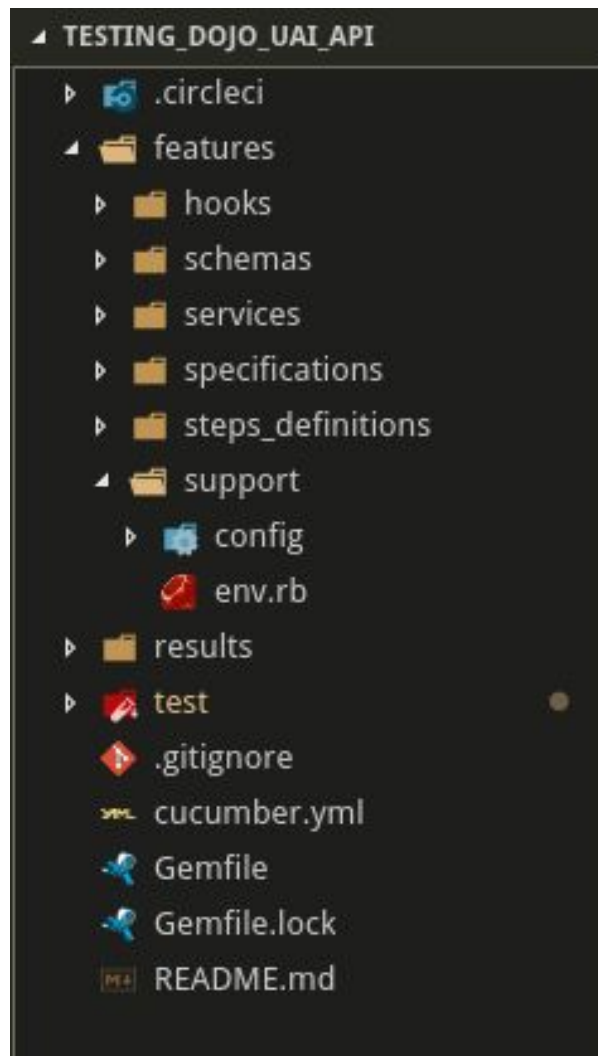
```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$ cucumber
Using the default, pretty, json, dev and ns profiles...
# language: pt
@login
Funcionalidade: Autenticação API

  Cenário: Efetuando login com sucesso
    Quando realizar a requisição para logar na API
    Então o sistema retorna o código 200
    E o token da url é gerado criando a sessão
      ox5uFWxxVdcruvssZnNV

  Cenário: Efetuando logout
    Quando realizar a requisição para fazer logout na API
    Então o sistema retorna o código 204

2 cenários (2 passed)
5 steps (5 passed)
0m2.353s
```

Estrutura do Projeto



- .circleci
- features
 - hooks
 - schemas
 - services
 - specifications
 - step_definitions
 - support
 - config
- results

- Tem como objetivo garantir que o conteúdo fornecido não tenha sido modificado.
- **É possível validar** se os dados continuam do mesmo **tipo**, como os valores **limites**, a restrição de **valores** recebidos, se a **estrutura** foi ou não modificada etc.
- **Contrato** firmado em Client / Server

```

1  {
2    "description": "schema response login",
3    "type": "object",
4    "properties": {
5      "data": {
6        "type": "object",
7        "properties": {
8          "id": {
9            "type": "string"
10         },
11         "type": {
12           "type": "string"
13         },
14         "attributes": {
15           "type": "object",
16           "properties": {
17             "email": {
18               "type": "string"
19             },
20             "auth-token": {
21               "type": "string"
22             },
23             "created-at": {
24               "type": "string"
25             },
26             "updated-at": {
27               "type": "string"
28             }
29           }
30         }
31       }
32     }
33   }

```

Schemas



```
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$
```

```

{"data"=>
  {"id"=>"25",
    "type"=>"users",
    "attributes"=>
      {"email"=>"rbercamapi@gmail.com",
        "auth-token"=>"JLxWybxe8ebfNKSTDKtz",
        "created-at"=>"2018-10-06T01:31:58.010Z",
        "updated-at"=>"2018-10-14T13:34:23.515Z"}}}

```

Schemas



EXPLORADOR

EDITORES ABERTOS

- 01-login_steps.rb features/st...

TESTING_DOJO_UAI_API_TESTE

- .circleci
- features
 - hooks
 - schemas
 - .gitkeep
 - 01-login_schema.json
 - services
 - specifications
 - steps_definitions
 - support
 - results
 - test

01-login_steps.rb x

```
1 Quando("realizar a requisição para logar na API") do
2   $payload = @login.post_login
3   expect($payload).to match_json_schema("01-login_schema")
4 end
5
6 Então("o sistema retorna o código {int}") do |int|
7   expect($payload.code).to eq(int)
8 end
9
10 Então("o token da url é gerado criando a sessão") do
11   puts $token = $payload['data']['attributes']['auth-token']
12 end
13
14 Quando("realizar a requisição para fazer logout na API") do
15   $payload = @login.delete_logout($token)
16 end
```


Rodando o protótipo



```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api_teste]$ cucumber
Using the default, pretty, json, dev and ns profiles...
# language: pt
@login
Funcionalidade: Autenticação API

  Cenário: Efetuando login com sucesso
    Quando realizar a requisição para logar na API
    Então o sistema retorna o código 200
    E o token da url é gerado criando a sessão
      LikHKcb8mCFudUssnSLr

  Cenário: Efetuando logout
    Quando realizar a requisição para fazer logout na API
    Então o sistema retorna o código 204

2 scenarios (2 passed)
5 steps (5 passed)
0m1.832s
```

- Alterando o tipo de **string** para **number**

```
"properties":{  
  "email":{  
    "type": "string"  
  },  
  "auth-token":{  
    "type": "number"  
  },  
  "created-at":{  
    "type": "string"  
  },  
  "updated-at":{  
    "type": "string"  
  }  
}
```

Rodando o protótipo | fail



```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Using the default, pretty, json, dev and ns profiles...
# language: pt
@login
Funcionalidade: Autenticação API

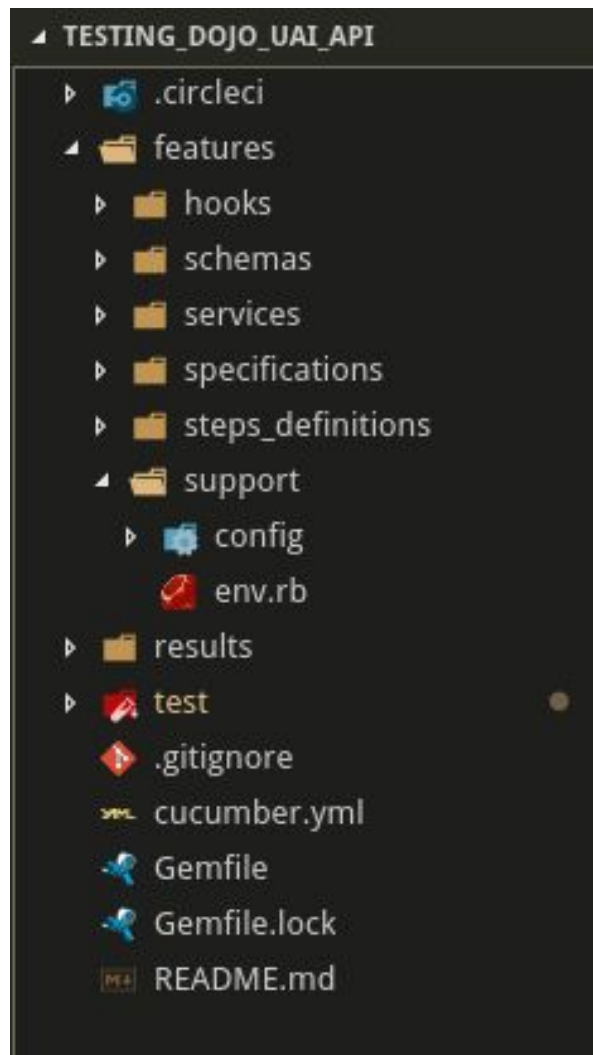
  Cenário: Efetuando login com sucesso
    Quando realizar a requisição para logar na API
      #/data/attributes/auth-token: failed schema #/properties/data
      /properties/attributes/properties/auth-token: For 'properties/auth-
token', "xcr9dQHjZA4D4kGsV-i8" is not a number.

      ---
      expected

      {
        "data": {
          "id": "25",
          "type": "users",
          "attributes": {
            "email": "rbercamapi@gmail.com",
            "auth-token": "xcr9dQHjZA4D4kGsV-i8",
            "created-at": "2018-10-06T01:31:58.010Z",
```




Estrutura do Projeto



- .circleci
- features
 - hooks
 - schemas
 - services
 - specifications
 - step_definitions
 - support
 - config
- results

Criando feature para Tarefas com Autenticação



**Api de Tarefas**
bruno batista • apidetarefas

DocumentationInspector?

Download
API Blueprint

Fork on GitHub
brunobatista25/api_de_tarefas

INTRODUCTION

REFERENCE

Contatos

Usuários

Login

Tarefas

Tarefas

- Para listar/criar/editar/excluir tarefas é preciso fazer login.
- token (required, string)
- title (required, string)
- description (optional, string)
- deadline (optional, datetime)
- done (optional, boolean)

Listar Tarefas

Criar Tarefas

Editar Tarefas

Editar uma Parte da Tarefas

Criando feature para Tarefas com Autenticação



```
02-tarefas.feature x
1  #language: pt
2  @tarefas @login
3  Funcionalidade: Manter dados de Tarefas
4
5  Contexto: Login API
6  |    Dado que o cliente esteja logado na API
7
8  Cenário: Criar uma tarefa
9  |    Quando realizar a requisição para cadastrar uma nova tarefa
10 |    Então o sistema retorna o código 201
11 |    E o corpo da resposta da tarefa
12 |    E ao final o cliente irá deslogar da API
13
14 Cenário: Listar Tarefa por id
15 |    Quando realizar a requisição para consultar tarefas
16 |    Então o sistema retorna o código 200
17 |    E o corpo da resposta da tarefa
18 |    E ao final o cliente irá deslogar da API
```

Criando feature para Tarefas com Autenticação



```
02-tarefas_hook.rb x
You, 2 days ago | 1 author (You)
1 Before '@tarefas' do
2   today_time = Time.now
3   body = {
4     "task": {
5       "title" => Faker::Lorem.characters(7),
6       "description"=> Faker::Lorem.characters(15),
7       "deadline": DateFormat.change_to(today_time, "GENERAL_DATE"),
8       "done": true
9     }
10  }
11  @body = JSON.generate(body)
12
13 end You, 2 days ago • update
```

Criando feature para Tarefas com Autenticação



```
02-tarefas_services.rb x
1  class Tarefa
2    include HTTParty
3    base_uri CONFIG['base_uri']
4
5    def initialize (body, token)
6      @options = {:headers => {
7        "Content-Type" => "application/json",
8        "Accept" => "application/vnd.taskmanager.v2",
9        "Authorization" => token
10     }, :body => body
11    }
12  end
13
14  def post_task
15    self.class.post("/tasks", @options)
16  end
17
18  def get_task(id)
19    self.class.get("/tasks/#{id}", @options)
20  end
21
```

Criando feature para Tarefas com Autenticação



```
02-tarefas_step.rb x
1  Dado("que o cliente esteja logado na API") do
2    puts $payload = @login.post_login
3    expect($payload).to match_json_schema("login_service_schema")
4    $token = $payload['data']['attributes']['auth-token']
5    @tarefa = Tarefa.new(@body, $token)
6  end
7
```


Criando feature para Tarefas com Autenticação



```
7
8 Quando("realizar a requisição para cadastrar uma nova tarefa") do
9     $payload = @tarefa.post_task
10     $id = $payload["data"]["id"]
11 end
12
13 Então("o corpo da resposta da tarefa") do
14     puts $payload
15 end
16
17 Quando("realizar a requisição para consultar tarefas") do
18     $payload = @tarefa.get_task($id)
19 end
```

Criando feature para Tarefas com Autenticação



```
20
21 Então("ao final o cliente irá deslogar da API") do
22     $payload = @login.delete_logout($token)
23     puts expect($payload.code).to eq(204)
24 end
```

Criando feature para Tarefas com Autenticação



```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
[rbercan@rbercan-pc testing_dojo_uai_api]$ cucumber
```

Hands On!



Download do Projeto



Usuário git:

```
git clone
```

```
https://github.com/rbercam/testing_dojo_uai_api.git
```

Ou Download .zip

```
https://github.com/rbercam/testing_dojo_uai_api/archive/master.zip
```




Você deve implementar:

- ✓ Método Editar Tarefas
- ✓ Método Editar Parte Tarefas
- ✓ Método Deletar Tarefas

<https://apidetarefas.docs.apiary.io/#reference/0/tarefas>


- A CircleCI foi nomeada líder em integração contínua pela **Forrester** em 2017 e foi nomeada para várias listas de melhores ferramentas de DevOps.
- Milhares de empresas líderes, incluindo **Facebook, Kickstarter, Spotify, Lyft, Palantir, Coinbase, Expedia, Stitch Fix** e **Dollar Shave Club** contam com o CircleCI para acelerar a entrega e melhorar a qualidade.



Paul Biggar • 2º
Making coding 100x easier
Baía de São Francisco e Região, Estados Unidos

✓ Pendente

Mais...

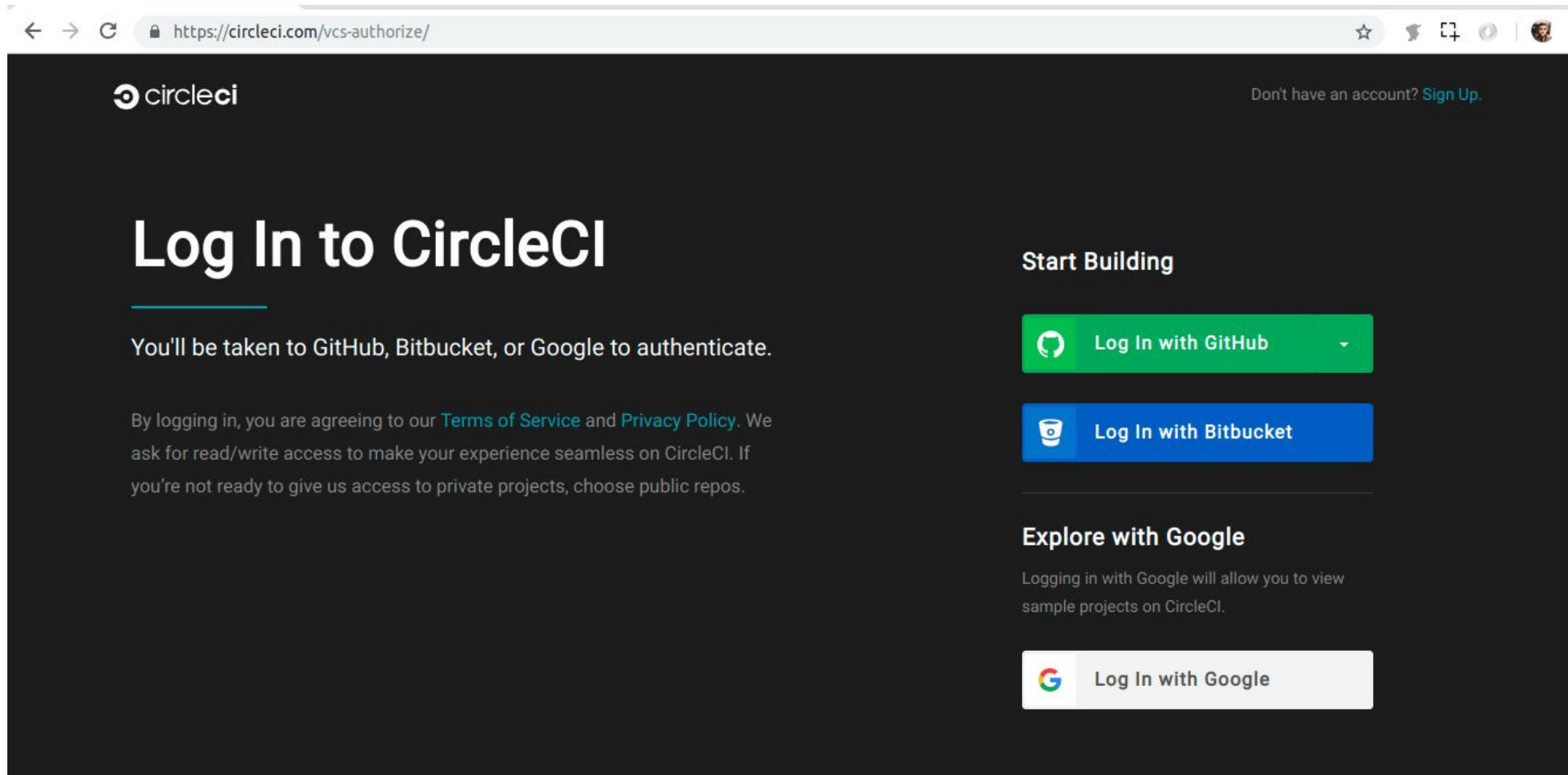


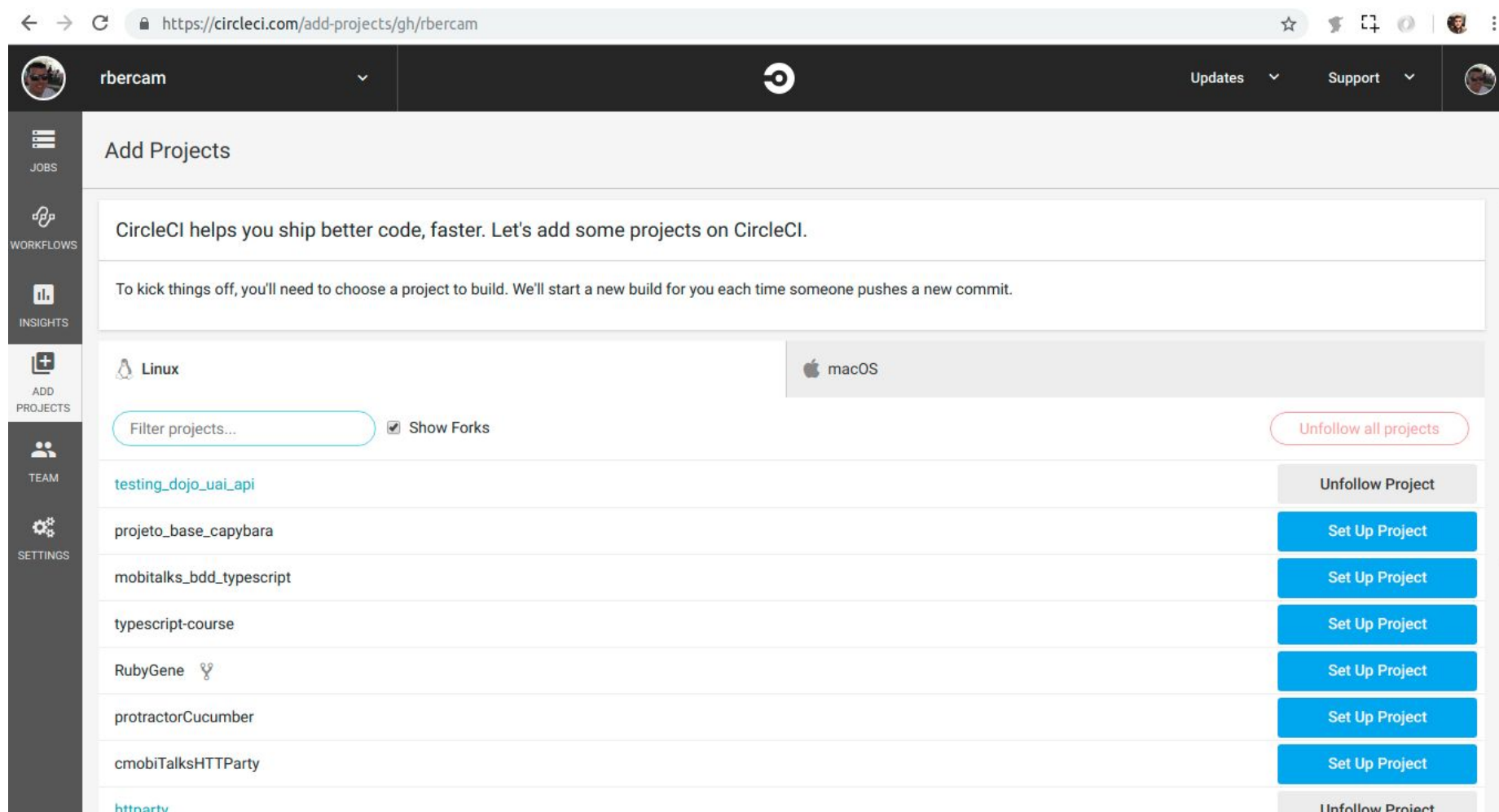
Allen Rohner • 3º
Founder, CTO at Griffin
Austin, Texas

Enviar mensagem

...

Entrepreneur, Hacker.





← → ↻ https://circleci.com/add-projects/gh/rbercam

rbercam Updates Support

Add Projects

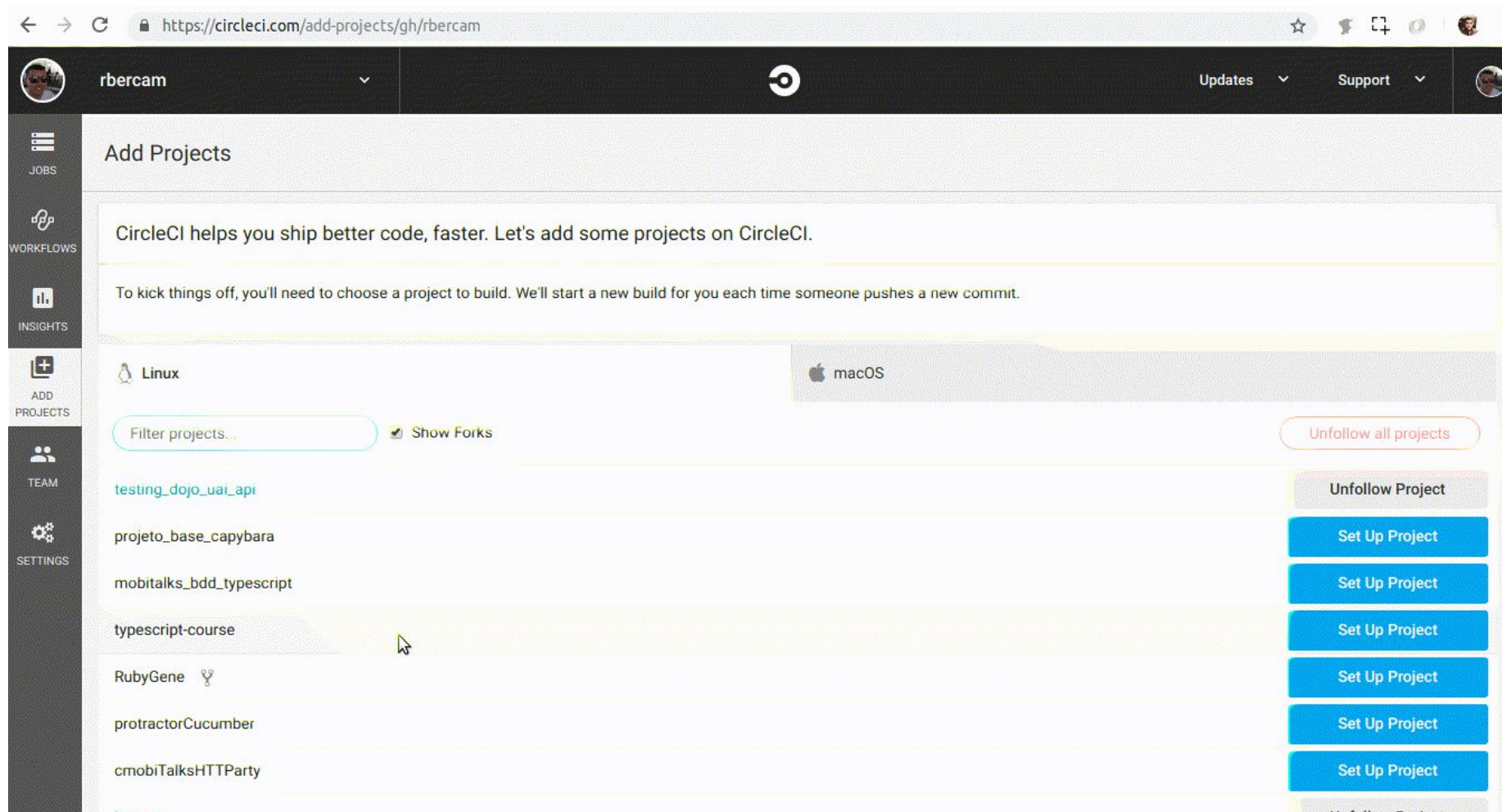
CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

Linux macOS

Filter projects... ☒ Show Forks [Unfollow all projects](#)

| | |
|--------------------------|------------------|
| testing_dojo_uai_api | Unfollow Project |
| projeto_base_capybara | Set Up Project |
| mobitalks_bdd_typescript | Set Up Project |
| typescript-course | Set Up Project |
| RubyGene | Set Up Project |
| protractorCucumber | Set Up Project |
| cmobiTalksHTTParty | Set Up Project |
| httparty | Unfollow Project |



CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

Linux macOS

Filter projects... Show Forks Unfollow all projects

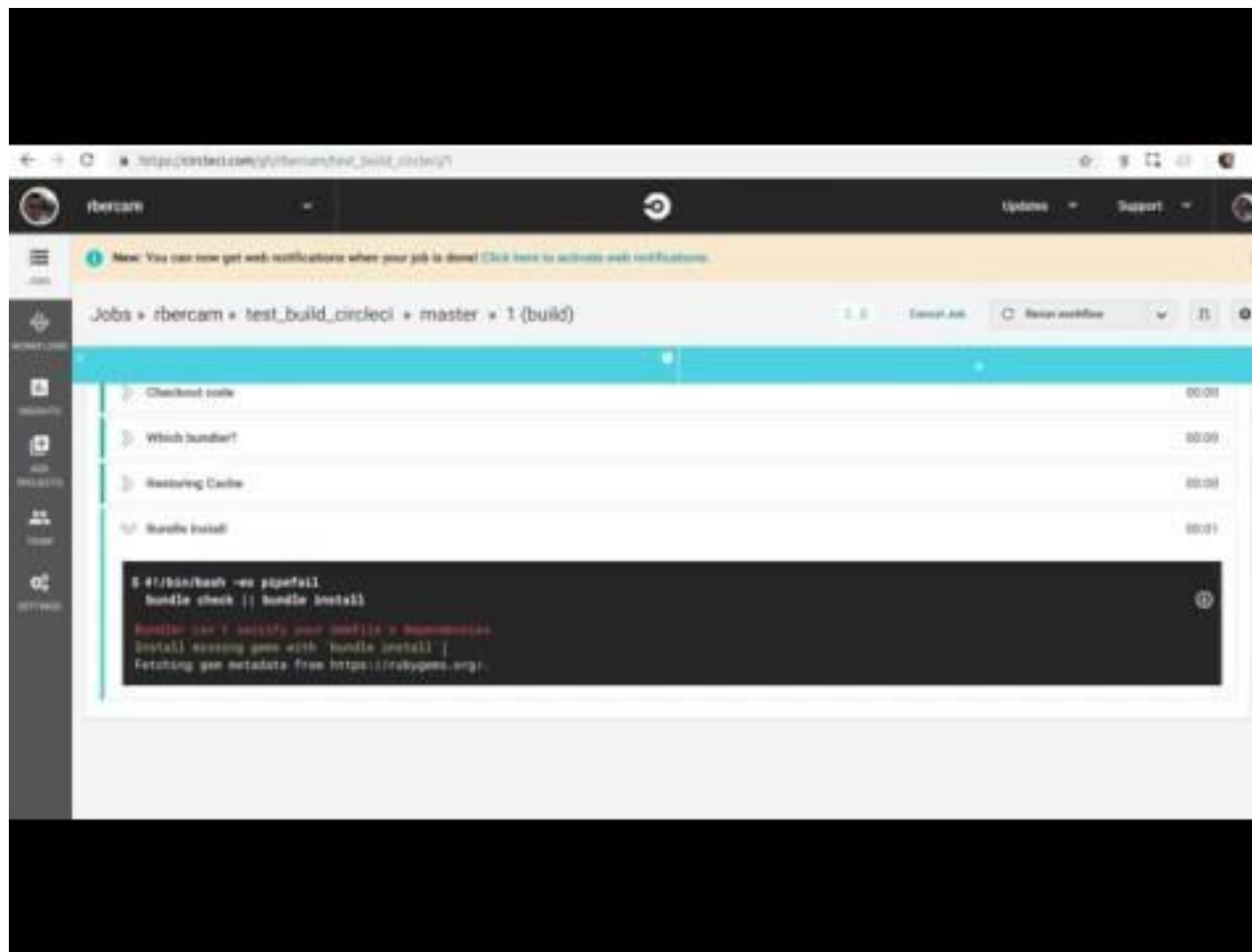
| Project Name | Action |
|--------------------------|------------------|
| testing_dojo_uai_api | Unfollow Project |
| projeto_base_capybara | Set Up Project |
| mobitalks_bdd_typescript | Set Up Project |
| typescript-course | Set Up Project |
| RubyGene | Set Up Project |
| protractorCucumber | Set Up Project |
| cmobiTalksHTTParty | Set Up Project |


```
config.yml x
1  rbercam, 3 days ago | 1 author (rbercam)
2  version: 2 # use CircleCI 2.0
3  jobs: # a collection of steps
4    build:
5      working_directory: ~/testin_dojo_uai_api # directory where steps will run
6      docker: # run the steps with Docker
7        - image: circleci/ruby:2.6-rc-node # ...with this image as the primary container; t
8          environment: # environment variables for primary container
9            BUNDLE_JOBS: 3
10           BUNDLE_RETRY: 3
11           BUNDLE_PATH: vendor/bundle
12           PGHOST: 127.0.0.1
13           PGUSER: circleci-demo-ruby
14           RAILS_ENV: test
```

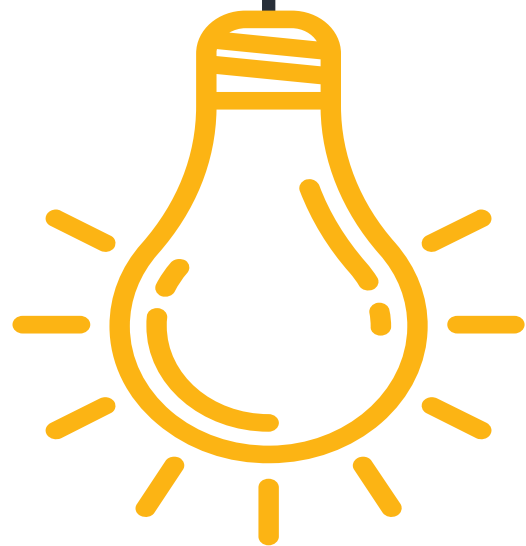
```
config.yml x
14  steps: # a collection of executable commands
15    - checkout # special step to check out source code to working directory
16    # Which version of bundler?
17    - run:
18      name: Which bundler?
19      command: bundle -v
20
21    # Restore bundle cache
22    - restore_cache:
23      keys:
24        - rails-demo-bundle-v2-{{ checksum "Gemfile.lock" }}
25        - rails-demo-bundle-v2-
26
27    - run:
28      name: Bundle Install
29      command: bundle check || bundle install
30
31    # Store bundle cache
32    - save_cache:
33      key: rails-demo-bundle-v2-{{ checksum "Gemfile.lock" }}
34      paths:
35        - vendor/bundle
```



```
config.yml x
37 - run:
38     name: Run Cucumber
39     command: |
40         bundle exec cucumber
41
42     # Save test results for timing analysis
43 - store_test_results:
44     path: ~/testin_dojo_uai_api/results
45 - store_artifacts:
46     path: ~/testin_dojo_uai_api/results
47
48 .ows:
49     sion: 2
50
51 _commits:
52     jobs:
53         - build:
54             filters:
55                 branches:
56                     ignore:
57                         - developer
58                         - test
59
60     # See https://circleci.com/docs/2.0/deployment-integrations/ for example deploy configs
```



- **Aprendemos a configurar um ambiente de projeto para automatizar APIs com entrega contínua.**
- **Como realizar as requisições e inspecionar os atributos do payload.**
- **Realizar validações de contratos de API com `json_matchers`**



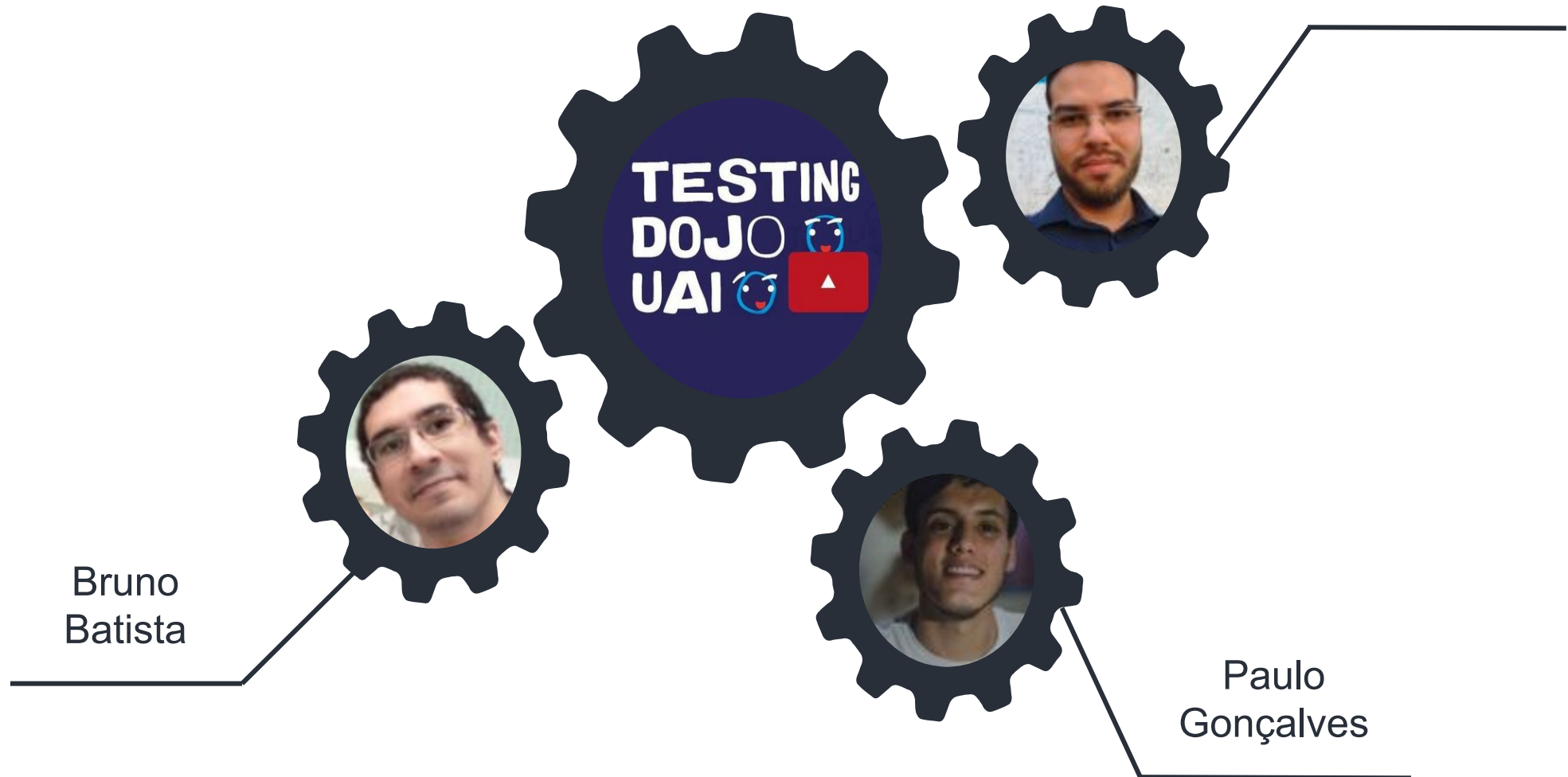
DÚVIDAS ? PERGUNTAS?

Mídias Sociais



- [LinkedIn](#)
- [GitHub](#)
- [Facebook](#)
- [Twitter](#)
- faelbercam@gmail.com

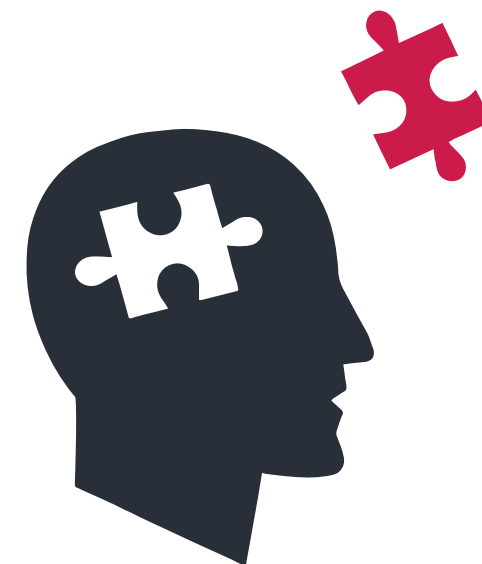
Agradecimentos





**“Busquem
conhecimento...”**

Bilu, o ET.



Referências

- <https://github.com/jnunemaker/httparty>
- <https://github.com/stympy/faker>
- <https://medium.com/@JanisGraubins/open-banking-api-explained-in-3-gifs-b806f14ca2ca>
- <https://github.com/brunobatista25/RubyGene>
- https://www.rubydoc.info/gems/json_matchers/0.6.3
- <https://circleci.com/docs/>
- https://cswr.github.io/JsonSchema/spec/basic_types/