

Desafio #2

webservice

// Para este procedimento, é necessário que o computador do usuário tenha o git já instalado

1. clonar o repositório do Github :

```
$ git clone
```

```
https://github.com/alexandremot/inmetrics-challenge
```

```
λ git clone https://github.com/alexandremot/inmetrics-challenge
Cloning into 'inmetrics-challenge'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
```

o diretório terá a seguinte configuração

```
inmetrics-challenge
├── webservice
│   ├── features
│   │   ├── step_definitions
│   │   └── support
```

2. acessar a pasta webservice

```
λ cd webservice\
```

3. instalar o bundle do projeto (gems requeridas para o funcionamento)

```
$ bundle install
```

```
λ bundle install
Using backports 3.15.0
Using builder 3.2.3
Using bundler 2.0.2
Using cucumber-tag_expressions 1.1.1
Using gherkin 5.1.0
Using cucumber-core 3.2.1
Using cucumber-expressions 6.0.1
Using cucumber-wire 0.0.1
Using diff-lcs 1.3
Using multi_json 1.13.1
Using multi_test 0.1.2
Using cucumber 3.1.2
Using mime-types-data 3.2019.0331
Using mime-types 3.2.2
Using multi_xml 0.6.0
Using httparty 0.17.0
Using json 2.1.0
Using rspec-support 3.8.2
Using rspec-core 3.8.2
Using rspec-expectations 3.8.4
Using rspec-mocks 3.8.1
Using rspec 3.8.0
Bundle complete! 4 Gemfile dependencies, 22 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

4. executar o teste / aplicação com o cucumber

```
$ git clone
```

```
λ cucumber
# language: pt
Funcionalidade: Busca dados de filmes em API
  Como um usuário do sistema
  Eu quero realizar requisições na API SWAPI
  Recebendo os dados/informações de filmes como resposta

Cenário: Sucesso na transmissão da listagem de filmes
  # features/consulta_filmes.feature:13
  Dado que o usuário faça uma requisição
  # features/step_definitions/steps.rb:24
  Quando o sistema receber o JSON de resposta
  # features/step_definitions/steps.rb:29
  Entao o código de status de resposta deve ser igual a 200
  # features/step_definitions/steps.rb:33
  Entao o sistema deve exibir mensagem "HTTP response: 200 (ok)"
  !" # features/step_definitions/steps.rb:37
  HTTP response: 200 (ok)!
```

5. resultado esperado

```
Lista de filmes com George
Lucas e Rick McCallum
=====
A New Hope
The Phantom Menace
Attack of the Clones
Revenge of the Sith
=====

2 cenários (2 passed)
7 steps (7 passed)
0m13.996s
```

Desafio #3 | devops

// Para este procedimento, é necessário que o computador do usuário tenha o Docker Desktop instalado.

Segue abaixo as instruções “do zero”. Caso a máquina já tenha o Docker instalado, pular para o item 5

1. acessar a página para download do Docker Desktop :

<https://www.docker.com/products/docker-desktop>



2. selecionar a versão indicada para o sistema do usuário (Windows ou Linux) e realizar o download

// para esta etapa, será necessário realizar um cadastro no site da Docker Hub, porém o serviço não tem custo para testes



3. seguir com o processo de instalação do Docker Desktop

O link abaixo dá uma boa orientação para este procedimento

<https://docs.docker.com/docker-for-windows/install/>

4. uma vez instalado o Docker no computador do usuário, confirmar a versão

```
$ docker --version
```

```
λ docker --version
Docker version 19.03.1, build 74b1e89
```

5. “baixar” - realizar o pull - da imagem disponibilizada no repositório 

```
$ docker pull alexandremot/new_image:latest
```

```
λ docker pull alexandremot/new_image:latest
latest: Pulling from alexandremot/new_image
Digest: sha256:0e5a3631b1ce76056bb5079492ad52f634bb96f1a2a80ca0e455a32971613986
```

6. confirmar se a imagem foi baixada com sucesso

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alexandremot/new_image	latest	1fdb8e6cb874	27 hours ago	1.66GB

// a imagem disponibilizada tem 1.66 GB

```
// a imagem disponi
IMAGE ID      CREATED      SIZE
1fdb8e6cb874  27 hours ago 1.66GB
```

7. inicializar a imagem

```
$ docker start <nome_da_imagem>
```

```
λ docker start jenkins_cucumber
jenkins_cucumber
```

8. confirmar que esta foi inicializada com sucesso

```
$ docker ps -a
```

```
λ docker ps -a
CONTAINER ID   IMAGE      PORTS
486820c8f8ad   ubuntu    22/tcp
2ea90bdb99fb   alexandremot/new_image  22/tcp
f4fc0f8c00a7   ifdb8e6cb874  22/tcp
bba2000941e    ubuntu    22/tcp
```

9. acessar a o console (bash) do cointainer

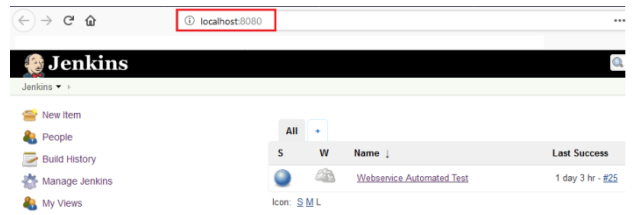
```
$ docker exec -i -t <nome_do_container> /bin/bash
```

```
λ docker exec -i -t jenkins_cucumber /bin/bash
root@bba2000941e:/#
```

10. iniciar o processo do Jenkins (já instalado e configurado neste container)

```
root@bba2000941e:/# /etc/init.d/jenkins start
Correct java version found
* Starting Jenkins Automation Server jenkins [ OK ]
```

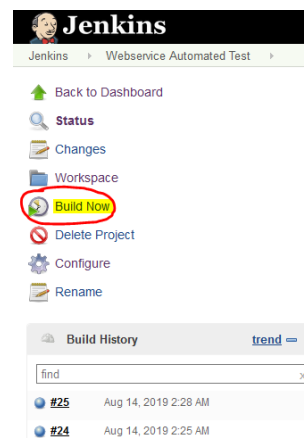
11. uma vez inicializado o serviço, acessar o endereço <http://localhost:8080/> no browser/navegador



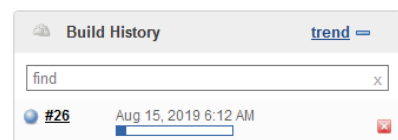
11. acessar o projeto de automação



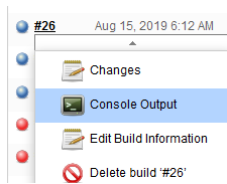
12. no menu do canto esquerdo, inicializar o job, clicando em “Build Now”



// note que a build inicializará o procedimento (testes)



// para acompanhar a evolução do teste,
basta clicar no numero da build e selecionar
a opção “Console Output”



12. resultado esperado

```
Lista de filmes com George
Lucas e Rick McCallum
=====
A New Hope
The Phantom Menace
Attack of the Clones
Revenge of the Sith
=====

2 scenarios (2 passed)
7 steps (7 passed)
0m23.431s
Finished: SUCCESS 😊
```