



## Aula 8

- Refazer a aula
- Automatizar os testes do Sauce Demo:
  - 1) Login:
    - a) Login com credenciais válidas
    - b) Login com username inválido
    - c) Login com senha inválida
    - d) Login com username vazio
    - e) Login com senha vazia
  - 2) AddToCart
    - a) Adicionar um item ao carrinho
    - b) Remover um item do carrinho
  - 3) Checkout
    - a) Checkout com sucesso

SauceDemo	cypress
>Login Functionality	
✓ Login valid	
✓ Login invalid Username	
✓ Login invalid Password	
✓ Login with empty Username	
✓ Login with empty Password	
AddToCart Functionality	
✓ Adding an item from Products screen	
✓ Removing an item from Products screen	
Checkout Functionality	
✓ Checkout with success	

Tech Fixe  
prof Herzio Pinto

Carmen Cabral  
30/11/2023

## Índice

O QUE É CYPRESS? .....	3
Por que escolher o Cypress? .....	3
Instalar o Cypress .....	3
Executar o Cypress .....	4
Interface gráfica do Cypress .....	4
E2E Testing .....	4
Navegadores / Browsers.....	5
Localhost no Chrome .....	5
Criar uma spec / teste.....	6
Abrir o Cypress no navegador.....	7
Arrow Function => .....	7
Executar o Cypress no terminal com browser padrão .....	10
Executar o Cypress no terminal e mostrar no browser padrão.....	12
Executar o Cypress no terminal com o chrome .....	12
Executar o Cypress no terminal com browser padrão e um spec/teste específico .....	12
Locators: encontrar itens no navegador .....	13
SINTAXE.....	13
LOCATOR COM SELETORES CSS .....	13
FAZER REFERÊNCIA .....	13
LOCATOR COM XPATH E INSTALAR O PLUGIN.....	16
Assertions: verificar/validar condições .....	18
IMPLÍCITAS / BUILT-IN .....	18
EXPLÍCITAS .....	19
Diferença entre var e let.....	19
<b>Exercícios: Refazer a aula e automatizar os testes do Sauce Demo.....</b>	21
1 Login / describe .....	21
a) LOGIN COM CREDENCIAIS VÁLIDAS / it .....	22
b) LOGIN COM USERNAME INVÁLIDO / it .....	23
c) LOGIN COM SENHA INVÁLIDA / it .....	23
d) LOGIN COM USERNAME VAZIO / it.....	24
e) LOGIN COM SENHA VAZIA / it.....	25
2 Add to cart / describe.....	25
a) ADICIONAR UM ITEM AO CARRINHO / it .....	25
b) REMOVER UM ITEM DO CARRINHO / it .....	26
3 Checkout / describe .....	27
a) CHECKOUT COM SUCESSO / it .....	27

## O QUE É CYPRESS?

Ferramenta de testes para páginas web modernas.

### Antes do Cypress

- \***Framework:** Mocha, Jasmine.
- \***Assertion library:** Chai, Expect.js.
- \*Instalar o **Selenium**.
- \***Wrapper:** Web driver
- \***Libraries** adicionais
- Com o Cypress:** All-in-one

### Por que escolher o Cypress?

- \*Não usa Selenium.
- \*Foca em fazer muito bem **end-to-end testing**.
- \*Funciona com qualquer tipo de framework de front-end (é compatível)
- \*Escrito somente com **JavaScript**.
- \*É para DEVs e para QAs.
- \*É muito rápido.
- \*Tem interface gráfica.

### Instalar o Cypress

No VS Code, Abrir o terminal - CTRL + C:

1 Criar o package.json

Digitar:

```
npm init -y
```

The screenshot shows the VS Code interface with the terminal tab selected. The command 'npm init -y' is being typed into the terminal window. The terminal output area is currently empty.

2 Instalar o Cypress (cria a pasta node\_modules)

Digitar:

```
npm install cypress
```

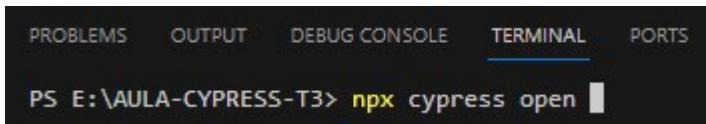
The screenshot shows the VS Code interface with the terminal tab selected. The command 'npm install cypress' is being typed into the terminal window. The terminal output area is currently empty.

The screenshot shows the terminal output after running 'npm install cypress'. It displays two 'npm notice' messages warning about the deprecation of TLS 1.0 and 1.1, and encourages the use of TLS 1.2 or higher. It also shows that the installation took 4 seconds and that 40 packages are looking for funding.

```
PS E:\AULA-CYPRESS-T3> npm install cypress
npm notice Beginning October 4, 2021, all connections to the npm registry - including for package installation - must use TLS 1.2 or higher. You are currently using plaintext http to connect. Please visit the GitHub blog for more information: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/
npm notice Beginning October 4, 2021, all connections to the npm registry - including for package installation - must use TLS 1.2 or higher. You are currently using plaintext http to connect. Please visit the GitHub blog for more information: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/
up to date in 4s
40 packages are looking for funding
  run `npm fund` for details
```

## Executar o Cypress

`npx cypress open`



## Interface gráfica do Cypress

The Cypress graphical interface shows the following screen:

### Welcome to Cypress!

Review the differences between each testing type →

#### E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

Not Configured

#### Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

Not Configured

## E2E Testing

Clicar em 'Continue'.

The Cypress graphical interface shows the following screen:

### Configuration files

We added the following files to your project:

- cypress.config.js**  
The Cypress config file for E2E testing.
- cypress\support\e2e.js**  
The support file that is bundled and loaded before each E2E spec.
- cypress\support\commands.js**  
A support file that is useful for creating custom Cypress commands and overwriting existing ones.
- cypress\fixtures\example.json**  
Added an example fixtures file/folder

**Continue**

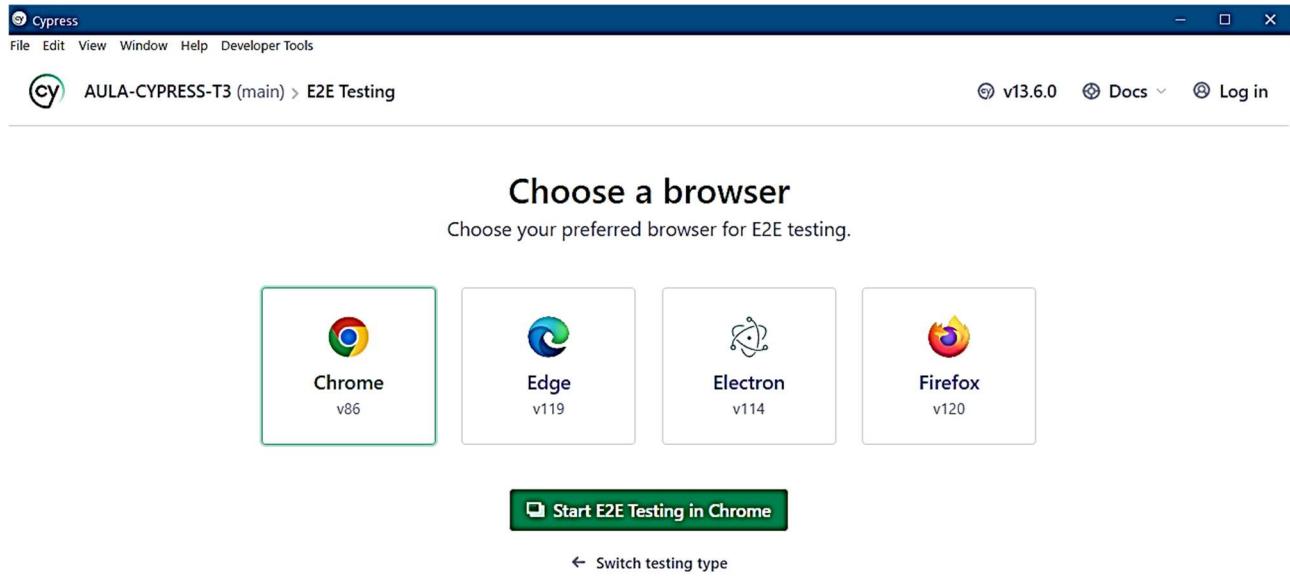
## Navegadores / Browsers

**Electron:** nativo do Cypress.

**Chrome:** tem instalado no meu computador.

**Edge:** se tiver o Windows instalado.

**Safari:** se tiver um MAC.

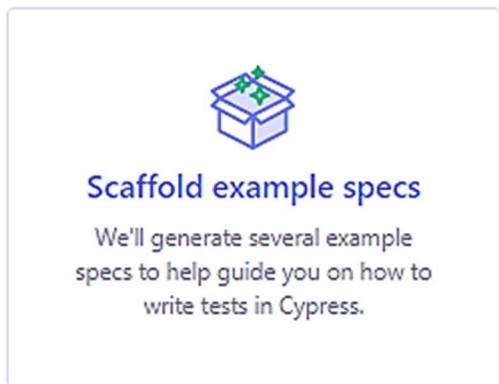


**Clicar no Chrome de depois no botão 'Start E2E Testing in Chrome'**

## Localhost no Chrome

**Cada spec é um arquivo de teste.**

## Scaffold example specs



## Criar uma spec / teste

Enter the path for your new spec X

cypress\e2e\MyFirstTest.cy.js

 **Create new spec**  
We'll generate a template spec file which can be used to start testing your application.

Create spec Back

Great! The spec was successfully added X

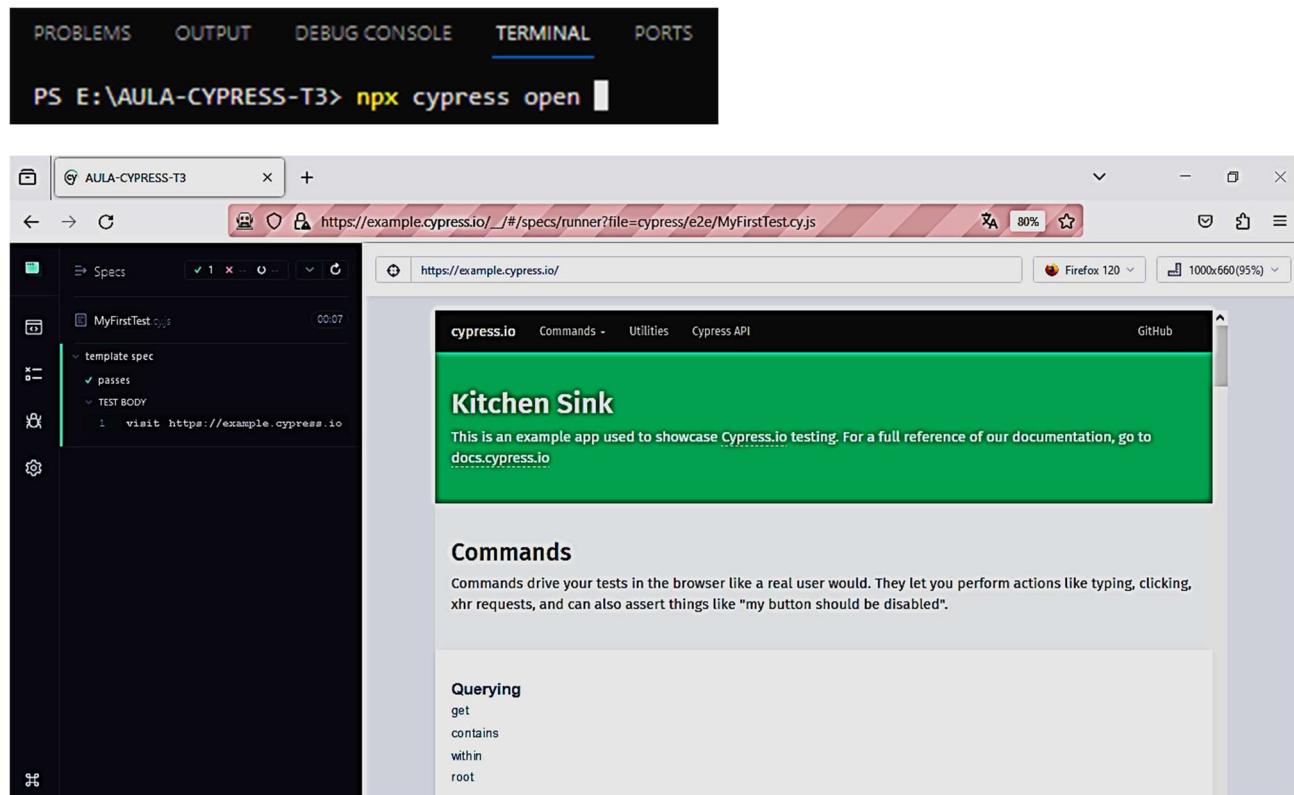
✓ cypress\e2e\MyFirstTest.cy.js ^

```
1 describe('template spec', () => {
2   it('passes', () => {
3     cy.visit('https://example.cypress.io')
4   })
5 })
```

Okay, run the spec + Create another spec

Abrir o Cypress no navegador

`npm cypress open`



Arrow Function =>

```
JS MyFirstTest.cy.js U X
AULA-CYPRESS-T3 > cypress > e2e > JS MyFirstTest.cy.js > ...
1  describe('template spec', () => {
2    it('passes', () => {
3      cy.visit('https://example.cypress.io')
4    })
5  })
```

The screenshot shows a Cypress test runner window and a browser window side-by-side.

**Cypress Test Runner:**

- File: MyFirstTest.cy.js
- Specs: 1 passed, 0 failed, 0 pending.
- Test: passes (0:17)
- Test Body:

```
1 describe('template spec', () => {
2   it('passes', () => {
3     cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
4     cy.title().should('eq', 'OrangeHRM')
5   })
6 })
```

**Browser View:**

- Address bar: https://opensource-demo.orangehrmlive.com/\_/#/specs/runner?file=cypress/e2e/MyFirstTest.cy.js
- User Agent: Chrome 86
- Resolution: 1000x660 (68%)
- Page Content:
  - OrangeHRM logo: OPEN SOURCE HR MANAGEMENT
  - Header: Login
  - Form fields:
    - Username: Admin  
Password: admin123
    - Username input field
    - Password input field
  - Login button
  - Forgot your password?
  - Footer: OrangeHRM OS 5.5  
© 2005 - 2023 OrangeHRM, Inc. All rights reserved.

**JS MyFirstTest.cy.js X**

```
AULA-CYPRESS-T3 > cypress > e2e > JS MyFirstTest.cy.js > ⚡ describe('template spec') callback
  1  describe('template spec', () => {
  2    it('TC1 - Verificar o título da página - Positivo', () => {
  3      cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
  4      cy.title().should('eq', 'OrangeHRM')
  5    })
  6
  7    it('TC2 - Verificar o título da página - Negativo', () => {
  8      cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
  9      cy.title().should('eq', 'OrangeHRM123')
 10  })
 11
 12})
```

AULA-CYPRESS-T3    +

← → 🔍 Não seguro | opensource-demo.orangehrmlive.com/\_/#/specs/runner?file=cypress/e2e/MyFirstTest.cy.js    🌐 🔎 ☆ ⓘ Atualizar :)

O Chrome está sendo controlado por um software de teste automatizado.

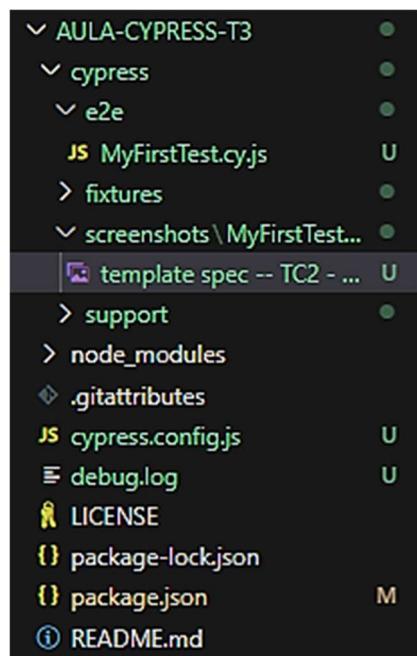
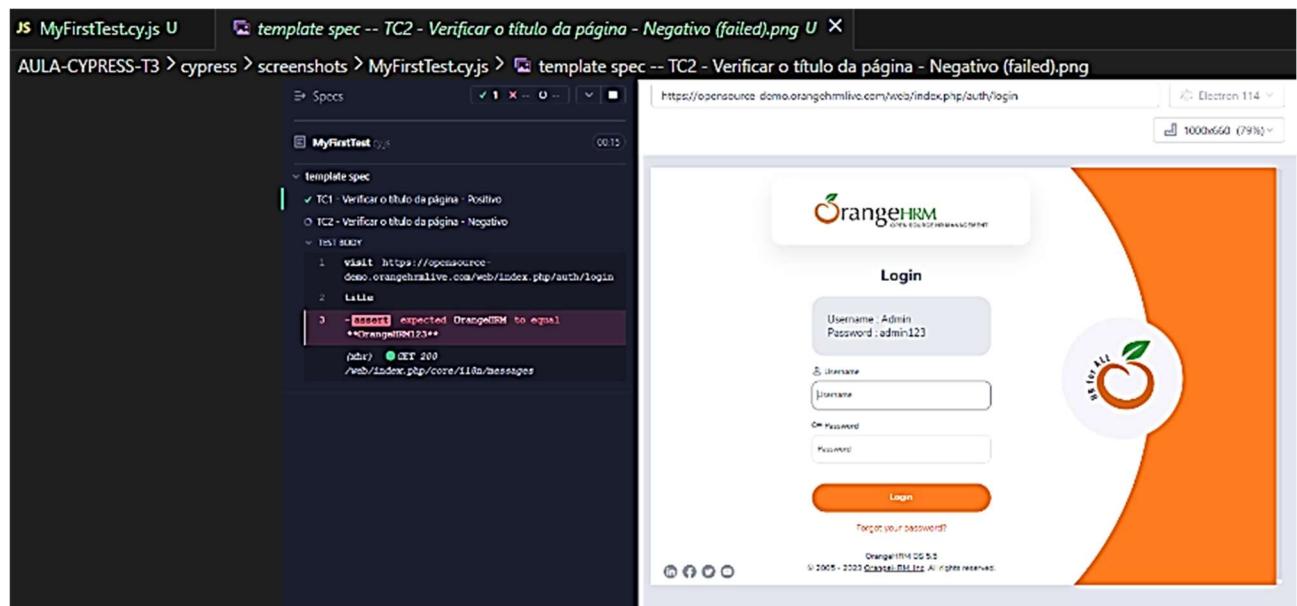
Executar o Cypress no terminal com browser padrão

`npx cypress run`

**Não abre a interface gráfica, navegador, para rodar o teste. Mostra o resultado no terminal. O Cypress utiliza o seu browser padrão: Electron.**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\AULA-CYPRESS-T3> npx cypress run
```

Gera screenshot com a falha:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\AULA-CYPRESS-T3> npx cypress run

DevTools listening on ws://127.0.0.1:60458/devtools/browser/8cbcf905-f46a-463d-b693-53c6d71e8659
=====
(Run Starting)

Cypress: 13.6.0
Browser: Electron 114 (headless)
Node Version: v20.9.0 (C:\Program Files\nodejs\node.exe)
Specs: 1 found (MyFirstTest.cy.js)
Searched: cypress/e2e/**/*.cy.{js,jsx,ts,tsx}

=====

Running: MyFirstTest.cy.js (1 of 1)

template spec
✓ TC1 - Verificar o título da página - Positivo (5076ms)
1) TC2 - Verificar o título da página - Negativo

1 passing (15s)
1 failing

1) template spec
  TC2 - Verificar o título da página - Negativo:
    Timed out retrying after 4000ms
    + expected - actual
      - 'OrangeHRM'
      +'OrangeHRM123'

    at Context.eval (webpack:///aula-cypress-t3./cypress/e2e/MyFirstTest.cy.js:9:15)

=====

(Results)

Tests: 2
Passing: 1
Failing: 1
Pending: 0
Skipped: 0
Screenshots: 1
Video: false
Duration: 15 seconds
Spec Ran: MyFirstTest.cy.js

=====

(Screenshots)

- E:\AULA-CYPRESS-T3\screenshots\MyFirstTest.cy.js\template spec -- TC2 - Verificar o título da página - Negativo (failed).png
=====

(Run Finished)

          Spec                               Tests  Passing  Failing  Pending  Skipped
  ✘ MyFirstTest.cy.js                      00:15      2       1       1       -       -
  ✘ 1 of 1 failed (100%)                  00:15      2       1       1       -       -

```

Executar o Cypress no terminal e mostrar no browser padrão

```
npx cypress run --headed
```

**Abre a interface gráfica, o navegador, para rodar o teste e depois fecha, mostrando o resultado no terminal. O Cypress utiliza o seu browser padrão: Electron.**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\AULA-CYPRESS-T3> npx cypress run --headed
```

Executar o Cypress no terminal com o chrome

```
npx cypress run --browser chrome
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\AULA-CYPRESS-T3> npx cypress run --browser chrome
```

DevTools listening on ws://127.0.0.1:60886/devtools/browser/cac81948-cdce-4c9d-8c2b-4ace82a50abc

=====

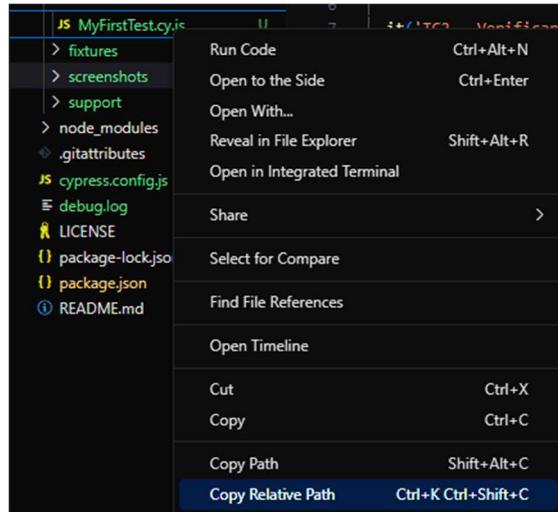
(Run Starting)

Cypress:	13.6.0
Browser:	Chrome 86 (headless)
Node Version:	v20.9.0 (C:\Program Files\nodejs\node.exe)
Specs:	1 found (MyFirstTest.cy.js)
Searched:	cypress/e2e/**/*.cy.{js,jsx,ts,tsx}

Executar o Cypress no terminal com browser padrão e um spec/teste específico

**Copiar o path do arquivo de teste/spec: clicar com botão direito em cima do arquivo e clicar em 'Copy Relative Path'. Colar no terminal após '--spec'.**

```
npx cypress run --spec cypress\e2e\MyFirstTest.cy.js
```



```
Cypress:      13.6.0
Browser:     Electron 114 (headless)
Node Version: v20.9.0 (C:\Program Files\nodejs\node.exe)
Specs:        1 found (MyFirstTest.cy.js)
Searched:    E:\AULA-CYPRESS-T3\cypress\e2e\MyFirstTest.cy.js

Running:  MyFirstTest.cy.js          (1 of 1)
```

## Locators: encontrar itens no navegador

\*Seletores CSS

\*Xpath

### SINTAXE

\*`cy.get(Locator)`

\*`cy.xpath(Locator)`

### LOCATOR COM SELETORES CSS

\* **tag + id >> tag#id**

Ex. `input#search`

\* **tag + classe >> tag.class**

Ex. `input.alternative-input__field--icon.alternative-input__field--cancel.alternative-input__field`

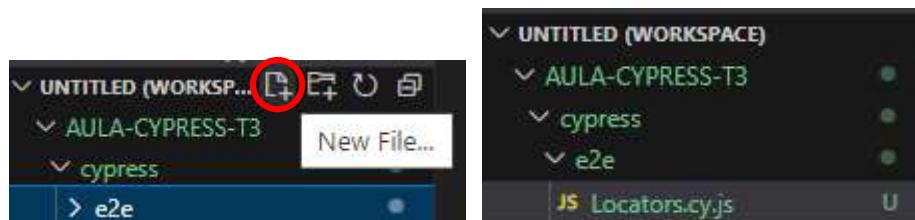
\* **tag + atributo -> tag[attribute='value']**

Ex. `input[placeholder='O que estás à procura?']`

\* **tag + class + atributo -> tag.class[attribute='value']**

Ex. `input.alternative-input__field--icon.alternative-input__field--cancel.alternative-input__field[placeholder='O que estás à procura?']`

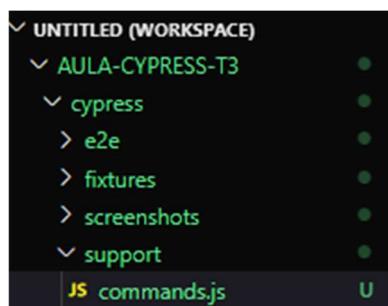
## Criar o arquivo 'Locators.cy.js'



## FAZER REFERÊNCIA

Server para retirar a linha gerada no início: `const { describe } = require("mocha")` do arquivo '`Locators.cy.js`'.

Incluir no final do arquivo '`commands.js`' o código: `/// <reference types="Cypress"/>`.



## CÓDIGO E PASSO A PASSO DO ARQUIVO LOCATORS.JS

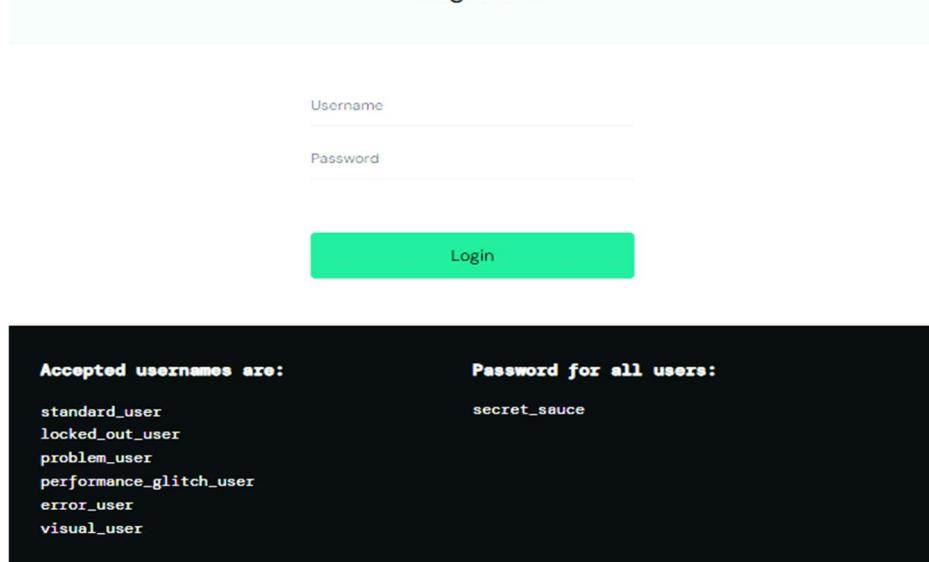
### 1 Visitar página:

```
//Visitar a página
cy.visit('https://www.saucedemo.com/')
```

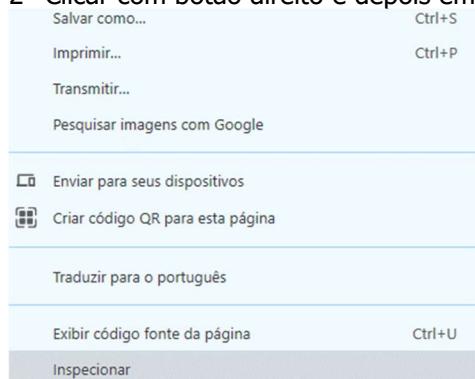
### 2 Inspecionar código no browser / Encontrar os seletores e elementos.

1º Abrir o site: <https://www.saucedemo.com/>

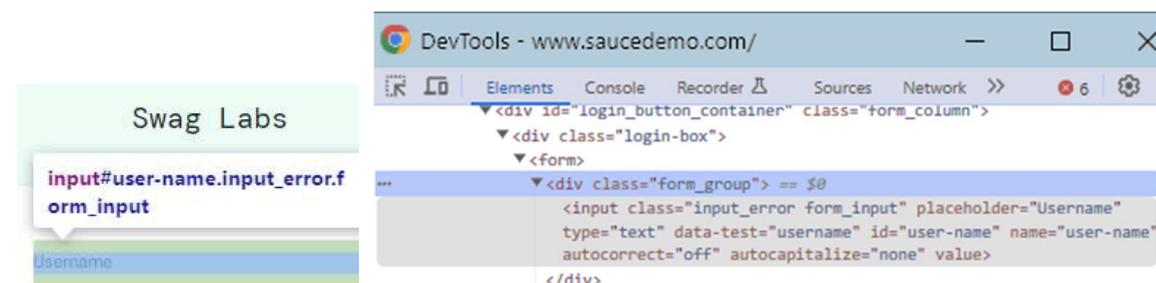
Swag Labs



2º Clicar com botão direito e depois em 'inspecionar'.



3º No DevTools, posicionar o mouse na parte indicada abaixo para localizar o elemento correspondente ao campo 'Username'. (**id="user-name"**)



Colocar o id para o usuário no arquivo 'Locators.cy.js'

```
//Preencher o campo usuário
cy.get('input#user-name').type('standard_user')
```

4º No **DevTools**, posicionar o mouse na parte indicada abaixo para localizar o elemento correspondente ao campo '**Password**'. (`placeholder="Password"`)

The screenshot shows the Saucedemo login page. A callout bubble points to the 'Password' input field. The DevTools Elements tab displays the corresponding HTML code:

```
<div class="form_group">
  <input class="input_error form_input" placeholder="Password"
    type="password" data-test="password" id="password" name="password"
    autocorrect="off" autocapitalize="none" value>
</div>
```

**Colocar o placeholder Password no arquivo 'Locators.cy.js'**

```
//Preencher o campo password
cy.get("input[placeholder='Password']").type('secret_sauce')
```

5º No **DevTools**, posicionar o mouse na parte indicada abaixo para localizar o elemento correspondente ao botão '**Login**'. (`class="submit-button.btn_action"; class="btn_action"`)

The screenshot shows the Saucedemo login page. A callout bubble points to the 'Login' button. The DevTools Elements tab displays the corresponding HTML code:

```
<input type="submit" class="submit-button btn_action" data-test="login-button" id="login-button" name="login-button" value="Login">
```

6º Verificar se está na página 'inventory.html'

No <https://www.saucedemo.com/v1/index.html>

```
//verificar se estou na página inventory.html
cy.get('title').contains('Swag Labs')
```

No <https://www.saucedemo.com/>

```
//verificar se estou na página inventory.html
cy.get('span.title').contains('Products')
```

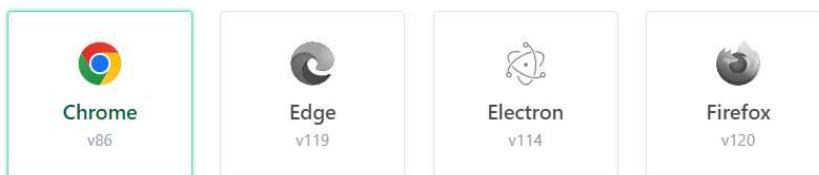
7º Abrir o terminal e executar

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\AULA-CYPRESS-T3> npx cypress open
```

8º Na interface gráfica do Cypress, escolher um browser

### Choose a browser

Choose your preferred browser for E2E testing.

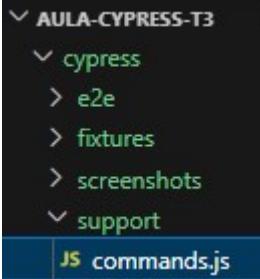


LOCATOR COM XPATH E INSTALAR O PLUGIN

`npm install -D cypress-xpath`

Incluir 'reference' para Xpath no final do arquivo commands.js

```
/// <reference types="cypress-xpath"/>
```



```

    ✓ AULA-CYPRESS-T3
      ✓ cypress
        > e2e
        > fixtures
        > screenshots
      ✓ support
        JS commands.js

```

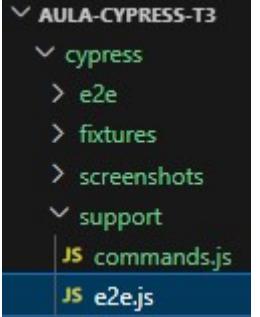
```

27  /// <reference types="Cypress"/>
28  /// <reference types="cypress-xpath"/>
29

```

Incluir 'require' no final do arquivo e2e.js

```
require ('cypress-xpath')
```



```

    ✓ AULA-CYPRESS-T3
      ✓ cypress
        > e2e
        > fixtures
        > screenshots
      ✓ support
        JS commands.js
        JS e2e.js

```

```

require ('cypress-xpath')

```

Copiar xpath



**BeforeEach()**

← → C 🔞 Não seguro | saucedemo.com/\_/#/specs/runner?file=cypress/e2e/Locators.cy.js

O Chrome está sendo controlado por um software de teste automatizado.

**Locators.cy.js**

```

describe('CSSLocators', () => {
    it('Login válido', () => {
        cy.visit('https://www.saucedemo.com/v1/index.html')
        cy.get('#user-name').type('standard_user')
        cy.get('#password').type('secret_sauce')
        cy.get('#login-button').click()
    })
})

it('Xpath', () => {
    //clicar no botão 'Add to cart'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').click()
    //verificar se o botão mudou para 'REMOVE'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').contains('REMOVE')
})
}
)

```

**TEST BODY**

```

1   get title
2   -contains Swag Labs

3   Xpath
4   BEFORE EACH
5   TEST BODY
6   -xpath //*[@id="inventory_container"]/div/div[1]/div[3]/button
7   -click
8   -xpath //*[@id="inventory_container"]/div/div[1]/div[3]/button
9   -contains REMOVE
10  (fetch) GET 200 /v1/img/peek.png
11  (fetch) GET 200 /v1/img/twitter.png

```

## Assertions: verificar/validar condições

### IMPLÍCITAS / BUILT-IN

Assertions padrão que já vem no Cypress. São built-in.

\*should, and

\*eq, contain, include, exist, have.length, have.value...

\*not

### MÚLTIPLAS ASSERTIONS IMPLÍCITAS PARA O MESMO ELEMENTO (should, and, and)

```
cy.url().should('include', 'orangehrmlive')
    .and('eq', 'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
    .and('contain', 'opensource')
    .and('not.contain', 'bug')
})
```

### ASSERTIONS IMPLÍCITAS: URL, TITLE, CONTAR LINKS, LOGO, VALOR DIGITADO

```
Assertion.cy.js
00:06

Assertion
  ✓ Assertions Implícitas
    TEST BODY
      visit https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
      1  url
      2
      3  - assert expected https://opensource-demo.orangehrmlive.com/web/index.php/auth/login to include
         orangehrmlive
      4  - assert expected https://opensource-demo.orangehrmlive.com/web/index.php/auth/login to equal
         **https://opensource-demo.orangehrmlive.com/web/index.php/auth/login**
      5  - assert expected https://opensource-demo.orangehrmlive.com/web/index.php/auth/login to include
         opensource
      6  - assert expected https://opensource-demo.orangehrmlive.com/web/index.php/auth/login to not include
         bug
      7  title
      8  - assert expected OrangeHRM to include Orange
      9  - assert expected OrangeHRM to equal **OrangeHRM**
     10  - assert expected OrangeHRM to include HRM
     11  - assert expected [ <a>, 4 more... ] to have a length of 5
        (xhr) GET 200 /web/index.php/core/i18n/messages
     12  - xpath //a
     13  get .orangehrm-login-logo > img
     14  - assert expected <img> to be visible
     15  - assert expected <img> to exist in the DOM
     16  get input[placeholder='Username']
     17  - type Admin
     18  get [placeholder='Username']
     19  - assert expected <input.oxd-input.oxd-input--focus> to have value Admin
```

## EXPLÍCITAS

Assertions que precisamos escrever explicitamente.

\***expect:** [BDD](#) (*Behavior Driven Development* / Desenvolvimento Dirigido pelo Comportamento)

\***assert:** [TDD](#) (*Test Driven Development* / Desenvolvimento Dirigido pelos Testes)

```
it['Assertions Explicitas', () => {
    //Entrar no site e fazer Login
    cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
    cy.get("input[placeholder='Username']").type('Admin')
    cy.get("input[placeholder='Password']").type('admin123')
    cy.xpath('//*[@id="app"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button').click()
    //Assertion Explicita
    let expName = "Alex Alex"
    cy.get('.oxd-userdropdown-name').then( (x) => {
        let actName = x.text()
        //BDD - Behavior Driven Development
        expect(actName).to.equal(expName)
        //TDD - Test Driven Development
        assert.equal(actName, expName)
    })
}]
```

```
get  input[placeholder='Username']
- type Admin
get  [placeholder='Username']
- assert expected <input.oxd-input.oxd-input--focus> to have value Admin
```

Diferença entre var e let

**LET:** [Local](#). Dentro do escopo.

**VAR:** [Global](#). Fora do escopo.

```
describe('Assertion', () => {

  it('Assertions Implícitas', () => {
    cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')

    //Should, and / útil para verificar token
    //  cy.url().should('include', 'orangehrmlive')
    //    .should('eq', 'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
    //    .should('contain', 'opensource')

    //ASSERTION NA URL
    cy.url().should('include', 'orangehrmlive')
    .and('eq', 'https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
    .and('contain', 'opensource')
    .and('not.contain', 'bug')

    //ASSERTION NO TITLE
    cy.title().should('include', 'Orange')
      .and('eq', 'OrangeHRM')
      .and('contain', 'HRM')

    //ASSERTION PARA CONTAR OS LINKS NA PÁGINA
    cy.xpath('/a').should('have.length', '5')

    //ASSERTION NO LOGO
    cy.get('.orangehrm-login-logo > img').should('be.visible')
    .and('exist')

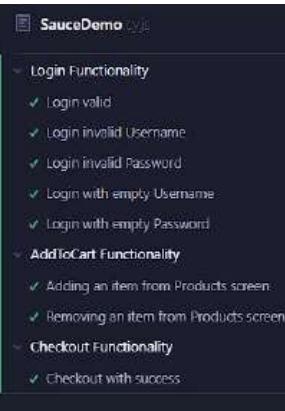
    //ASSERTION: VALOR DIGITADO CORRETAMENTE?
    cy.get("input[placeholder='Username']").type('Admin')
    cy.get("[placeholder='Username']").should('have.value', 'Admin')
  })

  it('Assertions Explícitas', () => {
    //Entrar no site e fazer Login
    cy.visit('https://opensource-demo.orangehrmlive.com/web/index.php/auth/login')
    cy.get("input[placeholder='Username']").type('Admin')
    cy.get("input[placeholder='Password']").type('admin123')
    cy.xpath('//*[@id="app"]/div[1]/div/div[1]/div/div[2]/div/form/div[3]/button').click()

    //Assertion Explícita
    let expName = "Alex Alex"
    cy.get('.oxd-userdropdown-name').then( (x) => {
      let actName = x.text()
      //BDD - Behavior Driven Development
      expect(actName).to.equal(expName)
      //TDD - Test Driven Development
      assert.equal(actName, expName)
    })
  })
})
```

## Exercícios: Refazer a aula e automatizar os testes do Sauce Demo

- Refazer a aula
  - Automatizar os testes do Sauce Demo:
- 1) Login:
    - a) Login com credenciais válidas
    - b) Login com username inválido
    - c) Login com senha inválida
    - d) Login com username vazio
    - e) Login com senha vazia
  - 2) AddToCart
    - a) Adicionar um item ao carrinho
    - b) Remover um item do carrinho
  - 3) Checkout
    - a) Checkout com sucesso



**Positivo:** Assertion para ver se está na página: URL (inventory), CONTAIN, EQ

**Negativo:** Mensagem de erro está visível? Texto está correto? Duas assertions para cada erro.

**Adicionar item ao carrinho:** se a bolinha com os itens apareceu e se o botão 'remove' apareceu.

**Remover um item do carrinho:** se a bolinha com os itens está visível ou não, se existe ou não. E se mudou de novo o botão para 'add to cart'.

**Checkout com sucesso:** uma assertion para validar a mensagem e verificar se está visível.

### 1 Login / describe

switch...case

```
let opcaoLogin = 2
function preencherCredenciais(){
  switch (opcaoLogin) {
    case 1: //Login com credenciais válidas
      //Preencher o campo usuário com dados válidos
      cy.get('input#user-name').type('standard_user')
      //Preencher o campo password com dados válidos
      cy.get("input[placeholder='Password']").type('secret_sauce')
      break;
    case 2: //Login com username inválido
      //Preencher o campo usuário com dados inválidos
      cy.get('input#user-name').type('standard_user123')
      //Preencher o campo password com dados válidos
      cy.get("input[placeholder='Password']").type("secret_sauce")
      break;
    case 3: //Login com senha inválida
      //Preencher o campo usuário com dados válidos
      cy.get('input#user-name').type('standard_user')
      //Preencher o campo password com dados inválidos
      cy.get("input[placeholder='Password']").type('secret_sauce123')
      break;
    case 4: //Login com username vazio
      //Preencher o campo password com dados válidos
      cy.get("input[placeholder='Password']").type('secret_sauce')
      break;
    case 5: //Login com senha vazia
      //Preencher o campo usuário com dados válidos
      cy.get('input#user-name').type('standard_user')
      break;
    default: //opção de Login inexistente
      break;
  }
}
```

beforeEach(() => { bloco })

```
beforeEach(() => {
  //Visitar a página
  cy.visit('https://www.saucedemo.com/v1/index.html')

  preencherCredenciais()

  //clicar no botão login
  cy.get('input.btn_action').click()
})
```

## a) LOGIN COM CREDENCIAIS VÁLIDAS / it

```
let opcaoLogin = 1
function preencherCredenciais(){
  switch (opcaoLogin) {
    case 1: //Login com credenciais válidas
      //Preencher o campo usuário com dados válidos
      cy.get('input#user-name').type('standard_user')
      //Preencher o campo password com dados válidos
      cy.get("input[placeholder='Password']").type('secret_sauce')
      break;
  }
}

describe('Login', () => {
  //opcaoLogin 1
  it('Login com credenciais válidas', () => {
    //Verificar se estou na página inventory.html
    cy.get('title').contains('Swag Labs')
    //Assertion na url
    cy.url().should('contain', 'saucedemo')
    .and('eq', 'https://www.saucedemo.com/v1/inventory.html')
  })
})
```

✓ Login com credenciais válidas

BEFORE EACH

- visit [\(fetch\)](https://www.saucedemo.com/v1/index.html) GET 200 /v1/img/Login\_Bot\_graphic.png
- get input#user-name
- type standard\_user
- (fetch) GET 200 /v1/img/SwagLabs\_logo.png
- get input[placeholder='Password']
- type secret\_sauce
- get input.btn\_action
- click
- (page load) --page loaded--
- (new url) [\(fetch\)](https://www.saucedemo.com/v1/inventory.html) GET 200 /v1/img/SwagBot\_Footer\_graphic.png
- (fetch) GET 200 /v1/img/peek.png
- (fetch) GET 200 /v1/img/twitter.png
- (fetch) GET 200 /v1/img/facebook.png
- (fetch) GET 200 /v1/img/linkedIn.png

TEST BODY

- get title
- contains Swag Labs
- url
- assert expected https://www.saucedemo.com/v1/inventory.html to include saucedemo
- assert expected https://www.saucedemo.com/v1/inventory.html to equal \*\*https://www.saucedemo.com/v1/inventory.html\*\*

## b) LOGIN COM USERNAME INVÁLIDO / it

```
let opcaoLogin = 2
function preencherCredenciais(){
  switch (opcaoLogin) {
    //opcaoLogin 2
    it('Login com username inválido', () => {
      //Verificar se username é inválido
      cy.get("[placeholder='Username']").should('have.value', 'standard_user123')
        .and('not.eq', 'standard_user123')
      //Verificar se mensagem de erro está visível
      cy.get('#login_button_container > div > form > h3').should('be.visible')
      //Verificar se texto da mensagem de erro está correto
      let expName = "Epic sadface: Username and password do not match any user in this service"
      cy.get('#login_button_container > div > form > h3').then( (x) => [
        let actName = x.text()
        //BDD - Behavior Driven Development
        expect(actName).to.equal(expName)
        //TDD - Test Driven Development
        // assert.equal(actName, expName)
      ])
    })
  }
}
```

```
✓ Login com username inválido
  ✓ BEFORE EACH
    1 visit https://www.saucedemo.com/v1/index.html
      (fetch) ⚡ GET 200 /v1/img/Login_Bot_graphic.png
    2 get input#user-name
    3 -type standard_user123
      (fetch) ⚡ GET 200 /v1/img/SwagLabs_logo.png
    4 get input[placeholder='Password']
    5 -type secret_sauce
    6 get input.btn_action
    7 -click

  ✓ TEST BODY
    1 get [placeholder='Username']
    2 -assert expected <input#user-name.form_input> to have value standard_user123
    3 -assert expected <input#user-name.form_input> to not equal standard_user123
    4 get #login_button_container > div > form > h3
    5 -assert expected <h3> to be visible
    6 get #login_button_container > div > form > h3
    7 -assert expected Epic sadface: Username and password do not match any user in this service to equal
      **Epic sadface: Username and password do not match any user in this service**
```

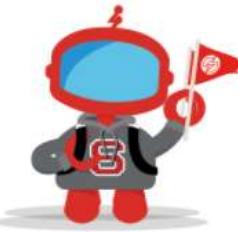


standard\_user123

\*\*\*\*\*

**LOGIN**

✖ Epic sadface: Username and password do not match any user in this service



## c) LOGIN COM SENHA INVÁLIDA / it

```
let opcaoLogin = 3
case 3: //Login com senha inválida
  //Preencher o campo usuário com dados válidos
  cy.get('input#user-name').type("standard_user")
  //Preencher o campo password com dados inválidos
  cy.get("input[placeholder='Password']").type('secret_sauce123')
  break;

//opcaoLogin 3
it('Login com senha inválida', () => {
  //Verificar se senha é inválida
  cy.get("[placeholder='Password']").should('have.value', 'secret_sauce123')
    .not('eq', 'secret_sauce123')
  //Verificar se mensagem de erro está visível
  cy.get('#login_button_container > div > form > h3').should('be.visible')
  //Verificar se texto da mensagem de erro está correto
  let expName = "Epic sadface: Username and password do not match any user in this service"
  cy.get('#login_button_container > div > form > h3').then( (x) => [
    let actName = x.text()
    //BDD - Behavior Driven Development
    expect(actName).to.equal(expName)
  ])
})
```

```
✓ Login com senha inválida
  ✓ BEFORE EACH
    1 visit https://www.saucedemo.com/v1/index.html
      (fetch) ⚡ GET 200 /v1/img/Login_Bot_graphic.png
    2 get input#user-name
    3 -type standard_user
      (fetch) ⚡ GET 200 /v1/img/SwagLabs_logo.png
    4 get input[placeholder='Password']
    5 -type secret_sauce123
    6 get input.btn_action
    7 -click

  ✓ TEST BODY
    1 get [placeholder='Password']
    2 -assert expected <input#password.form_input> to have value secret_sauce123
    3 not eq, secret_sauce123
    4 get #login_button_container > div > form > h3
    5 -assert expected <h3> to be visible
    6 get #login_button_container > div > form > h3
    7 -assert expected Epic sadface: Username and password do not match any user in this service to equal
      **Epic sadface: Username and password do not match any user in this service**
```



standard\_user

\*\*\*\*\*

**LOGIN**

✖ Epic sadface: Username and password do not match any user in this service



## d) LOGIN COM USERNAME VAZIO / it

```

let opcaoLogin = 4
case 4: //Login com username vazio
    //Preencher o campo password com dados válidos
    cy.get("input[placeholder='Password']").type('secret_sauce')
    break;

//opcaoLogin 4
it('Login com username vazio', () => [
    //Verificar se username está vazio
    cy.get("[placeholder='username']").should('have.length', 0)
    .and('not.exist')

    //Verificar se mensagem de erro está visível
    cy.get('#login_button_container > div > form > h3').should('be.visible')
    //Verificar se texto da mensagem de erro está correto
    let expName = "Epic sadface: Username is required"
    cy.get('#login_button_container > div > form > h3').then( (x) => {
        let actName = x.text()
        //TDD - Test Driven Development
        assert.equal(actName, expName)
    })
])

```

```

✓ Login com username vazio
  ✓ BEFORE EACH
    1 visit https://www.ssucedemo.com/v1/index.html
      (fetch) GET 200 /v1/img/Login_Bot_graphic.png
      (fetch) GET 200 /v1/img/SwagLabs_logo.png
    2 get input[placeholder='Password']
    3 -type secret_sauce
    4 get input.btn_action
    5 -click

  ✓ TEST BODY
    1 get [placeholder='username']
    2 -assert expected undefined to have a length of 0
    3 -assert expected undefined not to exist in the DOM
    4 get #login_button_container > div > form > h3
    5 -assert expected <h3> to be visible
    6 get #login_button_container > div > form > h3
    7 -assert expected Epic sadface: Username is required to equal **Epic sadface: Username is required**
```



## e) LOGIN COM SENHA VAZIA / it

```

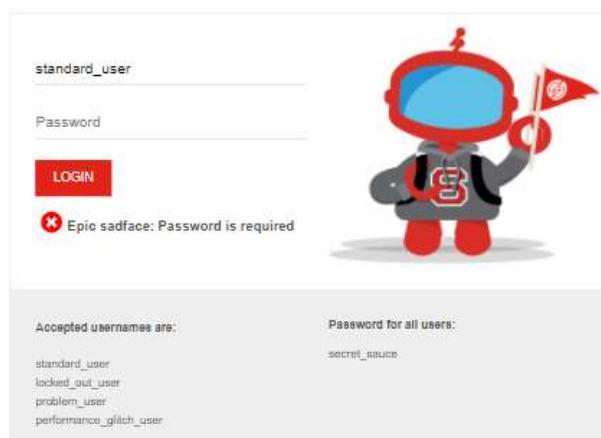
let opcaoLogin = 5
case 5: //Login com senha vazia
    //Preencher o campo usuário com dados válidos
    cy.get('input#user-name').type('standard_user')
    break;
//opcaoLogin 5
it('Login com senha vazia', () => {
    //Verificar se a senha está vazia
    cy.get("[placeholder='Password']").should('have.value', '')
    .and('be.empty')
    //Verificar se a mensagem de erro está visível
    cy.get('#login_button_container > div > form > h3').should('be.visible')
    //Verificar se o texto da mensagem de erro está correto
    let expName = "Epic sadface: Password is required"
    cy.get('#login_button_container > div > form > h3').then( (x) => {
        let actName = x.text()
        //TDD - Test Driven Development
        assert.equal(actName, expName)
    })
})

```

```

Login com senha vazia
✓ BEFORE EACH
  1 visit https://www.saucedemo.com/v1/index.html
    (fetch) GET 200 /v1/img/Login_Bot_graphic.png
    (fetch) GET 200 /v1/img/SwagLabs_logo.png
  2 get input#user-name
  3 -type standard_user
  4 get input.btn_action
  5 -click

TEST BODY
  1 get [placeholder='Password']
  2 -assert expected <input#password.form_input> to have value ''
  3 -assert expected <input#password.form_input> to be empty
  4 get #login_button_container > div > form > h3
  5 -assert expected <h3> to be visible
  6 get #login_button_container > div > form > h3
  7 -assert expected Epic sadface: Password is required to equal **Epic sadface: Password is required**
```



## 2 Add to cart / describe

## a) ADICIONAR UM ITEM AO CARRINHO / it

```

describe('Adicionar ao carrinho', () => {
  it('Adicionar um item ao carrinho', () => {
    //clicar no botão 'Add to cart'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').click()
    //verificar se o botão mudou para 'REMOVE'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').contains('REMOVE')
    //verificar se a bolinha com quantidade de itens está visível e se existe no carrinho
    cy.get('#shopping_cart_container > a > span').should('be.visible')
    .and('exist')
  })
})

```

```

Adicionar ao carrinho
✓ Adicionar um item ao carrinho
✓ BEFORE EACH
  1 visit https://www.saucedemo.com/v1/index.html
  2 get input#user-name
  3 -type standard_user
  4 get input[placeholder='Password']
  5 -type secret_sauce
  6 get input.btn_action
  7 -click
  (page load) --page loaded--
  (new url) https://www.saucedemo.com/v1/inventory.html

TEST BODY
  1 -xpath //*[@id="inventory_container"]/div/div[1]/div[3]/button
  2 -click
  3 -xpath //*[@id="inventory_container"]/div/div[1]/div[3]/button
  4 -contains REMOVE
  5 get #shopping_cart_container > a > span
  6 -assert expected <span.fa-layers-counter.shopping_cart_badge> to be visible
  7 -assert expected <span.fa-layers-counter.shopping_cart_badge> to exist in the DOM
```



## b) REMOVER UM ITEM DO CARRINHO / it

```

describe('Adicionar ao carrinho', () => {
  >   it('Adicionar um item ao carrinho', () => { ...
  >     });

  it('Remover um item do carrinho', () => {
    //clicar no botão 'ADD TO CART'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').click()
    //clicar no botão 'REMOVE'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').click()
    //verificar se o botão mudou para 'ADD TO CART'
    cy.xpath('//*[@id="inventory_container"]/div/div[1]/div[3]/button').contains('ADD TO CART')
    //verificar se a bolinha com quantidade de itens NÃO existe no carrinho
    cy.get('#shopping_cart_container > a > span').should('not.exist') //‘not.be.visible’: dá erro
  >   });
}

  ✓ Remover um item do carrinho
> BEFORE EACH
  ✓ TEST BODY
  1 - xpath //*[@id="inventory_container"]//div/div[1]/div[3]/button
  2 - click
  3 - xpath //*[@id="inventory_container"]//div/div[1]/div[3]/button
  4 - click
  5 - xpath //*[@id="inventory_container"]//div/div[1]/div[3]/button
  6 - contains ADD TO CART
  7 get #shopping_cart_container > a > span
  8 - assert expected #shopping_cart_container > a > span not to exist in the DOM
    (Fetch) GET 200 /v1/img/peek.png
    (Fetch) GET 200 /v1/img/twitter.png
    (Fetch) GET 200 /v1/img/facebook.png
    (Fetch) GET 200 /v1/img/linkedin.png

```



### 3 Checkout / describe

#### a) CHECKOUT COM SUCESSO / it

```

describe('Checkout', () => [
  it('Checkout com sucesso', () => {
    //Clicar no carrinho
    cy.xpath('//*[@@id="shopping_cart_container"]/a').click()
    //Verificar se estou na página 'cart.html (Your Cart)'
    cy.xpath('//*[@id="contents_wrapper"]/div[2]').contains('Your Cart')
    //Clicar no botão CHECKOUT
    cy.xpath('//*[@id="cart_contents_container"]/div/div[2]/a[2]').click()
    //Verificar se estou na página 'checkout-step-one.html (Checkout: Your Information)'
    cy.xpath('//*[@id="contents_wrapper"]/div[2]').contains('Checkout: Your Information')
    //Preencher campo 'First Name'
    cy.get('#first-name').type('Carmen')
    //Verificar se First Name foi digitado corretamente
    cy.get('[placeholder="First Name"]').should('have.value', 'Carmen')
    //Preencher campo 'Last Name'
    cy.get('#last-name').type('Cabral')
    //Verificar se Last Name foi digitado corretamente
    cy.get('[placeholder="Last Name"]').should('have.value', 'Cabral')
    //Preencher campo 'Zip/Postal Code'
    cy.get('#postal-code').type('4444-444')
    //Verificar se Zip/Code foi digitado corretamente
    cy.get('[placeholder="Zip/Postal Code"]').should('have.value', '4444-444')
    //Clicar no botão 'Continue'
    cy.xpath('//*[@id="checkout_info_container"]/div/form/div[2]/input').click()
    //Verificar se estou na página 'checkout-step-two.html (Checkout: Overview)'
    cy.xpath('//*[@id="contents_wrapper"]/div[2]').contains('Checkout: Overview')
    //Clicar no botão 'Finish'
    cy.xpath('//*[@id="checkout_summary_container"]/div/div[2]/div[8]/a[2]').click()
    //Verificar se estou na página 'checkout-complete.html (Finish)'
    cy.xpath('//*[@id="contents_wrapper"]/div[2]').contains('Finish')
    //mensagem de checkout: THANK YOU FOR YOUR ORDER
    //Verificar se mensagem de checkout está visivel
    cy.xpath('//*[@id="checkout_complete_container"]/h2').should('be.visible')
    //Verificar se texto da mensagem de checkout está correto
    let expName = "THANK YOU FOR YOUR ORDER"
    cy.xpath('//*[@id="checkout_complete_container"]/h2').then( (x) => {
      let actName = x.text()
      //TDD - Test Driven Development
      assert.equal(actName, expName)
    })
    //Verificar se a imagem do checkout está visivel e se existe
    cy.get('#checkout_complete_container > img').should('be.visible')
    .and('exist')
  })
])

```

✓ Checkout com sucesso

> BEFORE EACH

< TEST BODY

- xpath //\*[@id="shopping\_cart\_container"]/a
- click
- (page load) --page loaded--
- (new url) https://www.saucedemo.com/v1/cart.html
- xpath //\*[@id="contents\_wrapper"]/div[2]
- contains Your Cart
- xpath //\*[@id="cart\_contents\_container"]/div/div[2]/a[2]
- click
- (page load) --page loaded--
- (new url) https://www.saucedemo.com/v1/checkout-step-one.html
- xpath //\*[@id="contents\_wrapper"]/div[2]
- contains Checkout: Your Information
- get #first-name
- type Carmen
- get [placeholder="First Name"]
- assert expected <input#first-name.form\_input> to have value Carmen
- get #last-name
- type Cabral
- get [placeholder="Last Name"]
- assert expected <input#last-name.form\_input> to have value Cabral
- get #postal-code
- type 4444-444
- get [placeholder="Zip/Postal Code"]
- assert expected <input#postal-code.form\_input> to have value 4444-444
- xpath //\*[@id="checkout\_info\_container"]/div/form/div[2]/input
- click
- (page load) --page loaded--
- (new url) https://www.saucedemo.com/v1/checkout-step-two.html
- xpath //\*[@id="contents\_wrapper"]/div[2]
- contains Checkout: Overview
- xpath //\*[@id="checkout\_summary\_container"]/div/div[2]/div[8]/a[2]
- click
- (page load) --page loaded--
- (new url) https://www.saucedemo.com/v1/checkout-complete.html
- xpath //\*[@id="contents\_wrapper"]/div[2]
- contains Finish
- assert expected <h2.complete-header> to be visible
- xpath //\*[@id="checkout\_complete\_container"]/h2
- xpath //\*[@id="checkout\_complete\_container"]/h2
- assert expected THANK YOU FOR YOUR ORDER to equal \*\*THANK YOU FOR YOUR ORDER\*\*
- get #checkout\_complete\_container > img
- assert expected <img.pony\_express> to be visible
- assert expected <img.pony\_express> to exist in the DOM

