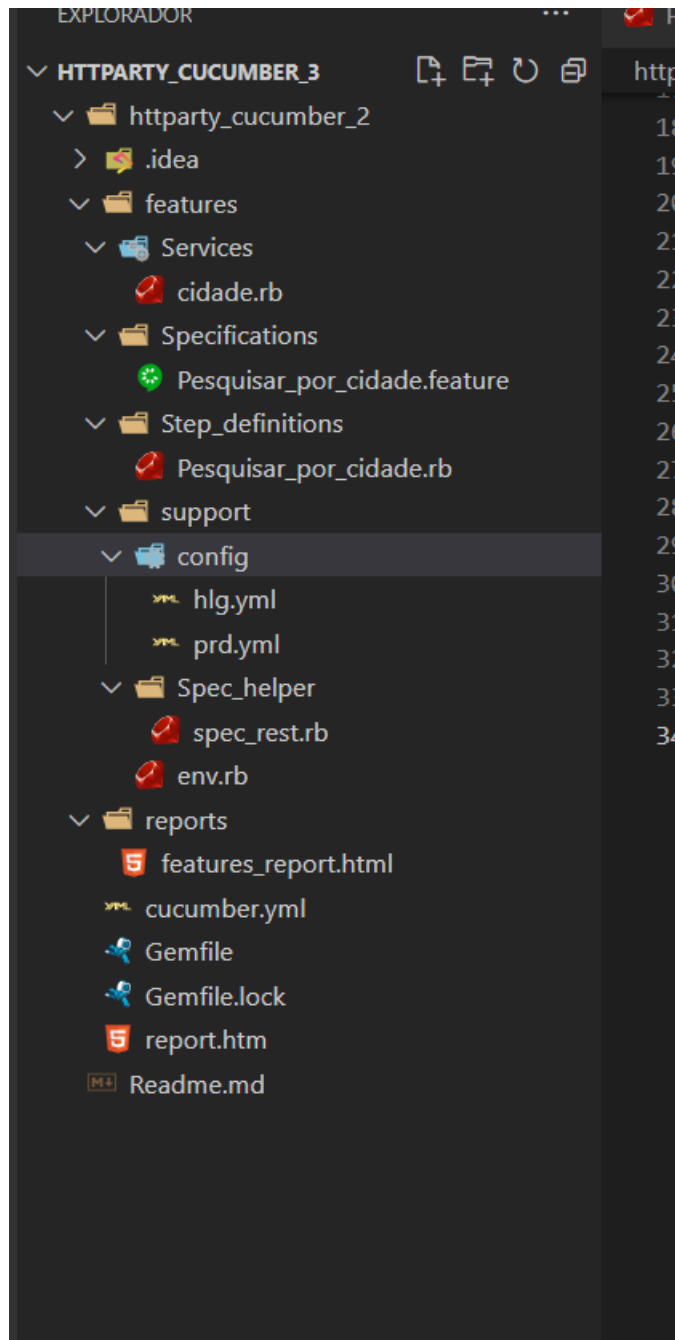


Visão Geral do projeto

- 1 – services
- 2 – specifications
- 3 – step_definitions
- 3 – services
- 4 – support
- 5 – spec helper
- 6 – reports



- 1 – services

Nesta pasta contém o código que com auxílio do HTTParty, realiza as requisições do teste, nela construímos as chamadas da API.

```
cidade.rb x
httparty_cucumber_2 > features > Services > cidade.rb
1 class Cidade
2   include HTTParty
3   base_uri CONFIG['base_uri']
4
5   def get_cidade(nome_cidade, key)
6     self.class.get('?q=' + nome_cidade + '&appid=' + key)
7   end
8
9 end
10
```

2 – specifications

Essa pasta contém o BDD do projeto

```
Pesquisar_por_cidade.feature x
httparty_cucumber_2 > features > Specifications > Pesquisar_por_cidade.feature
1 #language: pt
2 #utf-8
3
4 @Validar_clima_coordenadas
5 Funcionalidade: Validar clima e coordenadas
6
7 @cidade
8 Esquema do Cenário: Validar clima pela cidade
9 Quando gerar uma requisição do tipo get com <nome_cidade> e <key>
10 Então o retorno da requisição deve ser <retorno>
11 E o clima e coordenadas em formato Json
12
13 Exemplos:
14
15 | nome_cidade | key | retorno |
16 | 'New York' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
17 | 'California' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
18 | 'Boston' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
19
20
21 @cep
22 Esquema do Cenário: Validar clima pelo cep
23 Quando gerar uma requisição do tipo get com <zip_code> e <key>
24 Então o retorno da requisição deve ser <retorno>
25 E o clima e coordenadas em formato Json
26
27 Exemplos:
28
29 | zip_code | key | retorno |
30 | '10005' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
31 | '96162' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
32 | '05544' | 'd8861905a878ac2eca57d1b6117ccde' | 200 |
33
34
```

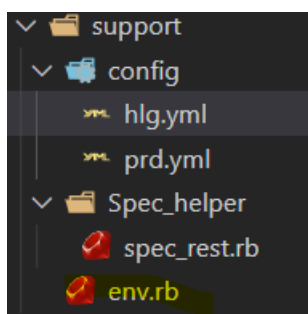
3 – step_definitions

Nesta pasta contém as informações dos cenários, de cada passa o que tem que ser executado para rodar.

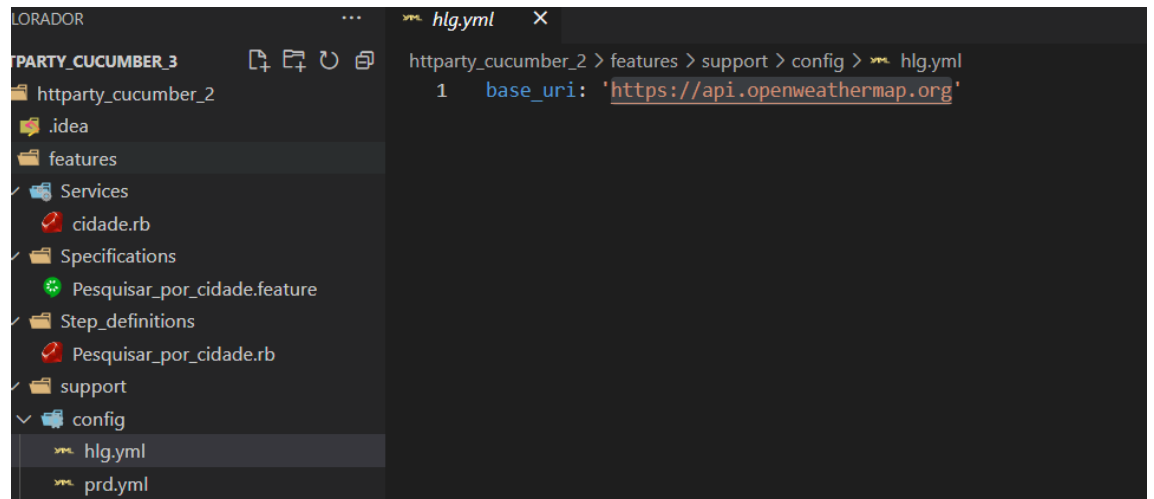
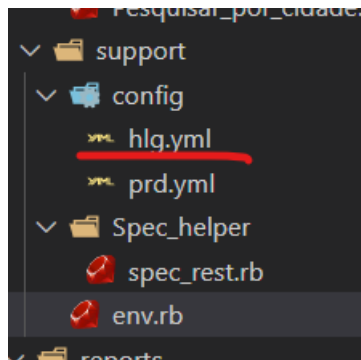
```
Pesquisar_por_cidade.rb X
httparty_cucumber_2 > features > Step_definitions > Pesquisar_por_cidade.rb
1  #encoding: UTF-8
2  Quando('gerar uma requisição do tipo get com {string} e {string}') do |nome_cidade, key|
3    @request_cidade = cidade.get_cidade(nome_cidade, key)
4  end
5
6  Então('o retorno da requisição deve ser {int}') do |retorno|
7    expect(@request_cidade.code).to eq(retorno)
8    #puts "\nResposta do Lines digital: #{@request_cidade.code}"
9    #puts "\nRetorno do Lines digital:\n #{JSON.pretty_generate()}"
10 end
11
12 Então('o clima e coordenadas em formato Json') do
13   puts JSON.pretty_generate(JSON.parse(@request_cidade.body))
14 end
15
```

4 – support

Nesta pasta contém os arquivos relacionados ao suporte do projeto, lá dentro neste caso temos o arquivos: env.rb, arquivo de massas e também configuração para ambientes (produção e homologação) e também temos o arquivo Spec_helper que contem módulos para que possa ser feita a conexão com a api.

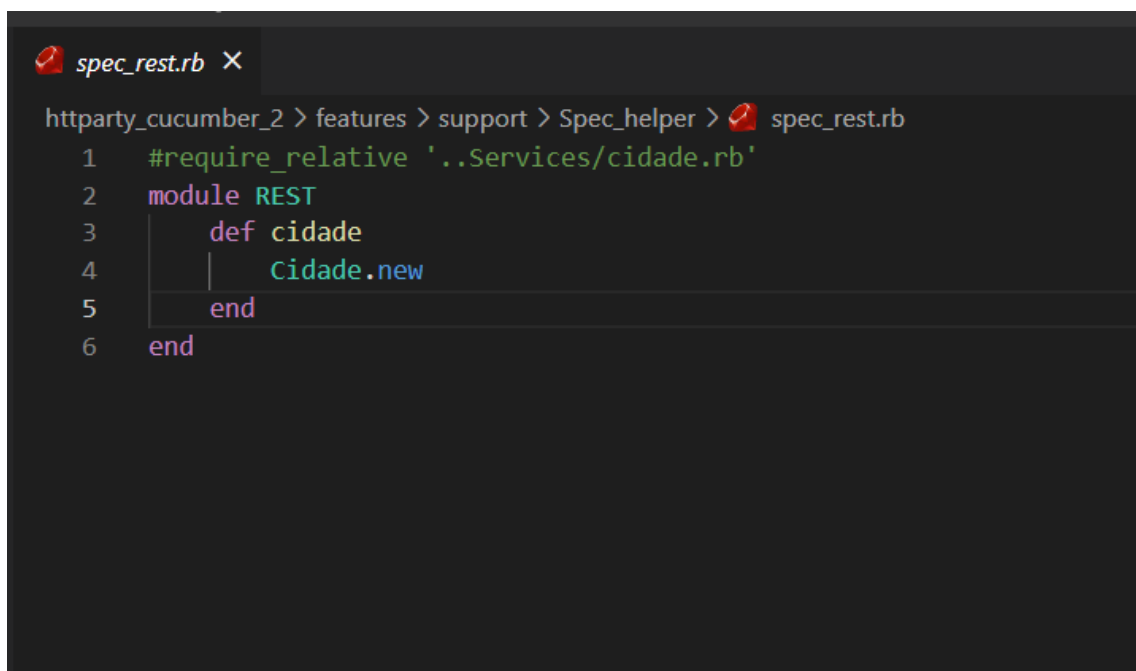


```
env.rb X
httparty_cucumber_2 > features > support > env.rb
1  require 'httparty'
2  require 'httparty/request'
3  require 'httparty/response/headers'
4  require 'rspec'
5  require 'allure-cucumber'
6  require 'cucumber'
7
8  require_relative 'Spec_helper/spec_rest'
9
10 CONFIG = YAML.load_file(File.dirname(__FILE__) + "/config/#{ENV['AMBIENTE']}.yml")
11
12
13 World(REST)
14
```



5 – spec helper

Criação de módulo do tipo rest para conexão via api



6 – reports

Utilizo reports com visão web porque visualmente são mais estruturados. Para gerar o relatório em html, utilizamos os comando: `cucumber -format html -out=report.htm`

passed

Execution summary
6 scenarios

Implementation: cucumber-ruby - 7.1.0
Runtime: ruby - 2.7.5
OS: mingw32 - 10.0.22000
CPU: x64

features/Specifications/Pesquisar_por_cidade.feature

@Validar_clima_coordenadas

Funcionalidade: Validar clima e coordenadas

@cidade

Esquema do Cenário: Validar clima pela cidade

Quando gerar uma requisição do tipo get com <nome_cidade> e <key>

Então o retorno da requisição deve ser <retorno>

E o clima e coordenadas em formato Json

Exemplos:

	nome_cidade	key	retorno
✓	'New York'	'd8861905a878ac2eca57d1b6117cccde'	200
✓	'California'	'd8861905a878ac2eca57d1b6117cccde'	200
✓	'Boston'	'd8861905a878ac2eca57d1b6117cccde'	200

@cep

Esquema do Cenário: Validar clima pelo cep

Quando gerar uma requisição do tipo get com <zip_code> e <key>

Então o retorno da requisição deve ser <retorno>

E o clima e coordenadas em formato Json

Exemplos:

	zip_code	key	retorno
✓	'10005'	'd8861905a878ac2eca57d1b6117cccde'	200
✓	'96162'	'd8861905a878ac2eca57d1b6117cccde'	200
✓	'05544'	'd8861905a878ac2eca57d1b6117cccde'	200