

THE
DEVELOPER'S
CONFERENCE

TDC INNOVATION

CHALLENGES FOR CREATING THE DIGITAL FUTURE

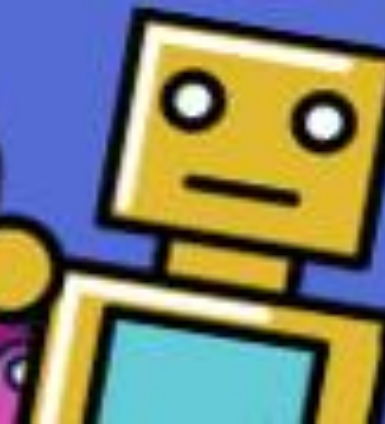
03 de junho - Florianópolis

AUTOMATIZANDO

APIS

COM

ROBOT FRAMEWORK



Quem sou eu?

Me chamo Danielle Cavalcanti, tenho 37 anos, QA por vocação e paixão, carioca, mãe da Malu Helena 🐼, Shuri Lee 🐼, Lolla Maria 🐼 e Meg Louise 🐼.

Apaixonada pela cultura ágil em especial Design Thinking, movida por desafios e em constante formação e evolução.



Dani



Perfil Profissional



Sou graduada em Análise e Desenvolvimento de Sistemas, certificada pela ISTQB (CTFL) e atualmente pós-graduanda em Engenharia de Software com Ênfase em Qualidade e Testes de Software.

Trabalho desde 2018 como QA, onde comecei como estagiária de Testes e de Apoio à Gerência de Projeto e após 5 meses como estagiária, fui contratada como Jr e, desde então, sigo crescendo dentro da área de QA.

No momento, sou QA Lead da WS Rocket QA na Sensedia onde, além de atuar na Torre de Open Insurance, promovo mentoria e capacitação de novos QAs.



Por quê usar o Robot Framework?

REST-Assured

```
private val ENDPOINT: String = "/turismo"
@LocalServerPort
internal var port: Int = 0 -> porta aleatória da aplicação
@Test
fun `Dado que a operação seja um sucesso é retornado uma lista com todos os guias turísticos`() {
    val before = repository.findall()
    val after = given() -> serão usadas posteriormente

    .port(port)
    .log().all()
    .`when` ()
    .get("$ENDPOINT/guias")
    .then()
    .log().all()
    .statusCode(OK.value()) -> Status Code 200
}
```

Pytest

```
from wsgiref import headers
import requests
def test_status():
    headers = {
        'Accept': '*/*',
        'User-Agent': 'request',
    }
    url = "http://exemploapi.com"
    resp = requests.get(url, headers=headers)
    resp_dict = resp.json

    status = resp_dict['status']

    assert status == 'sucess'
```

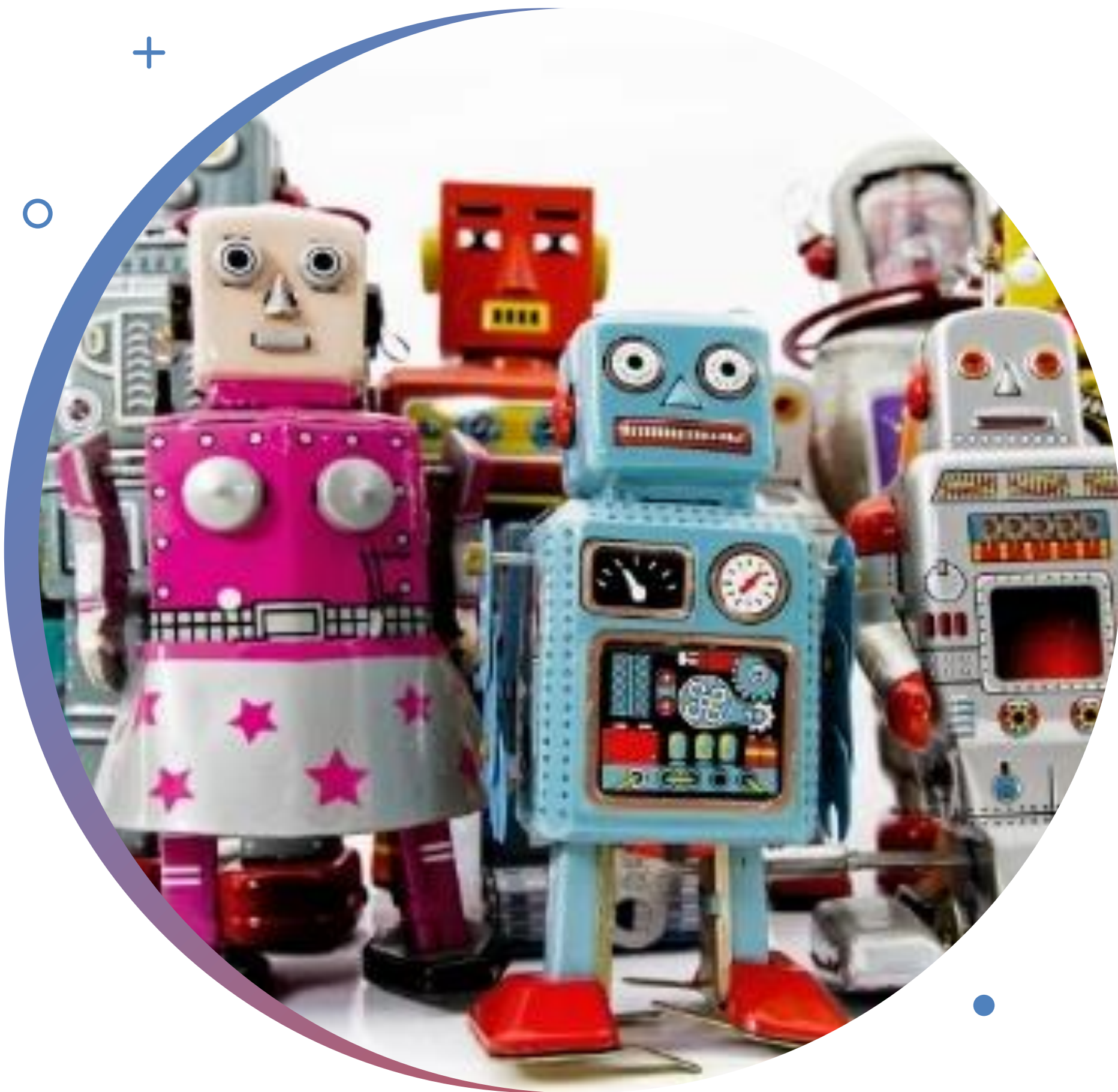
Robot Framework

```
*** Keywords ***
Conectar API
    Create Session    rocketqa-api    http://localhost:1302
```




Robot Framework

- ✓ O Robot Framework é uma estrutura para automação de testes de aceitação ATDD;
- ✓ Framework genérico que permite automação de qualquer tipo de sistema (web, API, mobile, desktop, etc);
- ✓ Baseado em keyword-driven;
- ✓ Estendido através de libraries implementadas em Python ou Java;
- ✓ Compatível com o Python 2 e 3;
- ✓ Compatível com Integração Contínua e Cross-browser;
- ✓ Multiplataforma (Windows, Linux, MAC e em containers Docker).



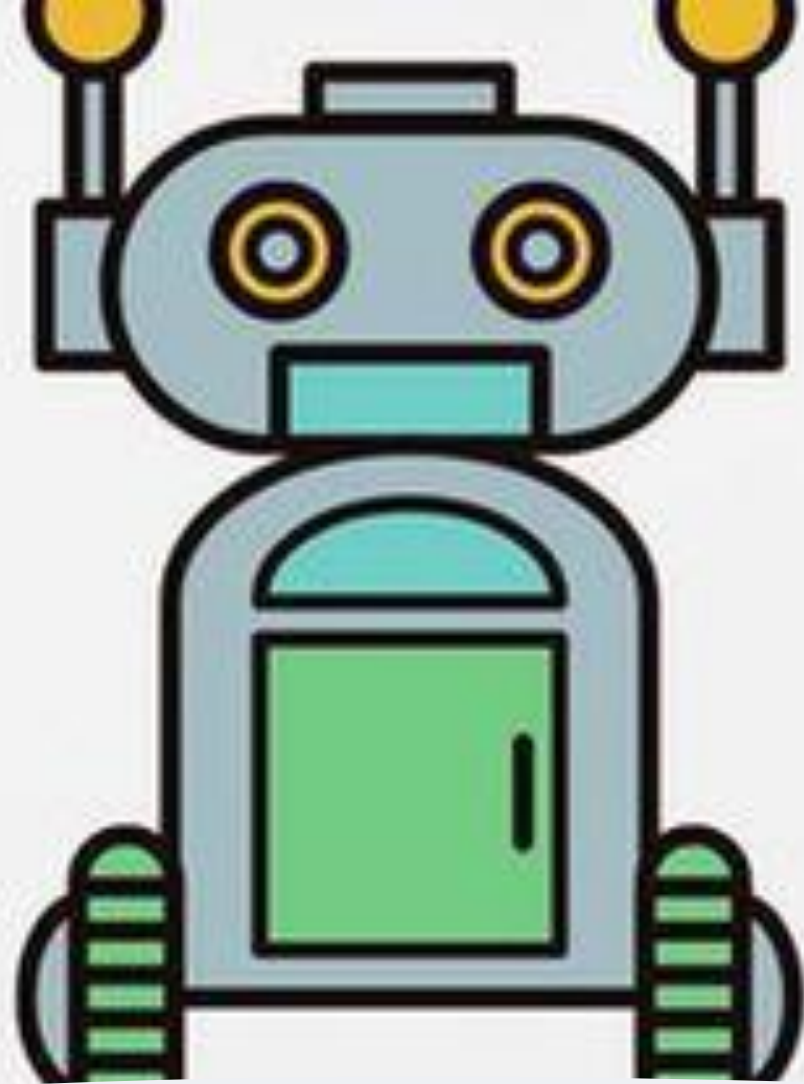
Keyword

- ✓ Ferramenta de automação que oferece um conjunto de palavras-chaves;
- ✓ Linguagem de alto nível (linguagem nativa) que representa a ação do usuário);
- ✓ Podem ser escritas de forma Procedural ou em BDD (Gerkin), dispensando frameworks auxiliares (Cucumber, SpecFlow, Behave, etc).

Keywords usadas no projeto

- Create Dictionary
- Convert To String
- Set Variable
- Evaluate
- Set To Dictionary
- Get File

- Create Session
- POST On Session
- GET On Session
- PUT On Session
- DELETE On Session
- Status Should Be



```
***Settings***
```

```
Library      OperatingSystem
```

```
***Keywords***
```

```
Get Json
```

```
    [Arguments]    ${file_name}
```

```
    ${json_file}
```

```
    ${json_dict}=
```

```
        Get File  
        Evaluate
```

```
    ${EXECDIR}/resources/fixtures/${file_name}
```

```
    json.loads(${json_file})    json
```

```
    [Return]        ${json_dict}
```

Keywords usadas no projeto



Library

- ✓ Conjunto de Keywords para determinada funcionalidade;
- ✓ São bem documentadas;
- ✓ Existem várias bibliotecas divididas em: Standard (inclusas na estrutura) e External (desenvolvidas separadamente que podem ser instaladas com base nas necessidades do teste).



Libraries usadas no projeto

- BuiltIn
- Collections
- Operating System
- **RequestsLibrary**

Libraries usadas no projeto

* Settings *

Library	Collections
Library	RequestsLibrary

Suite Setup Create Session jsonplaceholder <https://jsonplaceholder.typicode.com>

* Test Cases *

Get Request Test

Create Session google <http://www.google.com>

\${resp_google}= GET On Session google / expected_status=200

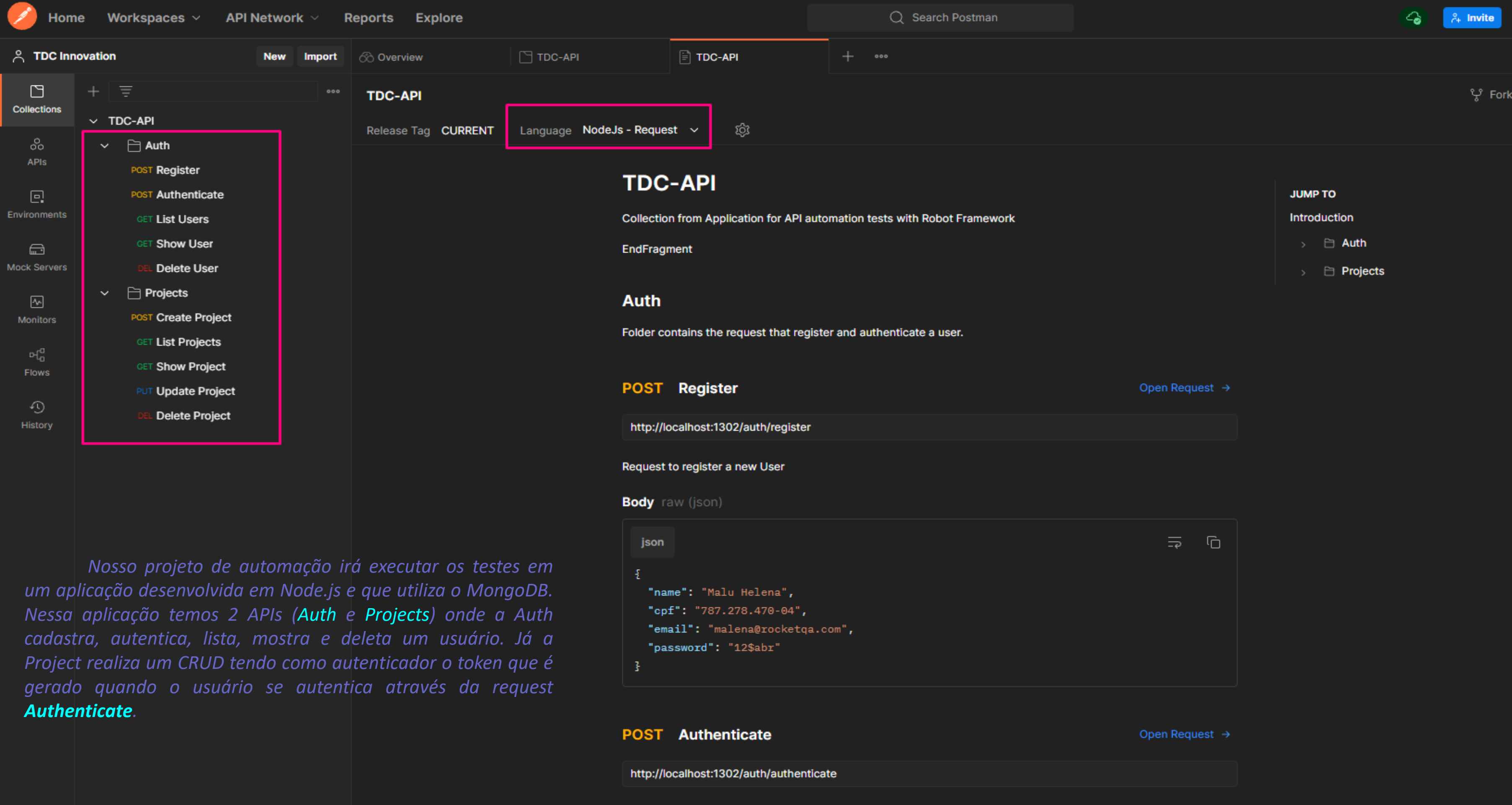
\${resp_json}= GET On Session jsonplaceholder /posts/1

Should Be Equal As Strings **\${resp_google.reason}** OK

Dictionary Should Contain Value **\${resp_json.json()}** sunt aut facere repellat provident

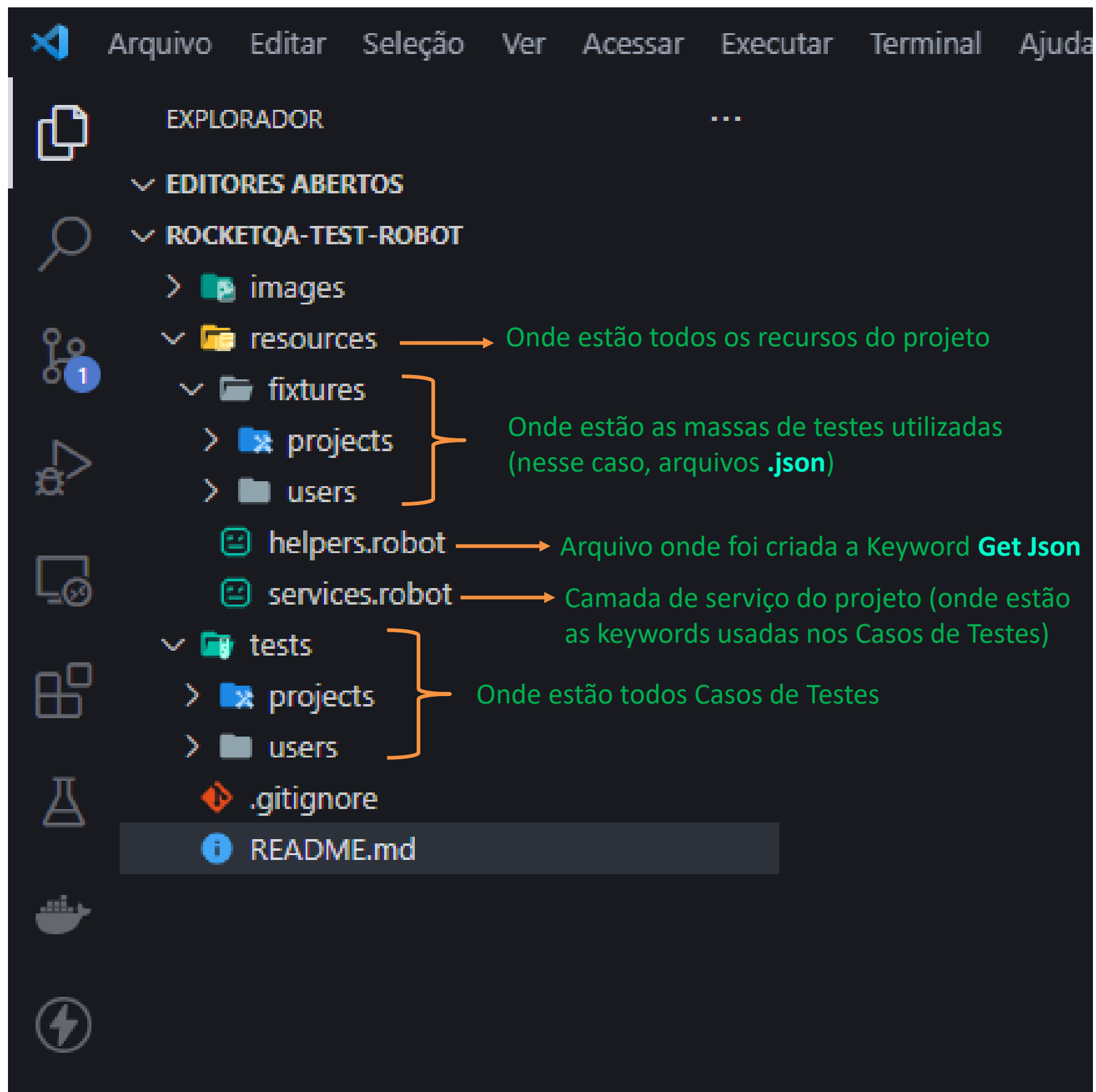
LET'S CODE





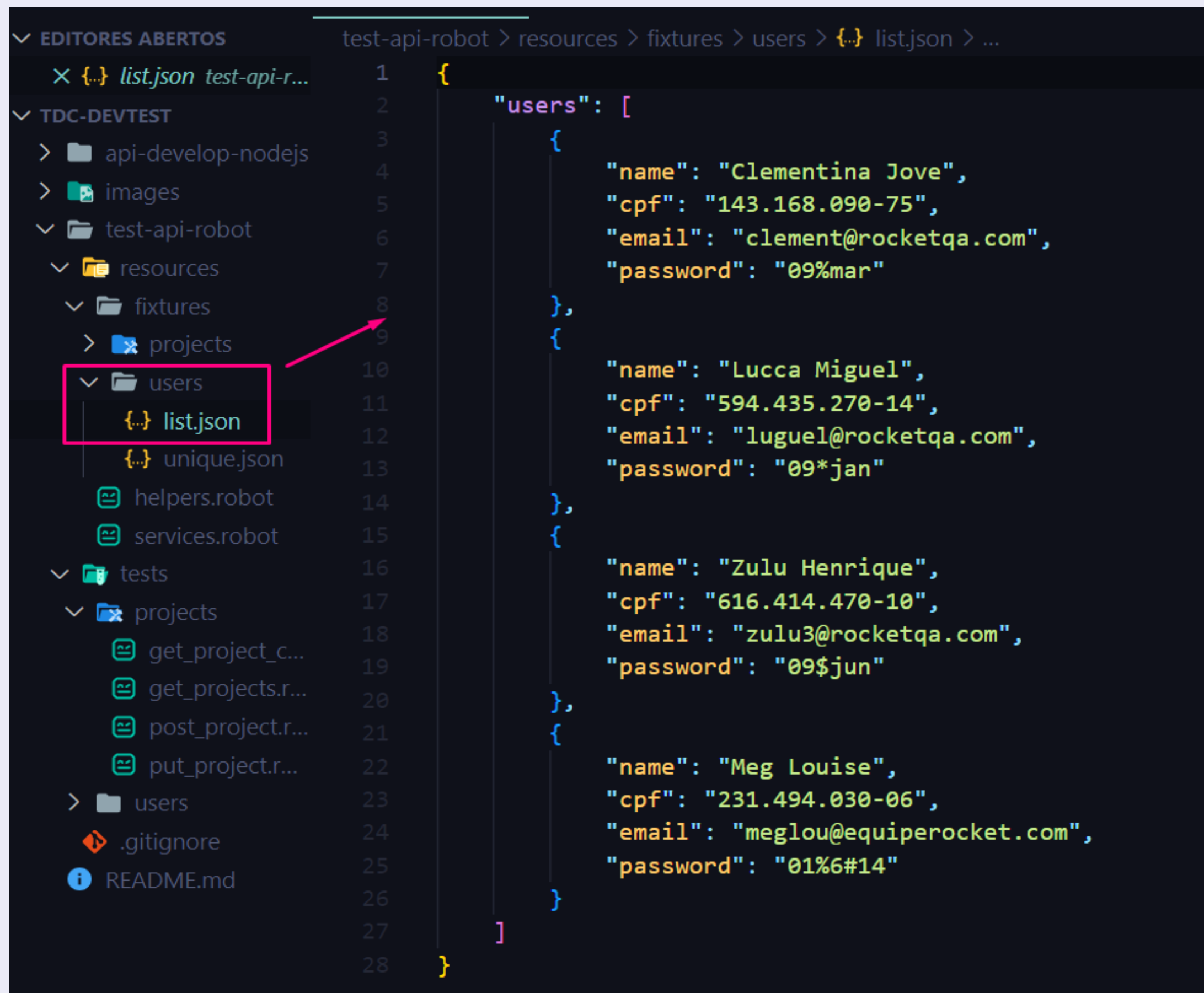
Nosso projeto de automação irá executar os testes em um aplicação desenvolvida em Node.js e que utiliza o MongoDB. Nessa aplicação temos 2 APIs (**Auth** e **Projects**) onde a Auth cadastra, autentica, lista, mostra e deleta um usuário. Já a Project realiza um CRUD tendo como autenticador o token que é gerado quando o usuário se autentica através da request **Authenticate**.

Collection do Projeto



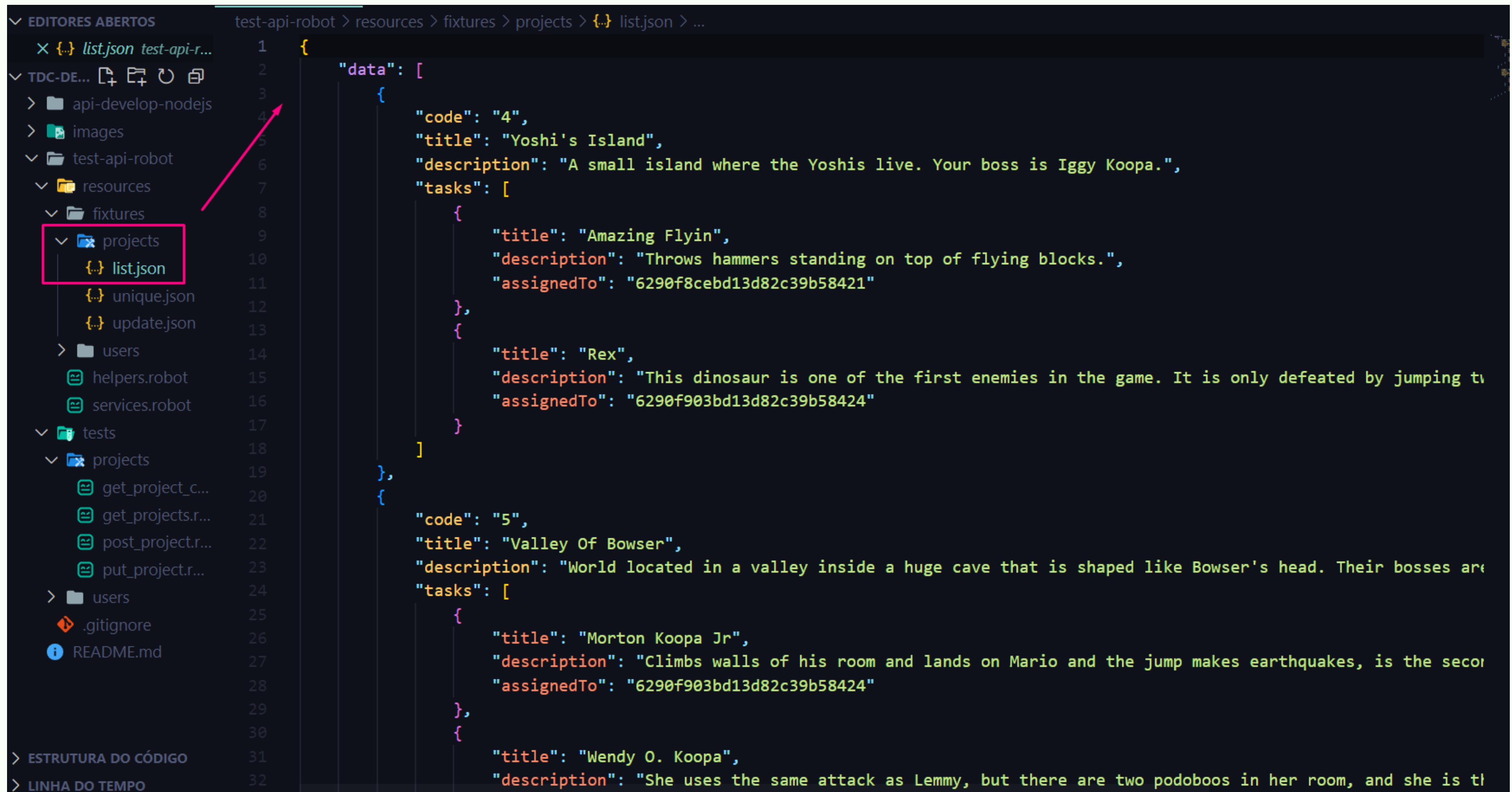
Estrutura do Projeto

Fixtures (users/list.json)



```
test-api-robot > resources > fixtures > users > {..} list.json > ...  
1 {  
2  
3   "users": [  
4     {  
5       "name": "Clementina Jove",  
6       "cpf": "143.168.090-75",  
7       "email": "clement@rocketqa.com",  
8       "password": "09%mar"  
9     },  
10    {  
11      "name": "Lucca Miguel",  
12      "cpf": "594.435.270-14",  
13      "email": "luguel@rocketqa.com",  
14      "password": "09*jan"  
15    },  
16    {  
17      "name": "Zulu Henrique",  
18      "cpf": "616.414.470-10",  
19      "email": "zulu3@rocketqa.com",  
20      "password": "09$jun"  
21    },  
22    {  
23      "name": "Meg Louise",  
24      "cpf": "231.494.030-06",  
25      "email": "meglou@equiperocket.com",  
26      "password": "01%6#14"  
27    }  
28  ]  
29 }
```

Fixtures (projects/list.json)



```
1 {
2   "data": [
3     {
4       "code": "4",
5       "title": "Yoshi's Island",
6       "description": "A small island where the Yoshis live. Your boss is Iggy Koopa.",
7       "tasks": [
8         {
9           "title": "Amazing Flyin",
10          "description": "Throws hammers standing on top of flying blocks.",
11          "assignedTo": "6290f8cebd13d82c39b58421"
12        },
13        {
14          "title": "Rex",
15          "description": "This dinosaur is one of the first enemies in the game. It is only defeated by jumping to the top of the screen.",
16          "assignedTo": "6290f903bd13d82c39b58424"
17        }
18      ]
19    },
20    {
21      "code": "5",
22      "title": "Valley Of Bowser",
23      "description": "World located in a valley inside a huge cave that is shaped like Bowser's head. Their bosses are Morton Koopa Jr, Wendy O. Koopa, and Bowser Jr.",
24      "tasks": [
25        {
26          "title": "Morton Koopa Jr",
27          "description": "Climbs walls of his room and lands on Mario and the jump makes earthquakes, is the second boss of the world.",
28          "assignedTo": "6290f903bd13d82c39b58424"
29        },
30        {
31          "title": "Wendy O. Koopa",
32          "description": "She uses the same attack as Lemmy, but there are two podoboos in her room, and she is the third boss of the world."
33        }
34      ]
35    }
36  ]
37 }
```

helpers.robot



The screenshot shows a code editor with a sidebar on the left displaying a project structure. The main editor area shows the content of the `helpers.robot` file. The sidebar includes sections for 'EDITORES ABERTOS' (Open Editors) and 'TDC-DEVTEST' (a folder containing subfolders like `api-develop-nodejs`, `images`, `test-api-rob...`, `resources`, `fixtures`, `helpers.r...` (highlighted with a red box), `services.robot`, `tests`, `projects`, `users`, `.gitignore`, and `README.md`). The main editor shows the following code:

```
test-api-robot > resources > helpers.robot > ...  
1  ***Settings***  
2  Library      OperatingSystem  
3  
4  
5  ***Keywords***  
6  Get Json  
7      [Arguments]    ${file_name}  
8  
9      ${json_file}    Get File      ${EXECDIR}/resources/fixtures/${file_name}  
10     ${json_dict}=    Evaluate      json.loads(${json_file})    json  
11  
12     [Return]         ${json_dict}  
13  
14  
15  
16  
17  
18  
19  
20  
21
```


services.robot

services.robot M X

test-api-robot > resources > services.robot > Put Project

```
1  ***Settings***
2  Documentation      Camada de serviços do projeto de automação
3
4  Library            RequestsLibrary
5  Library            Collections
6  Resource           helpers.robot
7
8  ***Variables***
9  ${base_url_api}    http://localhost:1302
10
11 ***Keywords***
12 ## Helpers
13 Conectar API
14 |   Create Session   rocketqa-api    ${base_url_api}
15
```

services.robot

test-api-robot > resources > services.robot > Post Authenticate

EDITORES ABERTOS

- services.ro... M
- get_user.ro... M
- register.robot tes...

TDC-DEVTEST

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
- projects
- users
 - auth.robot
 - get_use... M
 - register.robot
- .gitignore
- README.md

```
4 Library RequestsLibrary
5 Library Collections
6 Resource helpers.robot
7
8 ***Variables***
9 ${base_url_api} http://localhost:1302
10
11 ***Keywords***
12 ## Helpers
13 Conectar API
14     Create Session    rocketqa-api    ${base_url_api}
15
16 > Get Session Token ...
17
18 ## POST /register
19 Post Register
20     [Arguments]    ${payload}
21
22     Conectar API
23
24     &{headers}=    Create Dictionary    Content-type=application/json
25
26     ${resp}=    POST On Session    rocketqa-api    /auth/register
27     ...    json=${payload}    headers=${headers}
28
29     [Return]    ${resp}
```

services.robot

```
test-api-robot > resources > services.robot > ...
11 keyword us
12 ## Helpers
13 Conectar API
14     Create Session          rocketqa-api      ${base_url_api}
15
16 > Get Session Token ...
24 ## POST /register
25 > Post Register ...
38 ## POST /authenticate
39 Post Authenticate
40     [Arguments]      ${email}      ${password}
41
42     Conectar API
43
44     &{headers}=        Create Dictionary      Content-type=application/json
45     &{payload}=        Create Dictionary      email=${email}      password=${password}
46
47     ${resp}=          POST On Session          rocketqa-api      /auth/authenticate
48     ...               json=${payload}         headers=${headers}
49
50     [Return]          ${resp}
```


services.robot

services.ro... M

TDC-DEVTEST

api-develop-nodejs

images

test-api-rob...

resources

fixtures

helpers.r... M

services.r... M

tests

projects

users

auth.robot

get_use... M

register.robot

.gitignore

README.md

24 ## POST /register

25 > Post Register ...

38 ## POST /authenticate

39 > Post Authenticate ...

53 ##GET/ user

54 Get Users

55 Conectar API

56

57 &{headers}= Create Dictionary Content-type=application/json

58

59 \${resp}= GET On Session rocketqa-api /auth/users headers=\${headers}

60

61 [Return] \${resp}

62

63 Get User CPF

64 [Arguments] \${cpf}

65

66 Conectar API

67

68 &{headers}= Create Dictionary Content-type=application/json

69

70 \${resp}= GET On Session rocketqa-api /auth/users/\${cpf} headers=\${headers}

71

72 [Return] \${resp}

73

services.robot

EDITORES ABERTOS

services.ro... M

TDC-DEVTEST

api-develop-nodejs

images

test-api-rob...

resources

fixtures

helpers.r... M

services.r... M

tests

projects

users

auth.robot

get_use... M

register.robot

.gitignore

README.md

test-api-robot > resources > services.robot > Delete User

54 Get Users

55 Conectar API

56

57 &{headers}= Create Dictionary Content-type=application/json

58

59 \${resp}= GET On Session rocketqa-api /auth/users headers=\${headers}

60

61 [Return] \${resp}

62

63 > Get User CPF ...

74 ##DELETE /user

75 Delete User

76 [Arguments] \${cpf}

77

78 Conectar API

79

80 &{headers}= Create Dictionary Content-Type=application/json

81

82 DELETE On Session rocketqa-api /auth/users/\${cpf} headers=\${headers}

83

services.robot

EDITORES ABERTOS

services.ro... M

TDC-DEVTEST

api-develop-nodejs

images

test-api-rob... ●

resources ●

fixtures

helpers.r... M

services.r... M

tests ●

projects ●

get_proj... M

post_project.r... M

put_project.r...

users ●

.gitignore

README.md

test-api-robot > resources > services.robot > Post Create Project

Create Session

rocketqa-api

`${base_url_api}`

Get Session Token

`${resp}=` Post Authenticate `malena@rocketqa.com` 12\$abr

`${token}` Convert To String Bearer `${resp.json()['token']}`

[Return] `${token}`

Projects

POST /projects

Post Create Project

[Arguments] `${payload}`

Conectar API

`${token}=` Get Session Token

`&{headers}=` Create Dictionary `Content-type=application/json` `Authorization=${token}`

`${resp}=` POST On Session `rocketqa-api` `/projects`

`...` `json=${payload}` `headers=${headers}`

[Return] `${resp}`

services.robot

The image shows a VS Code interface with the following components:

- File Explorer (Left):** Displays the project structure. The 'test-api-robot' folder is expanded, showing subfolders like 'fixtures', 'helpers.r...', 'services.r...', 'tests', and 'projects'. The 'get_proj...' file in the 'projects' folder is highlighted with a red box.
- Editor (Right):** Displays the 'services.robot' file. The file contains two sections: 'Get Projects' and 'Get Unique Project'. The 'Get Projects' section is highlighted with a red box, and the 'Get Unique Project' section is highlighted with a blue box. The 'get_proj...' file in the file explorer is also highlighted with a red box.

The 'Get Projects' section contains the following code:

```
## GET /projects
Get Projects
  Conectar API

  ${token}=          Get Session Token
  &{headers}=        Create Dictionary          Content-type=application/json      Authorization=${token}

  ${resp}=           GET On Session            rocketqa-api          /projects      headers=${headers}

  [Return]           ${resp}
```

The 'Get Unique Project' section contains the following code:

```
Get Unique Project
[Arguments]         ${code}

  Conectar API

  ${token}=          Get Session Token
  &{headers}=        Create Dictionary          Content-type=application/json      Authorization=${token}

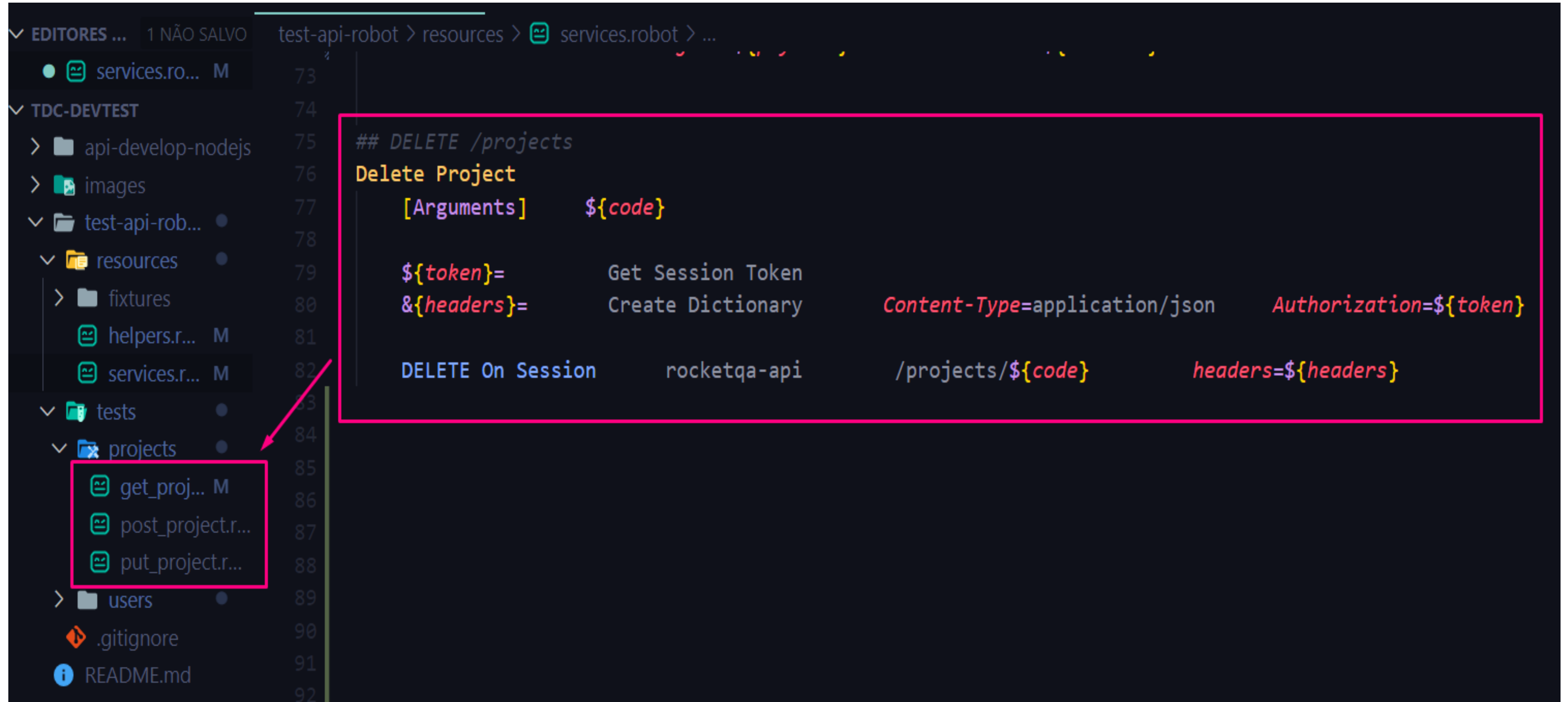
  ${resp}=           GET On Session            rocketqa-api          /projects/${code}      headers=${headers}
```

services.robot

```
test-api-robot > resources > services.robot > Delete Project

25  ## POST /projects
26  > Post Create Project ...
40  ## GET /projects
41  > Get Projects ...
51  > Get Unique Project ...
62  ## PUT /projects
63  Put Project
64  [Arguments]          ${payload}          ${code}
65
66  Conectar API
67
68  ${token}=             Get Session Token
69  &{headers}=           Create Dictionary      Content-type=application/json      Authorization=${token}
70
71  ${resp}=              PUT On Session        rocketqa-api          /projects/${code}
72  ...                   json=${payload}      headers=${headers}
73
74
```

services.robot



```
## DELETE /projects
Delete Project
    [Arguments]    ${code}

    ${token}=      Get Session Token
    &{headers}=    Create Dictionary    Content-Type=application/json    Authorization=${token}

    DELETE On Session    rocketqa-api    /projects/${code}    headers=${headers}
```


register.robot

test-api-robot > tests > users > register.robot > ...

```
1  *** Settings ***
2  Documentation      Caso de Teste da Collection: authControler - Cadastra um novo usuário
3  Resource            ../../resources/services.robot
4
5
6  *** Test Cases ***
7  Register User
8      ${payload}=      Get Json      users/unique.json
9
10     Delete User      ${payload['cpf']}
11
12     ${resp}=          Post Register  ${payload}
13
14     Status Should Be 201            ${resp}
```

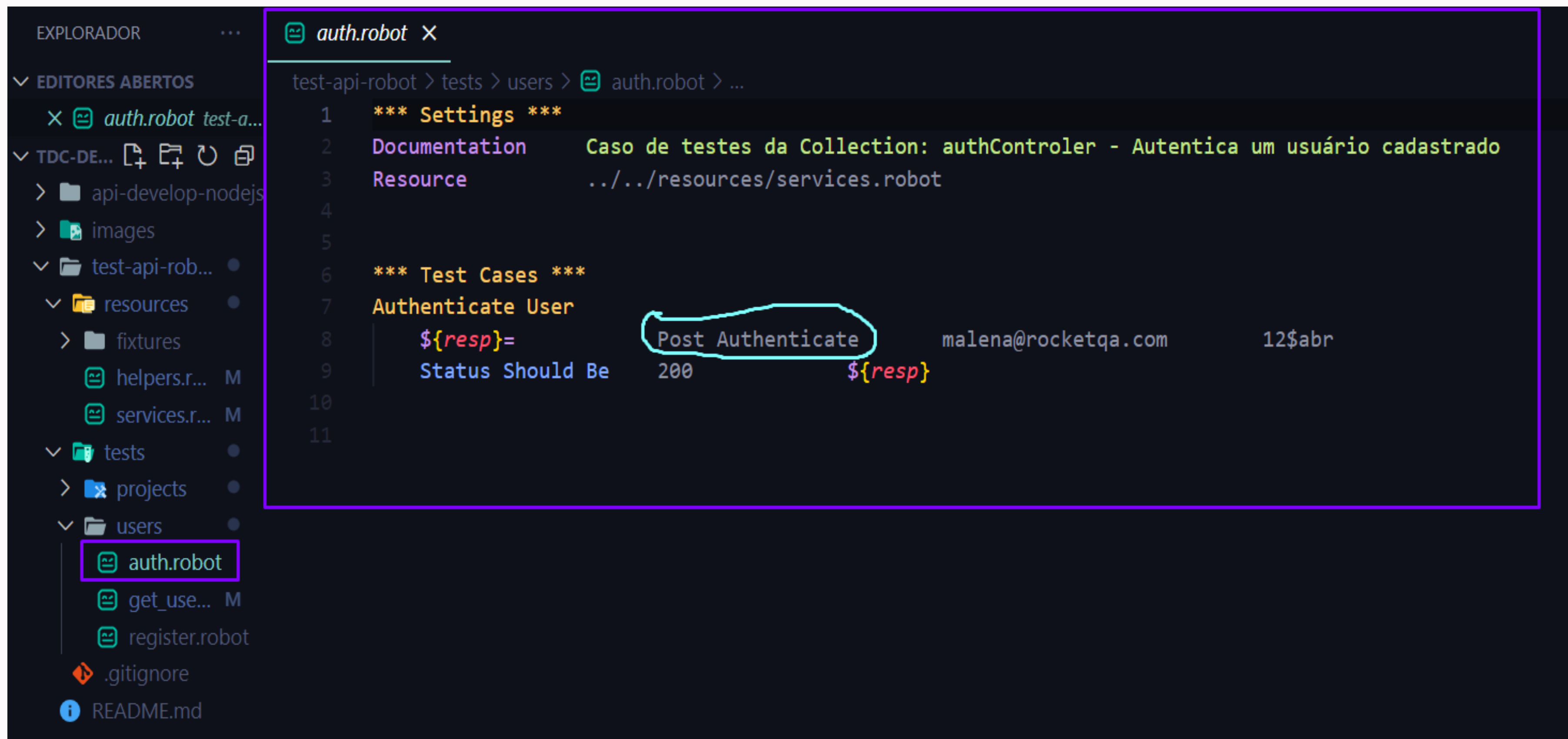
EDITORRES ABERTOS

- register.robot tes...

TDC-DEVTEST

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
 - projects
 - users
 - auth.robot
 - get_use... M
 - register.robot
- .gitignore
- README.md

auth.robot



The screenshot shows the Visual Studio Code interface with the `auth.robot` file open in the editor. The file is located in the `test-api-robot > tests > users` directory. The editor displays the following content:

```
1  *** Settings ***
2  Documentation      Caso de testes da Collection: authController - Autentica um usuário cadastrado
3  Resource            ../../resources/services.robot
4
5
6  *** Test Cases ***
7  Authenticate User
8      ${resp}=        Post Authenticate        malena@rocketqa.com        12$abr
9      Status Should Be 200                    ${resp}
10
11
```

The `Post Authenticate` step in the test case is highlighted with a red circle. The left sidebar shows the Explorer view with the `auth.robot` file selected under the `users` folder.

get_user.robot

EXPLORADOR

EDITORES ABERTOS

- get_user.ro... M

TDC-DEVTEST

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
 - projects
 - users
 - auth.robot
 - get_use... M
 - register.robot
- .gitignore
- README.md

test-api-robot > tests > users > get_user.robot > ...

```
1  *** Settings ***
2  Documentation      Caso de Teste da Collection: authControler - Lista todos os usuários cadastrados
3  Resource            ../../resources/services.robot
4
5  *** Test Cases ***
6
7  Get Users List
8
9      ${list}=          Get Json          users/list.json
10
11     FOR                ${item}          IN          @${list['users']}
12         Post Register  ${item}
13     END
14
15     ${resp}=           Get Users
16     Status Should Be  200              ${resp}
17
18
19  Get Show User
20
21     ${origin}=         Get Json          users/unique.json
22
23     Delete User        ${origin['cpf']}
24     ${resp}=           Post Register    ${origin}
25     ${user_cpf}=       Set Variable    ${resp.json()['user']['cpf']}
26
27
28     ${resp}=           Get User CPF     ${user_cpf}
29
30     Status Should Be  200              ${resp}
31
```


post_project.robot

EXPLORADOR

EDITORES ABERTOS

- post_project.robo...

TDC-DE...

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
 - projects
 - get_proj... M
 - post_project.r...
 - put_project.r...
- users
- .gitignore
- README.md

test-api-robot > tests > projects > post_project.robot > ...

```
1  *** Settings ***
2  Documentation      Caso de Teste da Collection: projectControler - Cadastra um novo projeto
3  Resource            ../../resources/services.robot
4
5  *** Test Cases ***
6  Create Project
7      ${payload}=      Get Json      projects/unique.json
8
9      Delete Project    ${payload['code']}
10
11     ${resp}=          Post Create Project    ${payload}
12
13     Status Should Be  201      ${resp}
14
15
```

get_projects.robot

EXPLORADOR

test-api-robot > tests > projects > get_projects.robot > ...

EDITORES ABERTOS

- get_projec... M

TDC-DEVTEST

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
 - projects
 - get_proj... M
 - post_project.r...
 - put_project.r...
- users
- .gitignore
- README.md

ESTRUTURA DO CÓDIGO

```
1  *** Settings ***
2  Documentation      Caso de Teste da Collection: projectControler - Lista todos os projetos cadastrados
3  Resource           ../../resources/services.robot
4
5  *** Test Cases ***
6  Get Project List
7      ${list}=          Get Json          projects/list.json
8
9      FOR              ${item}          IN          @${list['data']}
10         Post Create Project  ${item}
11     END
12
13
14     ${resp}=          Get Projects
15     Status Should Be  200          ${resp}
16
17
18  Get Show Project
19
20     ${origin}=          Get Json          projects/unique.json
21
22     Delete Project      ${origin['code']}
23     ${resp}=          Post Create Project  ${origin}
24
25     ${projectcode}=          Convert To String          ${resp.json()['project']['code']}
26
27
28     ${resp}=          Get Unique Project  ${projectcode}
29
30     Status Should Be  200          ${resp}
31
```

put_project.robot

EXPLORADOR ... put_project.robot X

EDITORES ABERTOS

- put_project.robot...

TDC-DEVTEST

- api-develop-nodejs
- images
- test-api-rob...
- resources
 - fixtures
 - helpers.r... M
 - services.r... M
- tests
 - projects
 - get_proj... M
 - post project.r...
 - put_project.r...
- users
- .gitignore
- README.md

test-api-robot > tests > projects > put_project.robot > ...

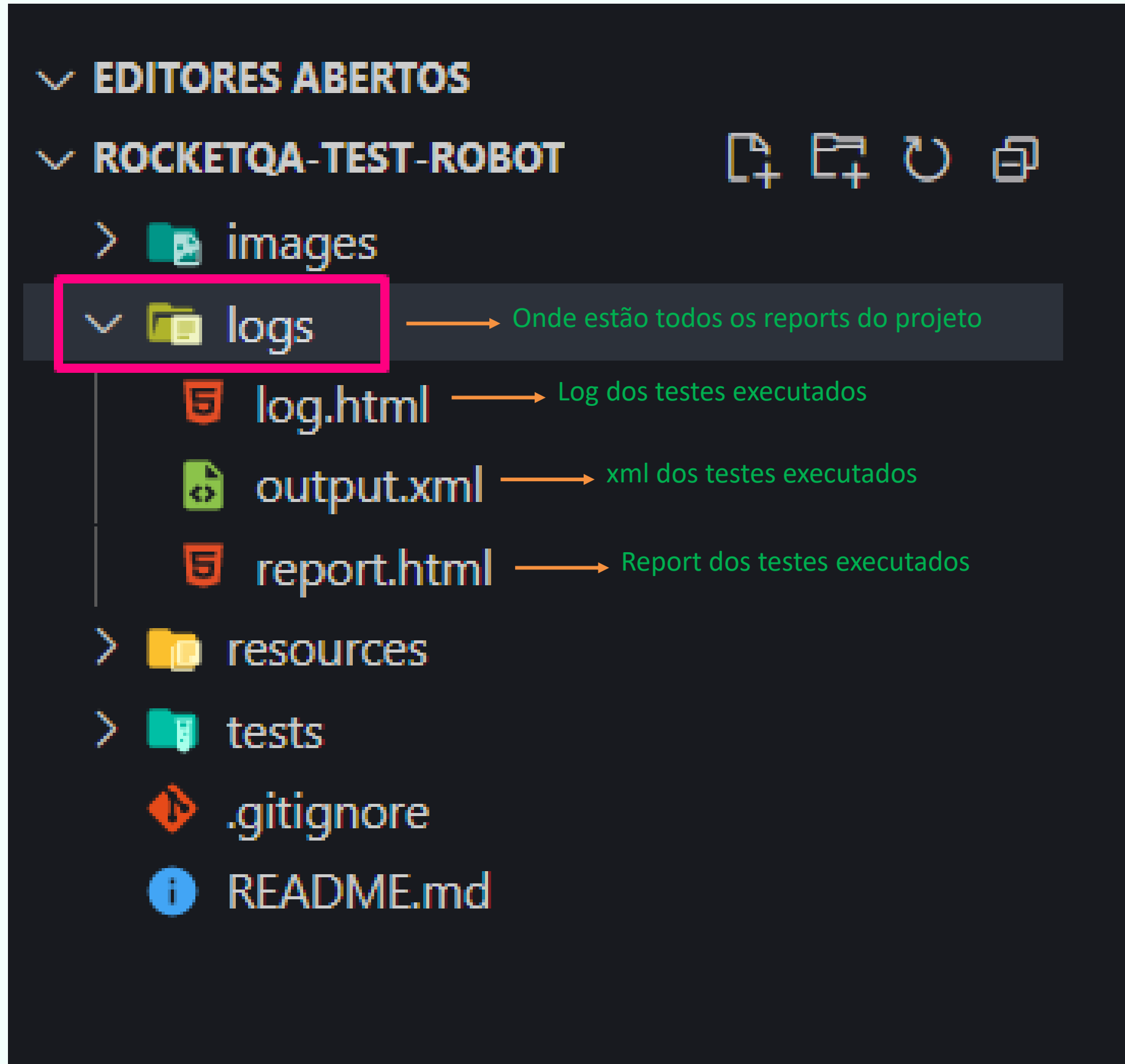
```
1 *** Settings ***
2 Documentation      Caso de Teste da Collection: projectControler - Atualiza um projeto cadastrado pelo seu code
3 Resource           ../../resources/services.robot
4
5 *** Test Cases ***
6
7 Update a Project
8
9     ${payload}=      Get Json           projects/unique.json
10    ${update}=       Get Json           projects/update.json
11
12    Delete Project    ${payload['code']}
13    ${resp}=        Post Create Project ${payload}
14
15    ${projectcode}=  Convert To String  ${resp.json()['project']['code']}
16
17    Set To Dictionary    ${update}    [tasks]    ${payload}
18
19    ${resp}=            Put Project    ${update}    ${projectcode}
20
21    Status Should Be    200            ${resp}
22
```

Executando os testes

```
λ Cmder

C:\rocketqa-test-robot (master -> origin)
λ robot -d ./logs tests\users\register.robot → Comando para executar o teste
=====
Register :: Caso de Teste da Collection: authControler - Cadastra um novo u...
=====
Register User | PASS |
=====
Register :: Caso de Teste da Collection: authControler - Cadastra ... | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\rocketqa-test-robot\logs\output.xml
Log: C:\rocketqa-test-robot\logs\log.html
Report: C:\rocketqa-test-robot\logs\report.html } Geração dos Reports que serão armazenados na pasta logs
```


Logs



Get User Cpf Log

Generated
20220527 18:56:05 UTC-03:00
10 seconds ago

Test Statistics

Test Execution Log

- SUITE Get User Cpf		00:00:00.353
Full Name:	Get User Cpf	
Documentation:	Documentação dos casos de testes da Collection: authControler	
Source:	C:\rocketqa-test-robot\tests\users\get_user_cpf.robot	
Start / End / Elapsed:	20220527 18:56:05.327 / 20220527 18:56:05.680 / 00:00:00.353	
Status:	1 test total, 1 passed, 0 failed, 0 skipped	
- TEST Get Show User		
Full Name:	Get User Cpf.Get Show User	
Start / End / Elapsed:	20220527 18:56:05.500 / 20220527 18:56:05.678 / 00:00:00.178	
Status:	PASS	
+ KEYWORD	\$_{origin} = helpers.Get Json users/unique.json	00:00:00.005
+ KEYWORD	services.Delete User \$_{origin}['cpf']	00:00:00.059
+ KEYWORD	\$_{resp} = services.Post Register \$_{origin}	00:00:00.098
+ KEYWORD	\$_{user_cpf} = BuiltIn.Set Variable \$_{resp.json()['user']['cpf']	00:00:00.001
+ KEYWORD	\$_{resp} = services.Get User CPF \$_{user_cpf}	00:00:00.010
+ KEYWORD	RequestsLibrary.Status Should Be 200, \$_{resp}	00:00:00.000

log.html

- TEST Get Show User		00:00:00.178
Full Name: Get User Cpf.Get Show User		
Start / End / Elapsed: 20220527 18:56:05.500 / 20220527 18:56:05.678 / 00:00:00.178		
Status: PASS		
- KEYWORD \${origin} = helpers.Get Json users/unique.json		00:00:00.005
Start / End / Elapsed: 20220527 18:56:05.501 / 20220527 18:56:05.506 / 00:00:00.005		
- KEYWORD \${json_file} = OperatingSystem.Get File \${EXECDIR}/resources/fixtures/\${file_name}		00:00:00.001
Documentation: Returns the contents of a specified file.		
Start / End / Elapsed: 20220527 18:56:05.501 / 20220527 18:56:05.502 / 00:00:00.001		
18:56:05.501	INFO	Getting file 'C:\rocketqa-test-robot\resources\fixtures\users\unique.json'.
18:56:05.502	INFO	<pre>\${json_file} = { "name": "Luz de Mila", "cpf": "502.858.430-36", "email": "lumila@equiperocket.com", "password": "14&410" }</pre>
- KEYWORD \${json_dict} = BuiltIn.Evaluate json.loads(\${json_file}), json		00:00:00.003
Documentation: Evaluates the given expression in Python and returns the result.		
Start / End / Elapsed: 20220527 18:56:05.502 / 20220527 18:56:05.505 / 00:00:00.003		
18:56:05.505	INFO	<pre>\${json_dict} = {'name': 'Luz de Mila', 'cpf': '502.858.430-36', 'email': 'lumila@equiperocket.com', 'password': '14&410'}</pre>
18:56:05.506	INFO	<pre>\${origin} = {'name': 'Luz de Mila', 'cpf': '502.858.430-36', 'email': 'lumila@equiperocket.com', 'password': '14&410'}</pre>
- KEYWORD services.Delete User \${origin['cpf']}		00:00:00.059
Start / End / Elapsed: 20220527 18:56:05.506 / 20220527 18:56:05.565 / 00:00:00.059		
- KEYWORD RequestsLibrary.Create Session rocketqa-api, \${base_url_api}		00:00:00.000
Documentation: Create Session: create a HTTP session to a server		
Start / End / Elapsed: 20220527 18:56:05.507 / 20220527 18:56:05.507 / 00:00:00.000		
18:56:05.507	INFO	Creating Session using : alias=rocketqa-api, url=http://localhost:1302, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=False, debug=0
- KEYWORD &{headers} = BuiltIn.Create Dictionary Content-Type=application/json		00:00:00.000
Documentation: Creates and returns a dictionary based on the given items.		
Start / End / Elapsed: 20220527 18:56:05.508 / 20220527 18:56:05.508 / 00:00:00.000		
18:56:05.508	INFO	<pre>&{headers} = { Content-Type=application/json }</pre>
- KEYWORD RequestsLibrary.DELETE On Session rocketqa-api, /auth/users/\${cpf}, headers=\${headers}		00:00:00.057
Documentation: Sends a DELETE request on a previously created HTTP Session.		
Start / End / Elapsed: 20220527 18:56:05.508 / 20220527 18:56:05.565 / 00:00:00.057		
18:56:05.564	INFO	<pre>DELETE Request : url=http://localhost:1302/auth/users/502.858.430-36 path_url=/auth/users/502.858.430-36 headers={'User-Agent': 'python-requests/2.27.1', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-Type': 'application/json', 'Content-Length': '0'} body=None</pre>
18:56:05.564	INFO	<pre>DELETE Response : url=http://localhost:1302/auth/users/502.858.430-36 status=204, reason=No Content headers={'X-Powered-By': 'Express', 'Access-Control-Allow-Origin': '*', 'Access-Control-Allow-Methods': 'GET,PUT,POST,DELETE', 'ETag': 'W/"25-iQsNwei0jlji1DEVaH1tWq5WmQ"', 'Date': 'Fri, 27 May 2022 21:56:05 GMT', 'Connection': 'keep-alive', 'Keep-Alive': 'timeout=5'} body=None</pre>

log.html

<div>+<div>KEYWORD</div></div>	<code>\${origin} = helpers.Get Json users/unique.json</code>	00:00:00.005
<div>+<div>KEYWORD</div></div>	<code>services.Delete User \${origin['cpf']}</code>	00:00:00.059
<div>-<div>KEYWORD</div></div>	<code>\${resp} = services.Post Register \${origin}</code>	00:00:00.098
Start / End / Elapsed: 20220527 18:56:05.566 / 20220527 18:56:05.664 / 00:00:00.098		
<div>-<div>KEYWORD</div></div>	<code>RequestsLibrary.Create Session rocketqa-api, \${base_url_api}</code>	00:00:00.000
Documentation: Create Session: create a HTTP session to a server		
Start / End / Elapsed: 20220527 18:56:05.566 / 20220527 18:56:05.566 / 00:00:00.000		
18:56:05.566 INFO Creating Session using : alias=rocketqa-api, url=http://localhost:1302, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=False, debug=0		
<div>-<div>KEYWORD</div></div>	<code>&{headers} = BuiltIn.Create Dictionary Content-type=application/json</code>	00:00:00.001
Documentation: Creates and returns a dictionary based on the given items.		
Start / End / Elapsed: 20220527 18:56:05.566 / 20220527 18:56:05.567 / 00:00:00.001		
18:56:05.567 INFO &{headers} = { Content-type=application/json }		
<div>-<div>KEYWORD</div></div>	<code>\${resp} = RequestsLibrary.POST On Session rocketqa-api, /auth/register, json=\${payload}, headers=\${headers}</code>	00:00:00.097
Documentation: Sends a POST request on a previously created HTTP Session.		
Start / End / Elapsed: 20220527 18:56:05.567 / 20220527 18:56:05.664 / 00:00:00.097		
18:56:05.664 INFO POST Request : url=http://localhost:1302/auth/register path_url=/auth/register headers={'User-Agent': 'python-requests/2.27.1', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-type': 'application/json', 'Content-Length': '106'} body=b'{"name": "Luz de Mila", "cpf": "502.858.430-36", "email": "lumila@equiperocket.com", "password": "14&410"}'		
18:56:05.664 INFO POST Response : url=http://localhost:1302/auth/register status=201, reason=Created headers={'X-Powered-By': 'Express', 'Access-Control-Allow-Origin': '*', 'Access-Control-Allow-Methods': 'GET,PUT,POST,DELETE', 'Content-Type': 'application/json; charset=utf-8', 'Content-Length': '375', 'ETag': 'W/"177-py+7I+dE066qjKZaYDvML4TYILE"', 'Date': 'Fri, 27 May 2022 21:56:05 GMT', 'Connection': 'keep-alive', 'Keep-Alive': 'timeout=5'} body={'message': "User Created", "user": {"name": "Luz de Mila", "cpf": "502.858.430-36", "email": "lumila@equiperocket.com", "_id": "629148f5bd13d82c39b58453", "createdAt": "2022-05-27T21:56:05.578Z", "__v": 0}, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyOTB0OGY1YmQxM2Q4MmMzOWI1ODQ1MyIsImhhdCI6MTY1MzY4ODU2NSwiZXhwIjoxNjUzNzc0OTY1fQ.gs8p3MSP24WxLPWnK0sX-GSS6lyIaWgAvbpckOVTruI"}		
18:56:05.664 INFO \${resp} = <Response [201]>		
18:56:05.664 INFO \${resp} = <Response [201]>		
<div>-<div>KEYWORD</div></div>	<code>\${user_cpf} = BuiltIn.Set Variable \${resp.json()['user']['cpf']}</code>	00:00:00.001
Documentation: Returns the given values which can then be assigned to a variables.		
Start / End / Elapsed: 20220527 18:56:05.665 / 20220527 18:56:05.666 / 00:00:00.001		
18:56:05.666 INFO \${user_cpf} = 502.858.430-36		
<div>-<div>KEYWORD</div></div>	<code>\${resp} = services.Get User CPF \${user_cpf}</code>	00:00:00.010
Start / End / Elapsed: 20220527 18:56:05.666 / 20220527 18:56:05.676 / 00:00:00.010		
<div>-<div>KEYWORD</div></div>	<code>RequestsLibrary.Create Session rocketqa-api, \${base_url_api}</code>	00:00:00.000
Documentation: Create Session: create a HTTP session to a server		
Start / End / Elapsed: 20220527 18:56:05.667 / 20220527 18:56:05.667 / 00:00:00.000		
18:56:05.667 INFO Creating Session using : alias=rocketqa-api, url=http://localhost:1302, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=False, debug=0		
<div>-<div>KEYWORD</div></div>	<code>&{headers} = BuiltIn.Create Dictionary Content-type=application/json</code>	00:00:00.001
Documentation: Creates and returns a dictionary based on the given items.		
Start / End / Elapsed: 20220527 18:56:05.667 / 20220527 18:56:05.668 / 00:00:00.001		
18:56:05.668 INFO &{headers} = { Content-type=application/json }		
<div>+<div>KEYWORD</div></div>	<code>\${resp} = RequestsLibrary.GET On Session rocketqa-api, /auth/users/\${cpf}, headers=\${headers}</code>	00:00:00.008
18:56:05.676 INFO \${resp} = None		
<div>+<div>KEYWORD</div></div>	<code>RequestsLibrary.Status Should Be 200, \${resp}</code>	00:00:00.000

log.html

<div><div></div><div>KEYWORD</div></div> <div><code>\${resp} = services.Get User CPF \${user_cpf}</code></div>	00:00:00.010
<div>Start / End / Elapsed: 20220527 18:56:05.666 / 20220527 18:56:05.676 / 00:00:00.010</div> <div><div><div></div><div>KEYWORD</div></div><div>RequestsLibrary.Create Session rocketqa-api, \${base_url_api}</div></div>	00:00:00.000
<div>Documentation: Create Session: create a HTTP session to a server</div> <div>Start / End / Elapsed: 20220527 18:56:05.667 / 20220527 18:56:05.667 / 00:00:00.000</div> <div>18:56:05.667 INFO Creating Session using : alias=rocketqa-api, url=http://localhost:1302, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=False, debug=0</div>	
<div><div><div></div><div>KEYWORD</div></div><div><code>&{headers} = BuiltIn.Create Dictionary Content-type=application/json</code></div></div>	00:00:00.001
<div>Documentation: Creates and returns a dictionary based on the given items.</div> <div>Start / End / Elapsed: 20220527 18:56:05.667 / 20220527 18:56:05.668 / 00:00:00.001</div> <div>18:56:05.668 INFO &{headers} = { Content-type=application/json }</div>	
<div><div><div></div><div>KEYWORD</div></div><div><code>\${resp} = RequestsLibrary.GET On Session rocketqa-api, /auth/users/\${cpf}, headers=\${headers}</code></div></div>	00:00:00.008
<div>Documentation: Sends a GET request on a previously created HTTP Session.</div> <div>Start / End / Elapsed: 20220527 18:56:05.668 / 20220527 18:56:05.676 / 00:00:00.008</div> <div>18:56:05.676 INFO GET Request : url=http://localhost:1302/auth/users/502.858.430-36 path_url=/auth/users/502.858.430-36 headers={'User-Agent': 'python-requests/2.27.1', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive', 'Content-type': 'application/json'} body=None</div> <div>18:56:05.676 INFO GET Response : url=http://localhost:1302/auth/users/502.858.430-36 status=200, reason=OK headers={'X-Powered-By': 'Express', 'Access-Control-Allow-Origin': '*', 'Access-Control-Allow-Methods': 'GET,PUT,POST,DELETE', 'Content-Type': 'application/json; charset=utf-8', 'Content-Length': '194', 'ETag': 'W/"c2-Env+OuSynHH8RKwTedFNcvc4z7Q"', 'Date': 'Fri, 27 May 2022 21:56:05 GMT', 'Connection': 'keep-alive', 'Keep-Alive': 'timeout=5'} body={"message": "User Showed", "user": [{"_id": "629148f5bd13d82c39b58453", "name": "Luz de Mila", "cpf": "502.858.430-36", "email": "lumila@equiperocket.com", "createdAt": "2022-05-27T21:56:05.578Z", "__v": 0}]}</div> <div>18:56:05.676 INFO \${resp} = <Response [200]></div> <div>18:56:05.676 INFO \${resp} = None</div>	
<div><div><div></div><div>KEYWORD</div></div><div>RequestsLibrary.Status Should Be 200, \${resp}</div></div>	00:00:00.000
<div>Documentation: Fails if response status code is different than the expected.</div> <div>Start / End / Elapsed: 20220527 18:56:05.677 / 20220527 18:56:05.677 / 00:00:00.000</div>	

output.xml

```
logs > output.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <robot generator="Robot 5.0 (Python 3.10.4 on win32)" generated="20220527 20:45:42.590" rpa="false" schemaversion="3">
3 <suite id="s1" name="Get User" source="C:\rocketqa-test-robot\tests\users\get_user.robot">
4 <test id="s1-t1" name="Get Users List" line="7">
5 <kw name="Get Json" library="helpers">
6 <var>${list}</var>
7 <arg>users/list.json</arg>
8 <kw name="Get File" library="OperatingSystem">
9 <var>${json_file}</var>
10 <arg>${EXECDIR}/resources/fixtures/${file_name}</arg>
11 <doc>Returns the contents of a specified file.</doc>
12 <msg timestamp="20220527 20:45:42.703" level="INFO" html="true">Getting file '<a href="file:///C:\rocketqa-test-robot\resources\fixtures\users\list.json">C:\rocketqa-test-robo
13 <msg timestamp="20220527 20:45:42.704" level="INFO">${json_file} = {
14     "users": [
15         {
16             "name": "Clementina Jove",
17             "cpf": "143.168.090-75",
18             "email": "clement@rocketqa.com",
19             "password": "09%mar"
20         },
21         {
22             "name": "Lucca Miguel",
23             "cpf": "594.435.270-14",
24     ...</msg>
25 <status status="PASS" starttime="20220527 20:45:42.703" endtime="20220527 20:45:42.704"/>
26 </kw>
27 <kw name="Evaluate" library="BuiltIn">
28 <var>${json_dict}</var>
29 <arg>json.loads(${json_file})</arg>
30 <arg>json</arg>
31 <doc>Evaluates the given expression in Python and returns the result.</doc>
32 <msg timestamp="20220527 20:45:42.707" level="INFO">${json_dict} = {'users': [{'name': 'Clementina Jove', 'cpf': '143.168.090-75', 'email': 'clement@rocketqa.com', 'password': '09%
33 <status status="PASS" starttime="20220527 20:45:42.704" endtime="20220527 20:45:42.707"/>
34 </kw>
35 <msg timestamp="20220527 20:45:42.707" level="INFO">${list} = {'users': [{'name': 'Clementina Jove', 'cpf': '143.168.090-75', 'email': 'clement@rocketqa.com', 'password': '09%mar'}]
36 <status status="PASS" starttime="20220527 20:45:42.703" endtime="20220527 20:45:42.707"/>
37 </kw>
38 <for flavor="IN">
39 <var>${item}</var>
40 <value>@{list['users']}</value>
41 <iter>
42 <var name="${item}">{'name': 'Clementina Jove', 'cpf': '143.168.090-75', 'email': 'clement@rocketqa.com', 'password': '09%mar'}</var>
43 <kw name="Post Register" library="services">
44 <arg>${item}</arg>
45 <kw name="Create Session" library="RequestsLibrary">
46 <arg>rocketqa-api</arg>
47 <arg>${base_url_api}</arg>
48 <doc>Create Session: create a HTTP session to a server</doc>
49 <msg timestamp="20220527 20:45:42.709" level="INFO">Creating Session using : alias=rocketqa-api, url=http://localhost:1302, headers={}, cookies={}, auth=None, t
```

Projects Report

Generated
20220528 05:07:32 UTC-03:00
2 minutes 14 seconds ago

Summary Information

Status: All tests passed
Start Time: 20220528 05:07:31.753
End Time: 20220528 05:07:32.905
Elapsed Time: 00:00:01.152
Log File: log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	4	4	0	0	00:00:01	
Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						
Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Projects	4	4	0	0	00:00:01	
Projects . Get Project Code	1	1	0	0	00:00:00	
Projects . Get Projects	1	1	0	0	00:00:00	
Projects . Post Project	1	1	0	0	00:00:00	
Projects . Put Project	1	1	0	0	00:00:00	

Test Details

AllTagsSuitesSearch

Suite:

Test:

Include:

Exclude:

Search

Clear

Help



Obrigada, pessoal!

Referências:

- ✓ <https://robotframework.org/#/>
- ✓ <http://robotframework.org/robotframework/#user-guide>
- ✓ <https://www.sympla.com.br/sympla-play/NlcKbLDsVpyB51-kQlJB-EUCrcsc9inkGSVtFomDzA/38734>
- ✓ Automação de Testes – Robot Framework – autora: Debora Matos (Arquiteta de Testes na Radix)
- ✓ <https://github.com/dani-cavalcanti/TDC-DevTest>

