

AULA 11

AUTOMAÇÃO DE TESTE (WEB)

Richard Okubo

Programador Pleno





Olá, eu sou o Richard Okubo

PROGRAMADOR

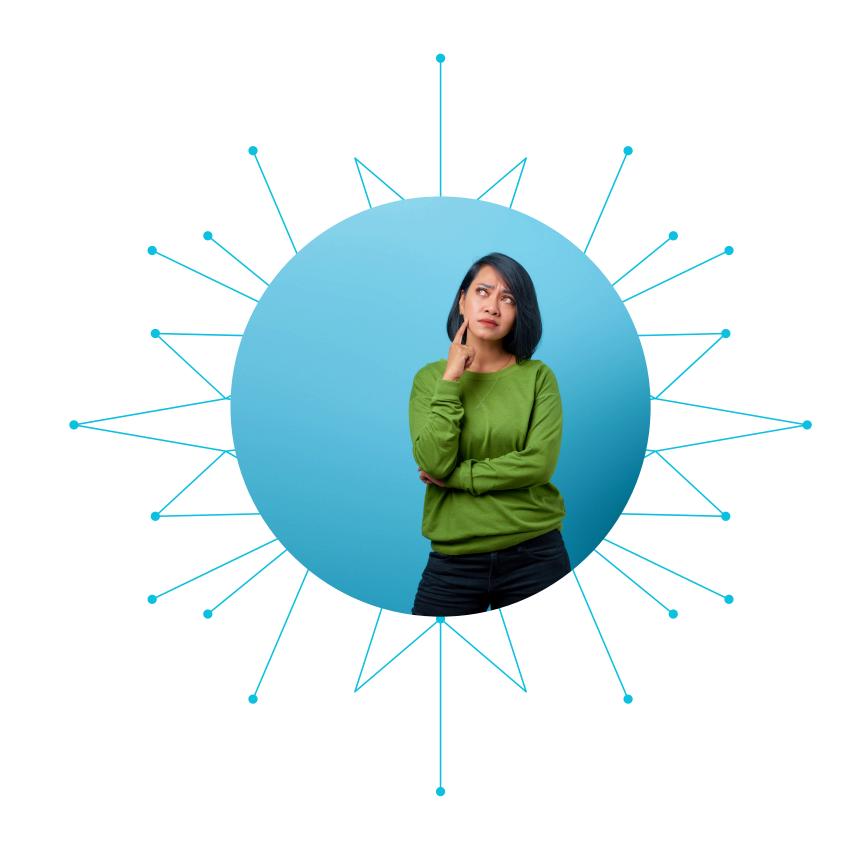
- Formação em Marketing / pós-graduando em Engenharia de Software
- Ex-residência em TIC/Software pelo Serratec (2021.2)
- Trabalho com desenvolvimento e qualidade de software a cerca de 3 anos na T2M.
- MAG Seguros e Petrobrás Transporte S.A.
- Membro dos times de sustentação e evolução dos sistemas da Transpetro, braço logístico da Petrobras.

OBSERVAÇÕES



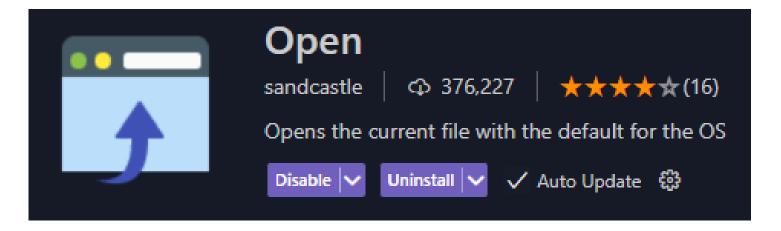
TÓPICOS DA AULA

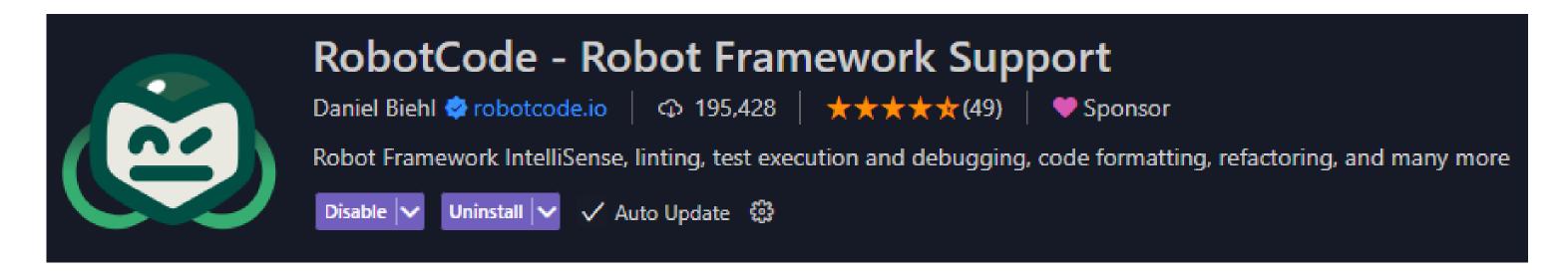
- Pré-requisitos
- Site SauceDemo
- Seções
- Palavras-chave
- Locators
- Verificações
- Page Object Model (POM)



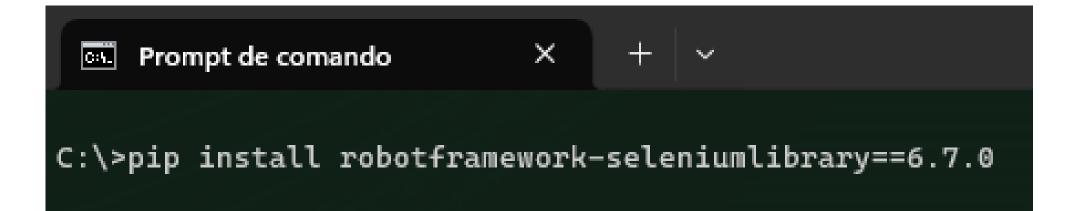
PRÉ-REQUISITOS







PRÉ-REQUISITOS



Prompt de comando X	+ ~
C:\>pip list	
Package	Version
attrs	25.1.0
certifi	2025.1.31
cffi	1.17.1
click	8.1.8
colorama	0.4.6
h11	0.14.0
idna	3.10
outcome	1.3.0.post0
pip	24.3.1
pycparser	2.22
PySocks	1.7.1
robotframework	7.2.2
nobotfnamowonk-pythonlibcono	1111
robotframework-seleniumlibrary	
secenium	4.28.1
sniffio	1.3.1
sortedcontainers	2.4.0
trio trio-websocket	0.29.0
	0.12.0
typing_extensions urllib3	4.12.2
websocket-client	2.3.0 1.8.0
wsproto	1.2.0
WSPIOCO	1.2.0
C:\>	

MÃO NA MASSA

SWAGLABS

Username Password LOGIN	
Accepted usernames are: standard_user locked_out_user problem_user performance_glitch_user	Password for all users: secret_sauce

Link: https://www.saucedemo.com/v1/

Seções

SETTINGS

Importação de bibliotecas, arquivos de recursos/variáveis Definição de metadados para suítes de teste e casos de teste

VARIABLES

Definir variáveis

KEYWORDS

Customização de palavras-chave a partir de outras existentes

TEST CASES

Criação de casos de teste a partir das palavras-chave disponíveis

```
*** Settings
   Variables
    Keywords
    Test Cases
```

Importação da biblioteca

SELENIUM LIBRARY

Biblioteca popular do Robot Framework que fornece um conjunto de palavras-chave para interagir com navegadores.

```
*** Settings ***

Library SeleniumLibrary

*** Test Cases ***
```

Link: https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Nomeando caso de teste

TEST CASES

Criaremos um caso de teste para realizar login no saucedemo com um usuário válido.

ABRIR PÁGINA WEB

Podemos utilizar a palavra-chave 'Open Browser' fornecida pela biblioteca 'SeleniumLibrary'. Passaremos dois parâmetros: *url* (endereço do site) e *browser* (navegador que usaremos).

```
1 *** Settings ***
2
3 Library SeleniumLibrary
4
5 *** Test Cases ***
6

> 7 TC01: realizar login com usuário válido
8 Open Browser url=https://www.saucedemo.com/v1/ browser=chrome
```

MAXIMIZAÇÃO/MINIMIZAÇÃO DE JANELAS

Vamos utilizar outra palavra-chave da biblioteca 'SeleniumLibrary'. Como o próprio nome diz, 'Maximize Browser Window' maximiza a janela do navegador atual, ajustando-a para ocupar todo o espaço disponível na tela. Existe também a palavra-chave 'Minimize Browser Window', que faz o inverso da anterior.

```
② 1 *** Settings ***
2
3 Library SeleniumLibrary
4
5 *** Test Cases ***
6
② 7 TC01: realizar login com usuário válido
8 Open Browser url=https://www.saucedemo.com/v1/ browser=chrome
9 Maximize Browser Window
10
```

FECHANDO NAVEGADOR

Agora utilizaremos a palavra-chave 'Close Browser' para fechar o nosso navegador.

DEFININDO UM VALOR DE ESPERA

Podemos também definir um valor de espera, como se fossemos pausar a execução do nosso teste por algum intervalo de tempo. Para isso, usamos a palavra-chave 'Sleep', nativa na própria biblioteca do Robot Framework.

Usaremos ela aqui apenas para fins de teste, só para visualizar melhor o navegador abrindo e fechando.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
 6
     TC01: realizar login com usuário válido
                         url=https://www.saucedemo.com/v1/
         Open Browser
                                                               browser=chrome
 8
         Maximize Browser Window
 9
         Sleep
10
         Close Browser
11
```

SETUP/TEARDOWN

Setup e Teardown são conceitos usados para definir ações que devem ser executadas antes e depois da execução dos testes. Por agora, usaremos 'Teardown' para garantirmos que, mesmo que o nosso teste falhe em algum ponto da execução, no final, ele deverá fechar o navegador.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
 6
     TC01: realizar login com usuário válido
                         url=https://www.saucedemo.com/v1/
         Open Browser
                                                               browser=chrome
 8
         Maximize Browser Window
 9
         Sleep
                  25
10
         [Teardown]
                       Close Browser
11
```

SETUP/TEARDOWN

Se quisermos, também podemos usar o 'Setup' para abrir o navegador. Ele funcionaria como uma pré-condição para o nosso teste.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
 6
     TC01: realizar login com usuário válido
                                    url=https://www.saucedemo.com/v1/
         [Setup]
                    Open Browser
                                                                         browser=chrome
 8
         Maximize Browser Window
         Sleep
                  25
10
         [Teardown]
11
                       Close Browser
12
```

RODANDO MAIS DE UMA PALAVRA-CHAVE

Se quisermos usar mais de uma palavra-chave em um 'Setup' ou 'Teardown', podemos usar 'Run Keywords' para isso. O 'AND' serve para concatenar as palavras-chave que serão executadas uma após a outra. O '...' é apenas uma quebra de linha.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
     TC01: realizar login com usuário válido
                    Run Keywords
         [Setup]
 8
                                    url=https://www.saucedemo.com/v1/
                    Open Browser
                                                                         browser=chrome
                                                                                           AND
                    Maximize Browser Window
10
11
         Sleep
12
         Teardown
                       Close Browser
13
```

LOCALIZANDO ELEMENTOS

O "bot" não consegue enxergar os elementos da página web. Precisamos fornecer instruções de como ele pode encontrá-los em um documento HTML. Para isso, podemos usamos algumas estratégias de busca (ou *locators*).

Strategy	Match based on	Example
id	Element id.	id:example
name	name attribute.	name:example
identifier	Either id or name.	identifier:example
class	Element class.	class:example
tag	Tag name.	tag:div
xpath	XPath expression.	<pre>xpath://div[@id="example"]</pre>
CSS	CSS selector.	css:div#example
dom	DOM expression.	<pre>dom:document.images[5]</pre>
link	Exact text a link has.	link:The example
partial link	Partial link text.	partial link:he ex
sizzle	Sizzle selector deprecated.	sizzle:div.example
data	Element data-* attribute	data:id:my_id
jquery	jQuery expression.	jquery:div.example
default	Keyword specific default behavior.	default:example

DIGITAR USERNAME

Agora que sabemos como localizar os elementos, podemos utilizar 'Input Text' para digitar no campo 'Username'. Passamos um *locator* que selecionamos e o *text* (para o nosso caso, usaremos um usuário válido).

Adicionamos outro 'Sleep' apenas para fins de teste e conseguirmos visualizar melhor o preenchimento desse campo.

```
*** Settings ***
               SeleniumLibrary
     Library
     *** Test Cases ***
     TC01: realizar login com usuário válido
                   Run Keywords
         [Setup]
8
                   Open Browser
                                   url=https://www.saucedemo.com/v1/
                                                                        browser=chrome
                   Maximize Browser Window
10
11
                 25
         Sleep
                       locator=id:user-name
12
                                              text=standard_user
         Input Text
13
         Sleep
                 25
14
         Teardown
                      Close Browser
15
```

DIGITAR SENHA

Vamos digitar a senha, mas usaremos 'Input Password'. Diferentemente de 'Input Text', ela não exibe o conteúdo textual nos logs de execução.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
     TC01: realizar login com usuário válido
8
         [Setup]
                    Run Keywords
                    Open Browser
                                    url=https://www.saucedemo.com/v1/
                                                                         browser=chrome
                                                                                           AND
                    Maximize Browser Window
10
11
         Sleep
                  25
                           locator=id:user-name
                                                    text=standard_user
12
         Input Text
                                                    password=secret_sauce
13
         Input Password
                           locator=css:#password
14
         Sleep
15
         Teardown
                       Close Browser
16
```

CLICAR NO BOTÃO

Por fim, usaremos 'Click button' para efetuar o nosso login.

```
*** Settings ***
                SeleniumLibrary
     Library
     *** Test Cases ***
 6
     TC01: realizar login com usuário válido
         [Setup]
                    Run Keywords
                                    url=https://www.saucedemo.com/v1/
                    Open Browser
                                                                         browser=chrome
                                                                                           AND
                    Maximize Browser Window
10
11
         Sleep
                  25
                                                    text=standard_user
12
         Input Text
                           locator=id:user-name
13
         Input Password
                           locator=css:#password
                                                    password=secret_sauce
                           locator=xpath://*[@id='login-button']
         Click Button
14
         Sleep
15
                  25
         [Teardown]
16
                       Close Browser
```

FINALIZAMOS O TESTE?



Realizando verificações

SHOULD...

Podemos verificar se, após o login com as credenciais válidas, o usuário foi redirecionado para outra página.

Nota: há diversas formas de realizar uma verificação como essa. Apenas para simplificar o nosso exemplo, usaremos essa verificação por agora.

```
*** Settings ***
               SeleniumLibrary
     Library
     *** Test Cases ***
     TC01: realizar login com usuário válido
         [Setup]
                   Run Keywords
                                   url=https://www.saucedemo.com/v1/
                   Open Browser
                                                                       browser=chrome
                   Maximize Browser Window
10
11
         Sleep
                2s
12
                          locator=id:user-name
                                                   text=standard user
         Input Text
13
                          locator=css:#password
                                                  password=secret sauce
         Input Password
                          locator=xpath://*[@id='login-button']
        Click Button
                   Get Location
15
         ${url}=
                                    /inventory.html
                          ${url}
16
         Should Contain
        Sleep 2s
         [Teardown]
18
                      Close Browser
19
```

Realizando verificações

SHOULD...

É sempre importante checarmos se as nossas verificações estão realmente realizando o teste que queremos. Para isso, podemos alterar os nossos dados de teste. Se estiver tudo certo, retornamos com o valor anterior e seguiremos a diante.

```
*** Settings ***
               SeleniumLibrary
     Library
     *** Test Cases ***
     TC01: realizar login com usuário válido
                   Run Keywords
         [Setup]
                   Open Browser
                                   url=https://www.saucedemo.com/v1/
                                                                       browser=chrome
                   Maximize Browser Window
10
11
         Sleep
                2s
                                                   text=standard user
12
         Input Text
                          locator=id:user-name
13
         Input Password
                          locator=css:#password
                                                  password=secret sauce
                          locator=xpath://*[@id='login-button']
14
         Click Button
        ${url}=
                   Get Location
15
        Should Contain
                          ${url}
16
                                    /iqs.html
        Sleep 2s
17
         [Teardown]
18
                      Close Browser
```

PAGE OBJECT MODEL (POM)

O padrão Page Object Model é uma forma de representar páginas e seus componentes em termos de código. Isso facilita na manutenção e reutilização do nosso script de teste. Nota: não estamos usando o conceito POM em sua totalidade. Há diversas formas de abordá-lo no Robot Framework.



VARIÁVEIS

Primeiro, vamos criar a seção "Variables". Perceba que na linha 7 adicionamos um comentário. Ele aqui servirá apenas para "fins estéticos", ou melhor, dividir nossas seções em subseções. Isso fará mais sentido quando criarmos outras variáveis, mas com propósitos diferentes.

Criamos um dicionário para representar a nossa página. Temos então a chave (o nome que vamos evocar para representar o elemento da página) e seu valor (o *locator* própriamente dito).

```
*** Settings ***
     *** Variables ***
     &{LOGIN PAGE}
           UsernameInput=id:user-name
10
           PassowordInput=css:#password
11
           LoginButton=xpath://*[@id='login-button']
12
13
     *** Test Cases ***
15
     TC01: realizar login com usuário válido
17
                   Run Keywords
         [Setup]
                   Open Browser
                                   url=https://www.saucedemo.com/v1/
18
                                                                        browser=chrome
                                                                                          AND
                   Maximize Browser Window
19
20
         Sleep
                 2s
                          locator id:user-name
                                                                   text=standard_user
21
         Input Text
22
         Input Password
                          locator css:#password
                                                                   password=secret_sauce
                          locator =xpath://*[@id='login-button']
23
         Click Button
                   Get Location
24
         ${url}=
        Should Contain
                          ${url}
                                  /inventory.html
25
26
         Sleep
27
         [Teardown]
                       Close Browser
```

Evocamos agora os elementos da página, ao invés dos locators. A medida que o código for crescendo, vocês compreenderão o benefício desse padrão de projeto.

```
*** Settings ***
     *** Variables ***
     &{LOGIN_PAGE}
           UsernameInput=id:user-name
           PassowordInput=css:#password
11
           LoginButton=xpath://*[@id='login-button']
12
13
     *** Test Cases ***
14
15
     TC01: realizar login com usuário válido
         [Setup]
                   Run Keywords
17
                                   url=https://www.saucedemo.com/v1/
18
                   Open Browser
                                                                       browser=chrome
                   Maximize Browser Window
19
         . . .
20
         Sleep
                25
                          locator ${LOGIN_PAGE.UsernameInput}
                                                                 text=standard_user
21
         Input Text
                          locator ${LOGIN_PAGE.PassowordInput}
22
                                                                 password=secret_sauce
         Input Password
                          locator ${LOGIN_PAGE.LoginButton}
23
         Click Button
                   Get Location
         ${url}=
                                   /inventory.html
         Should Contain
                          ${url}
25
         Sleep 2s
26
                      Close Browser
27
         [Teardown]
```

Outro caso de teste

TEST CASES

Podemos criaremos um outro caso de teste para realizar login no saucedemo com um usuário bloqueado.

Outro caso de teste

O código é semelhante ao teste anterior, mas só alteramos a verificação. Queremos verificar se a mensagens de erro foi exibida em tela. Adicionamos o *locator* no POM agora.

```
Settings ***
      *** Variables ***
      &{LOGIN_PAGE}
             UsernameInput=id:user-name
             PassowordInput=css:#password
            LoginButton=xpath://*[@id='login-button']
 13
             ErrorMessage=data:test:error
 14
      *** Test Cases ***
17 > TC01: realizar login com usuário válido…
      TC02: realizar login com usuário bloqueado
31
          [Setup]
                    Run Keywords
                                   url=https://www.sautedemo.com/v1/
32
                    Open Browser
                                                                        browser=chrome
                    Maximize Browser Window
34
                  2s
          Sleep
                                                                   text=locked out user
          Input Text
                           locator=${LOGIN PAGE.UsernameInpu}
36
          Input Password
                           locator=${LOGIN PAGE.PassowordInput
                                                                  password=secret_sauce
                           locator-$(LOGIN PAGE LoginButton)
37
                                Get Text    locator=${LOGIN PAGE.ErrorMessage}
          ${mensagem obtida}=
          ${mensagem_esperada}=
                                  Set Variable
                                                  Epic sadface: Sorry, this user has been locked out.
          Should Be Equal ${mensagem obtida}
                                                  ${mensagem esperada}
          Sleep 2s
42
          [Teardown]
                      Close Browser
43
```

Podemos notar que temos lógica semelhante em ambos os casos de teste. É importante simplificar a lógica para não atrapalhar na legibilidade e na manutenção dos nossos testes.

```
1 > *** Settings ***
5 > *** Variables *** ...
15 *** Test Cases ***
    TC01: realizar login com usuário válido
                  Run Keywords
                                url=https://www.saucedemo.com/v1/
                  Open Browser
                                                                   browser=chrome
                  Maximize Browser Window
        Sleep 2s
        Input Text
                        locator=${LOGIN PAGE.UsernameInput}
                                                              text=standard user
        Input Password
                        locator=${LOGIN_PAGE.PassowordInput}
                                                             password=secret sauce
        Click Button
                        locator=${LOGIN_PAGE.LoginButton}
        Should Contain
                        ${url} /inventory.html
        Sleep
              25
                    Close Browser
        Teardown
        [Setup]
                                url=https://www.saucedemo.com/v1/
                                                                   browser=chrome
                  Maximize Browser Window
        Sleep
              2s
        Input Text
                        locator=${LOGIN_PAGE.UsernameInput}
                                                             text=locked out user
                        locator=${LOGIN_PAGE.PassowordInput}
                                                             password=secret sauce
        Input Password
        Click Button
                        locator=${LOGIN PAGE.LoginButton}
                             Epic sadface: Sorry, this user has been locked out.
        ${mensagem_esperada}= Set Variable
        Should Be Equal
                         ${mensagem obtida}
                                              ${mensagem esperada}
        Sleep 2s
                    Close Browser
        [Teardown]
```

Separamos a URL e o BROWSER na seção de variáveis.

```
*** Variables ***
 6
     ### Dados de configuração ###
                    https://www.saucedemo.com/v1/
     ${URL}
 9
     ${BROWSER}
                    chrome
10
11
     # Page Object Model (POM)
12
13
     &{LOGIN PAGE}
14
            UsernameInput=id:user-name
15
            PassowordInput=css:#password
16
17
            LoginButton=xpath://*[@id='login-button']
18
            ErrorMessage=data:test:error
```

```
*** Test Cases ***
     TC01: realizar login com usuário válido
         [Setup]
                   Run Keywords
                                                browser=${BROWSER}
                   Open Browser
                                  url:${URL}
                   Maximize Browser Window
        Sleep 2s
        Input Text
                          locator=${LOGIN PAGE.UsernameInput}
                                                                text=standard user
                          locator=${LOGIN PAGE.PassowordInput}
        Input Password
                                                                password=secret sauce
                          locator=${LOGIN PAGE.LoginButton}
        Click Button
        ${url}= Get Location
31
        Should Contain
                         ${url}
                                   /inventory.html
        Sleep 2s
        [Teardown] Close Browser
     TC02: realizar login com usuário bloqueado
                   Run Keywords
         [Setup]
                                                browser: ${BROWSER}
                   Open Browser
                                  url: ${URL}
                   Maximize Browser Window
        Sleep 2s
        Input Text
                          locator=${LOGIN_PAGE.UsernameInput}
                                                                text=locked out user
        Input Password
                          locator=${LOGIN_PAGE.PassowordInput}
                                                                password=secret sauce
        Click Button
                          locator=${LOGIN PAGE.LoginButton}
        ${mensagem_obtida}= Get Text locator=${LOGIN_PAGE.ErrorMessage}
                                               Epic sadface: Sorry, this user has been locked out.
        ${mensagem esperada}=
                                Set Variable
         Should Be Equal ${mensagem_obtida}
                                               ${mensagem_esperada}
        Sleep 2s
         [Teardown]
                      Close Browser
```

Passamos o Setup/Teardown para a seção de configuração. Removemos também o 'Sleep' do código, pois não é mais necessário.

```
*** Settings ***
               SeleniumLibrary
     Test Setup
                     Run Keywords
                                     url=${URL}
                                                   browser=${BROWSER}
                     Open Browser
                     Maximize Browser Window
     Test Teardown
                     Close Browser
     *** Variables *** ···
     *** Test Cases ***
27
     TC01: realizar login com usuário válido
                          locator=${LOGIN_PAGE.UsernameInput}
29
         Input Text
                                                                 text=standard user
                          locator=${LOGIN_PAGE.PassowordInput}
                                                                 password=secret_sauce
        Input Password
                          locator=${LOGIN_PAGE.LoginButton}
31
        Click Button
32
        ${url}= Get Location
                          ${url}
        Should Contain
                                    /inventory.html
34
     TC02: realizar login com usuário bloqueado
         Input Text
                          locator=${LOGIN_PAGE.UsernameInput}
                                                                 text=locked out user
                          locator=${LOGIN_PAGE.PassowordInput}
37
         Input Password
                                                                 password=secret_sauce
        Click Button
                          locator=${LOGIN_PAGE.LoginButton}
                                         locator=${LOGIN_PAGE.ErrorMessage}
                               Get Text
         ${mensagem_obtida}=
                                                Epic sadface: Sorry, this user has been locked out.
        ${mensagem esperada}=
                                Set Variable
41
         Should Be Equal
                          ${mensagem_obtida}
                                                 ${mensagem esperada}
42
```

Criaremos nossa própria palavra-chave para simplificar a lógica de login.

```
*** Variables ***
     *** Keywords ***
     # Acões
     Realizar login com ${username} e ${password}
                                                                   text=${username}
         Input Text
                           locator=${LOGIN PAGE.UsernameInput}
                           locator=${LOGIN PAGE.PassowordInput}
                                                                   password=${password}
         Input Password
         Click Button
                           locator=${LOGIN PAGE.LoginButton}
     *** Test Cases ***
35
36
     TC01: realizar login com usuário válido
37
         Realizar login com standard_user e secret_sauce
38
         ${url}= Get Location
39
                                     /inventory.html
         Should Contain
                           ${url}
40
41
     TC02: realizar login com usuário bloqueado
42
         Realizar login com locked_out_user e secret_sauce
43
         ${mensagem_obtida}=
                                Get Text
                                           locator=${LOGIN_PAGE.ErrorMessage}
44
         ${mensagem_esperada}=
                                                  Epic sadface: Sorry, this user has been locked out.
                                  Set Variable
45
                                                  ${mensagem esperada}
46
         Should Be Equal
                            ${mensagem obtida}
47
```

Vamos separar a nossa massa de teste e deixá-las na seção 'Variables'

```
### Massa de teste ###
26
27
     ${USUARIO_VALIDO}
                             standard_user
28
     ${USUARIO_BLOQUEADO}
                             locked_out_user
29
                             secret_sauce
30
     ${SENHA}
31
     *** Keywords ***...
     *** Test Cases ***
41
42
     TC01: realizar login com usuário válido
43
         Realizar login com ${USUARIO VALIDO} e ${SENHA}
44
         ${url}=
                    Get Location
45
         Should Contain
                                     /inventory.html
                           ${url}
46
47
     TC02: realizar login com usuário bloqueado
48
         Realizar login com ${USUARIO BLOQUEADO} e ${SENHA}
49
         ${mensagem obtida}=
                                            locator=${LOGIN PAGE.ErrorMessage}
                                Get Text
50
         ${mensagem_esperada}=
                                                  Epic sadface: Sorry, this user has been locked out.
51
                                 Set Variable
                            ${mensagem_obtida}
                                                  ${mensagem esperada}
         Should Be Equal
52
```

Também vamos armazenar a lógica de verificação dos nossos testes na seção 'Keywords'

```
# Verificações
     Verificar se conseguiu realizar o login corretamente
                    Get Location
         ${url}=
44
         Should Contain
                                    /inventory.html
                           ${url}
45
46
     Verificar se não foi possivel realizar o login
47
                               Get Text
         ${mensagem obtida}=
                                           locator=${LOGIN PAGE.ErrorMessage}
48
         ${mensagem_esperada}=
                                  Set Variable
                                                 Epic sadface: Sorry, this user has been locked out.
49
                            ${mensagem obtida}
         Should Be Equal
                                                  ${mensagem_esperada}
50
     *** Test Cases ***
53
     TC01: realizar login com usuário válido
         Realizar login com ${USUARIO_VALIDO} e ${SENHA}
55
         Verificar se conseguiu realizar o login corretamente
56
57
     TC02: realizar login com usuário bloqueado
         Realizar login com ${USUARIO_BLOQUEADO} e ${SENHA}
59
         Verificar se não foi possivel realizar o login
60
61
```

Podemos usar tags em nossos testes. Isso é útil se quisermos executar apenas casos com uma determinada tag.

```
*** Test Cases ***
52
53
     TC01: realizar login com usuário válido
54
55
         [Tags]
                   FLUXO-POSITIVO
                                     RE0001
         Realizar login com ${USUARIO VALIDO} e ${SENHA}
56
         Verificar se conseguiu realizar o login corretamente
57
58
     TC02: realizar login com usuário bloqueado
59
60
                   FLUXO-NEGATIVO
                                     REQ002 <
         Tags
         Realizar login com ${USUARIO_BLOQUEADO} e ${SENHA}
61
         Verificar se não foi possivel realizar o login
62
```

Documentação também pode ser adicionada aos nossos casos de teste, palavras-chave e no arquivo de teste.

```
*** Settings ***
                       IQS - Automação de teste (WEB)
     Documentation
                       Exemplo de automação com Robot Framework e SeleniumLibrary.
     Library
                SeleniumLibrary
1 > *** Settings ***
11 > *** Variables ***...
32 > *** Keywords ***...
     *** Test Cases ***
53
    TC01: realizar login com usuário válido
54
        [Tags] FLUXO-POSITIVO REQ001
55
        [Documentation] Este caso de teste valida o login de um usuário
56
        Realizar login com ${USUARIO_VALIDO} e ${SENHA}
57
        Verificar se conseguiu realizar o login corretamente
58
59
     TC02: realizar login com usuário bloqueado
                FLUXO-NEGATIVO REQ002
61
        [Documentation] Este caso de teste valida o bloqueio de um usuário
62
        Realizar login com ${USUARIO BLOQUEADO} e ${SENHA}
63
        Verificar se não foi possivel realizar o login
64
```

É possível escrever os casos de teste no "estilo BDD". Nota: não usaremos em nossos testes.

```
② 1 Language: Brazilian Portuguese
2
3 *** Settings ***
```

É possível escrever os casos de teste no "estilo BDD". Nota: não usaremos em nossos testes.

```
② 1 Language: Brazilian Portuguese
2
3 *** Settings ***
```

OUTROS TÓPICOS

- Esperas implícitas e explícitas
- Importação de recursos
- Tratativas para flaky test
- Smoke testing x Sanity testing
- Testes distribuídos
- Código Python
- RemoteInterface
- Screenplay pattern



