Digital Business Company
Tech Up Together

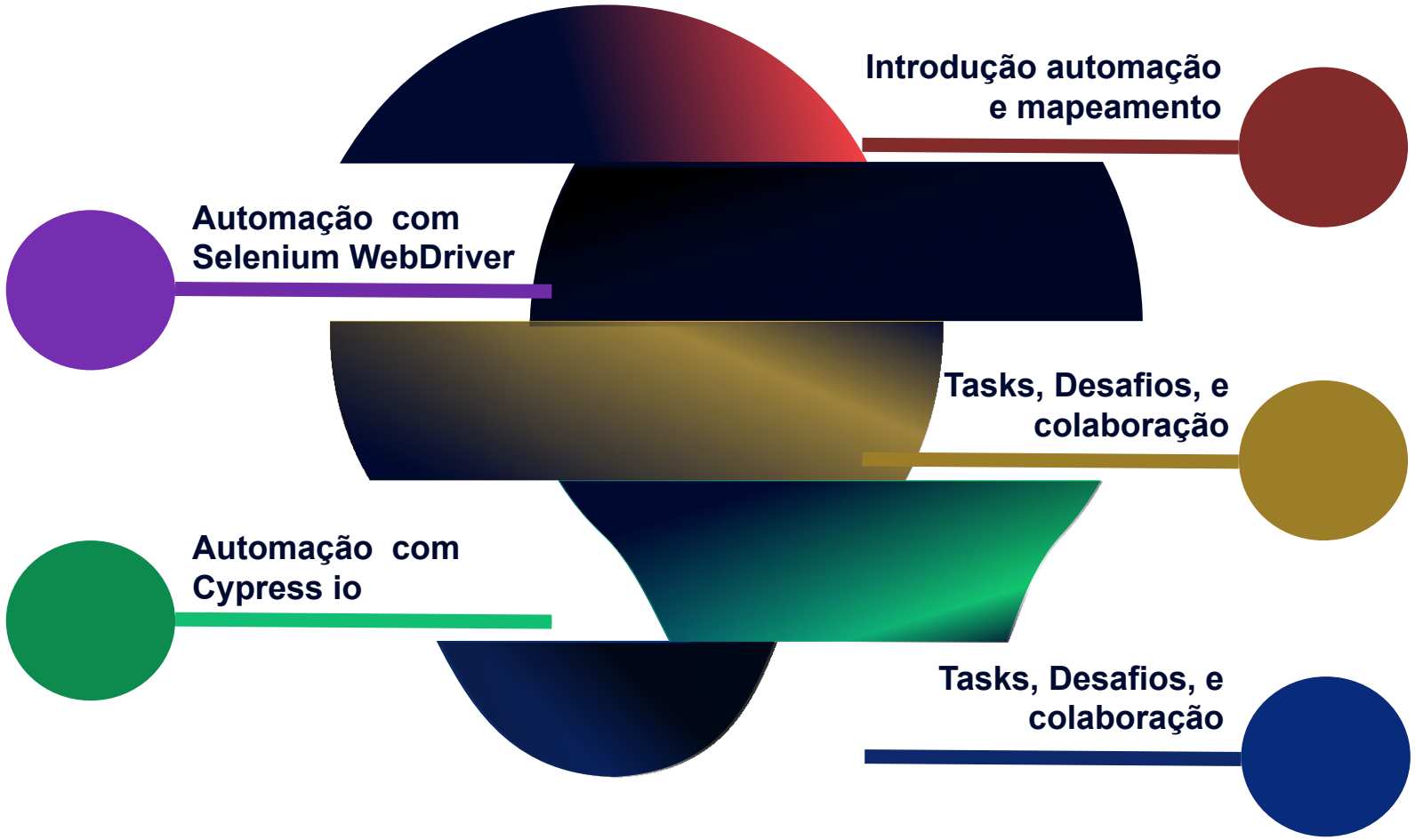
dbccompany.com.br

Tecnologia de Automação Front End

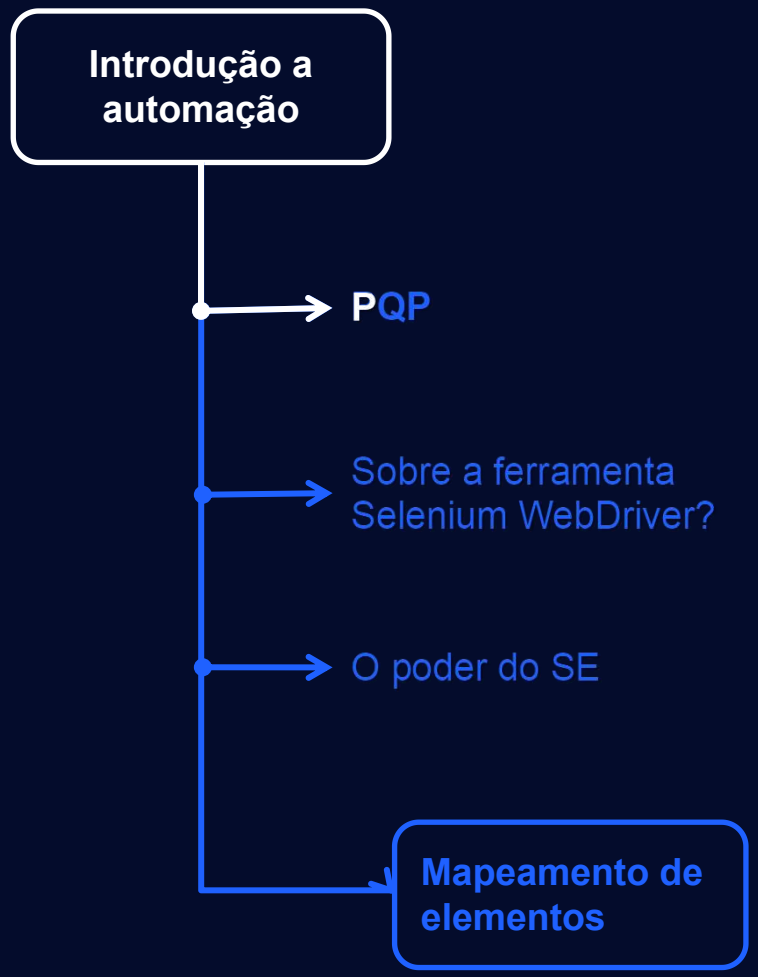
Tecnologias de Automação

FRONT END

Tecnologias de Automação Front End

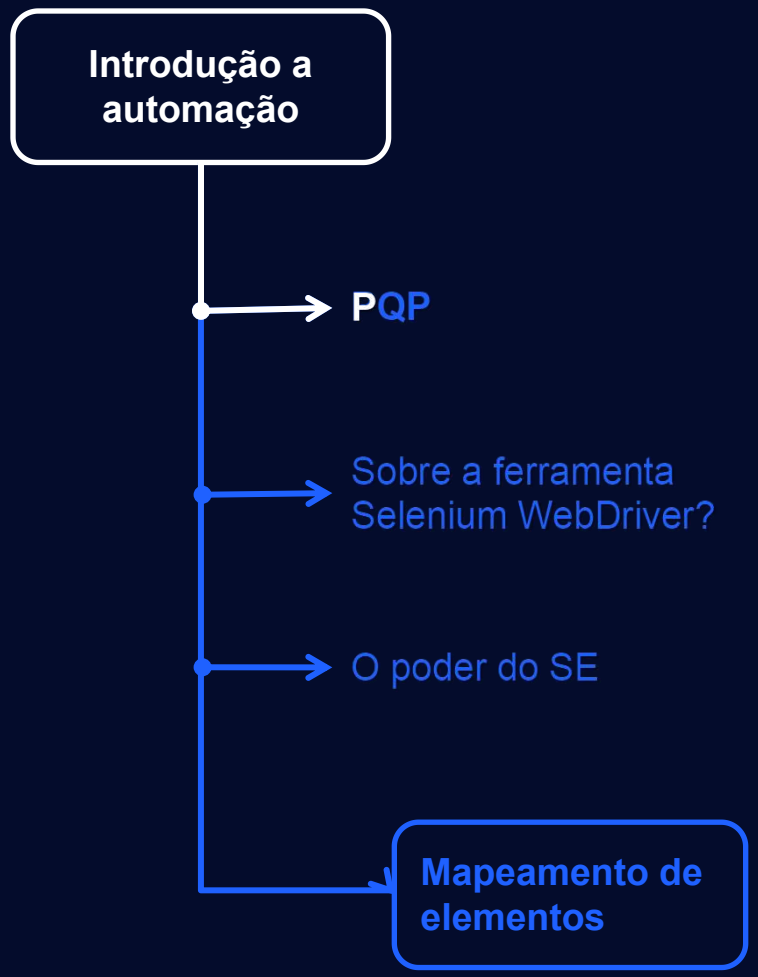


Tecnologias de Automação Front End



- Porque automatizar testes ?

Tecnologias de Automação Front End

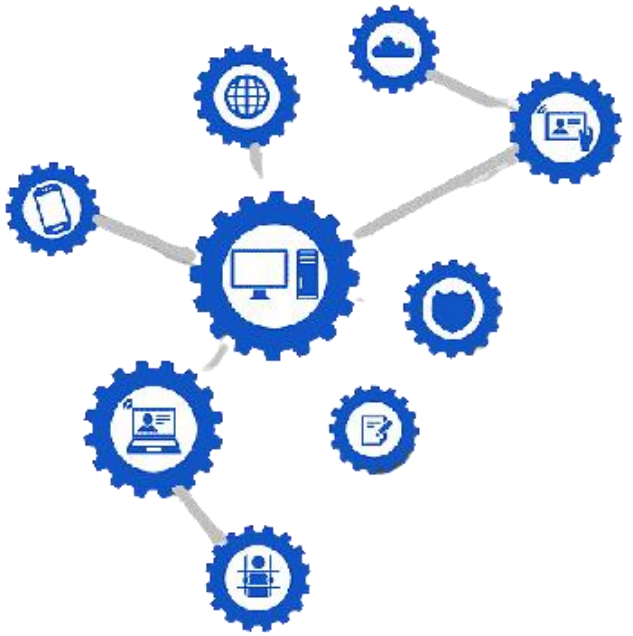


- Porque automatizar testes ?

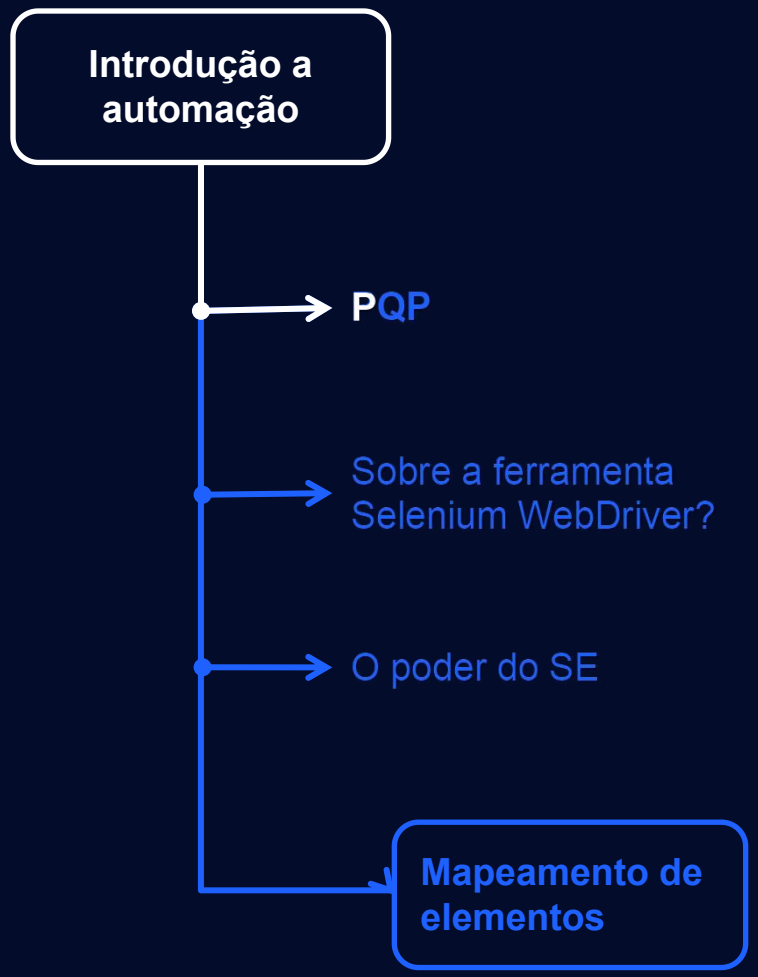
Imagine um sistema complexo de uma grande empresa que foi desenvolvido a mais de 15 anos, e hoje tem milhares de usuários em seu banco de dados rodando em produção.

- Historico de qualidade do sistema:

  - 120 funcionalidades testadas;
  - 600 Cenários de front
  - 1000 Cenários de api



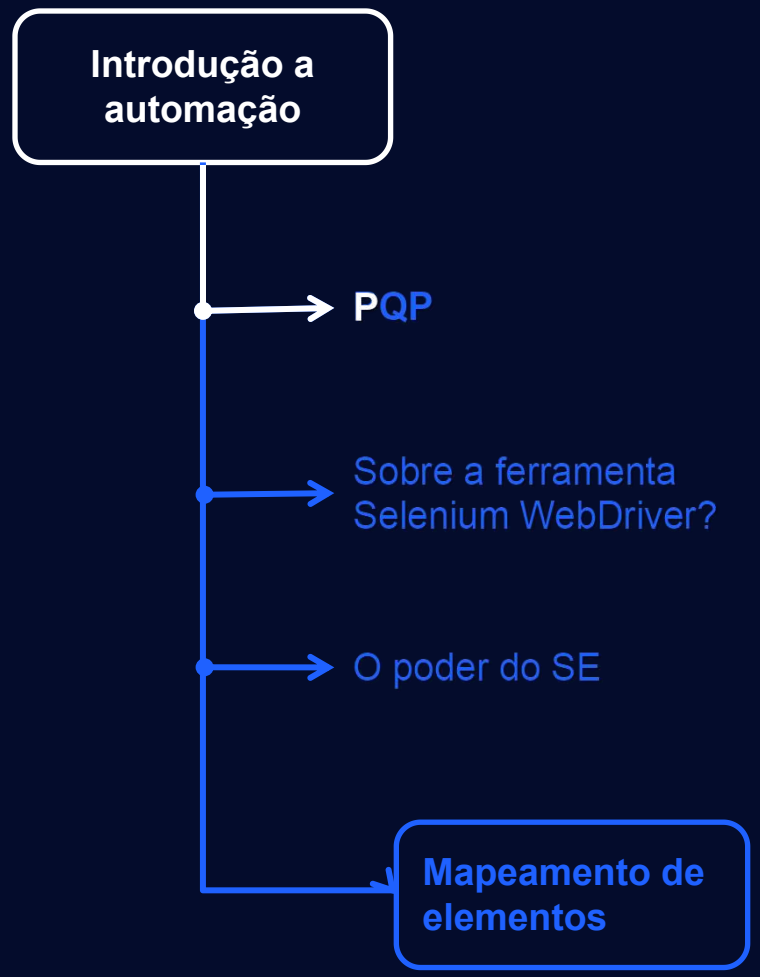
Tecnologias de Automação Front End



- Porque automatizar testes ?
  - Sabendo dessas informações quantas pessoas de qualidade precisariamos para fazer um retestes manualmente quando ouvesse um bug ou uma alteração no sistema ?

  - Quantos dias (tempo) levaria para garantir que todos os cenários foram testados é as funcionalidades estão de fato "ok" ?

Tecnologias de Automação Front End



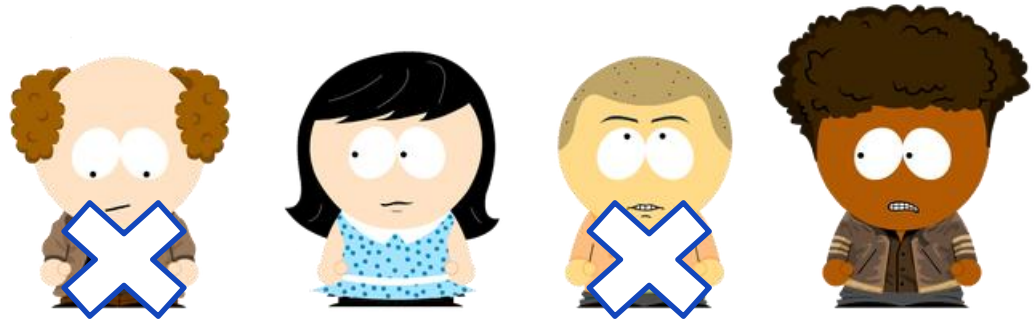
- QAs:

- Depende de alguns fatores:
  - Impacto de alteração da funcionalidade
  - Criticidade do Bug
  - Complexidade do cenário
  - Complexidade da massa de dados
  - Do prazo e planejamento investido na correção

Tecnologias de Automação Front End

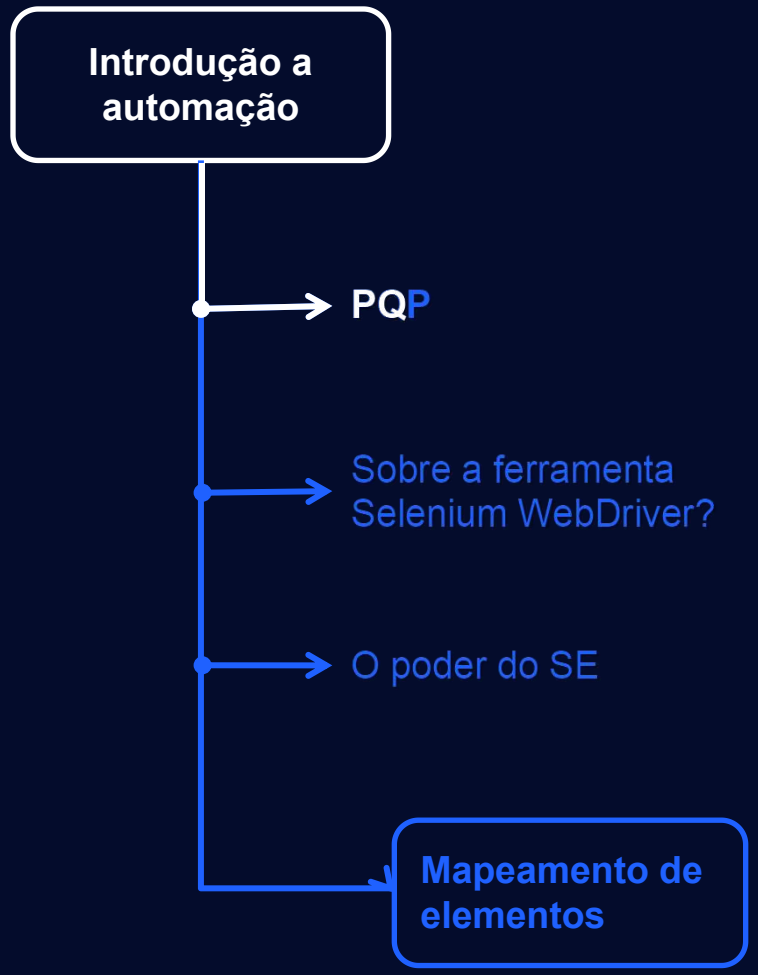
- Introdução a automação
  - PQP
  - Sobre a ferramenta Selenium WebDriver?
  - O poder do SE
  - Mapeamento de elementos

Beneficios de automatizar testes.



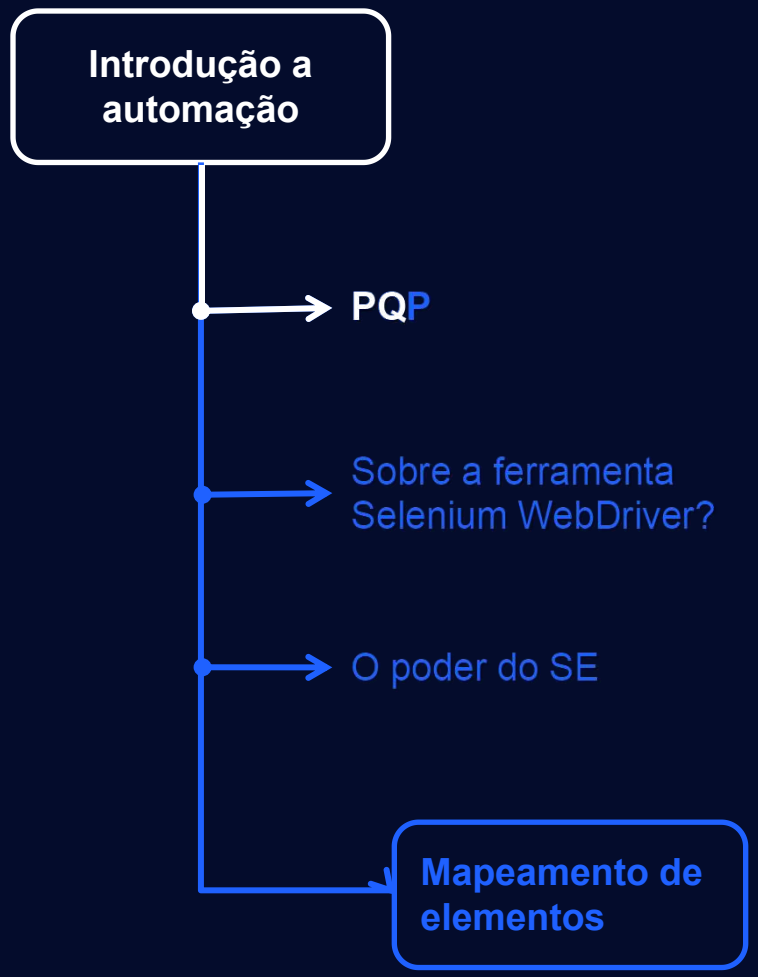
- Testes
  - Mais rapidos
  - Regressivos
  - Planejados
- Verifica a qualidade da funcinalidade após uma alteração do software
- Garante uma maior segurança em respostas rapidas

Tecnologias de Automação Front End



- Quando automatizar ?

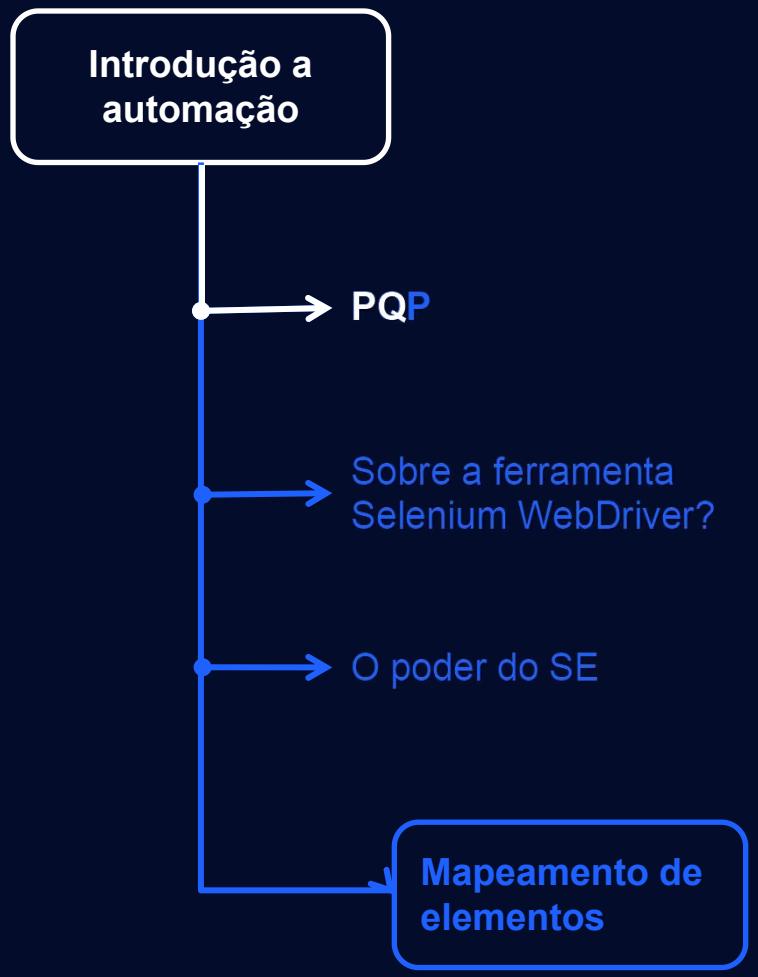
Tecnologias de Automação Front End



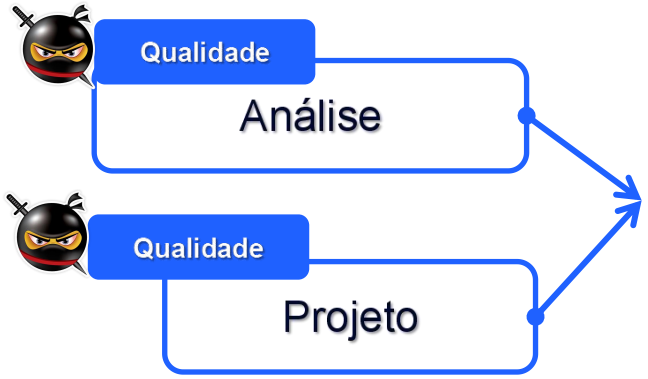
- Inicio de projeto

- Em qual etapa do ciclo devo começar a minha automação de cenário?

Tecnologias de Automação Front End



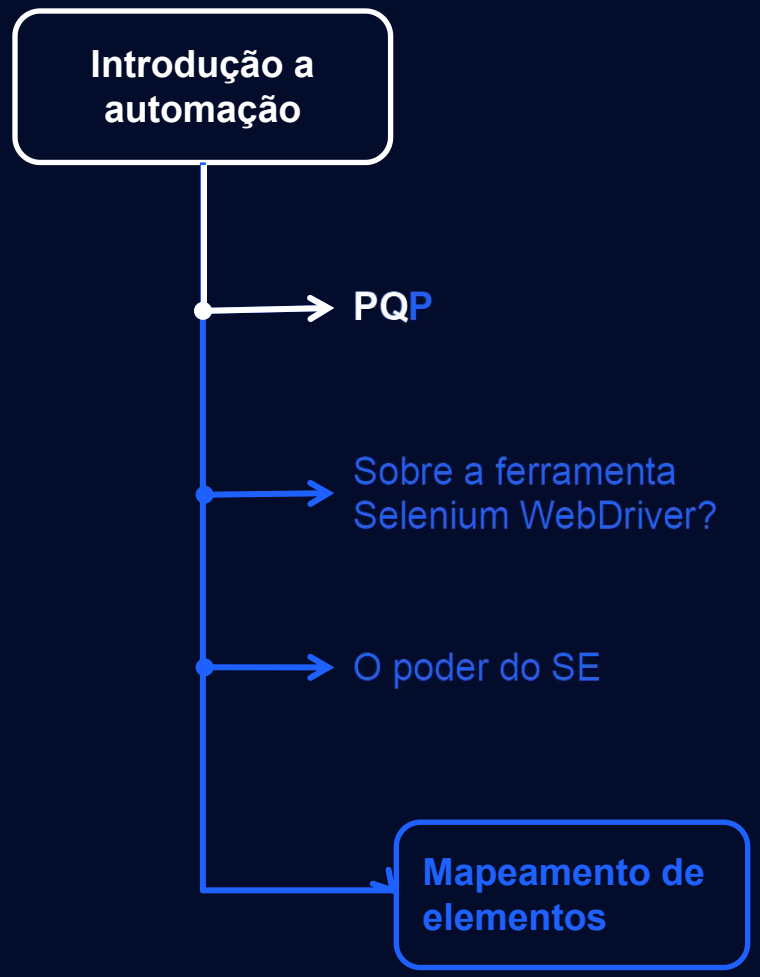
- Inicio de projeto



- Etapa que é desenvolvido as User Storys, RG e CA, (em conjunto com o QA)
- Cria os cenários de testes baseado no entendimento, das regras e dos criterios de aceite.

- Desenvolvedores começam a codificar as funcionalidades
- QAs começam codificar os cenários para adiantar o processo de automação.

Tecnologias de Automação Front End



- ## Quando automatizar ?

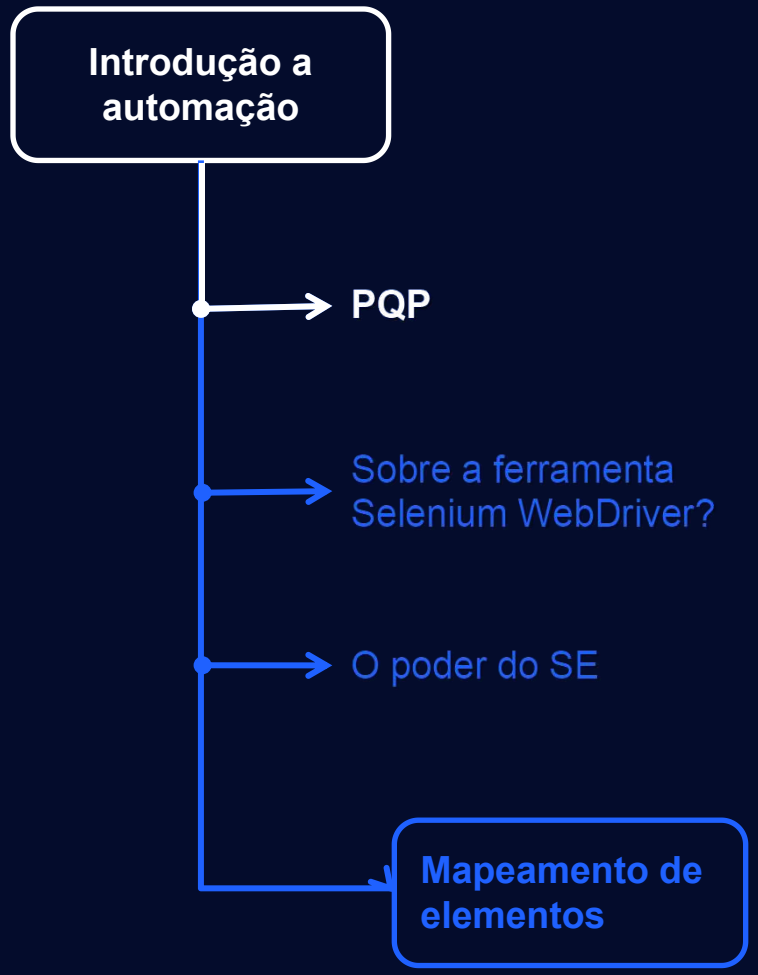
Logo após que as funcionalidades forem priorizadas na sprints, e as mesmas estejam com seus CA e RG bem definidas.

É que o QA tenha feito seu planejamento de cenário.

Priorizando sempre os fluxos principais

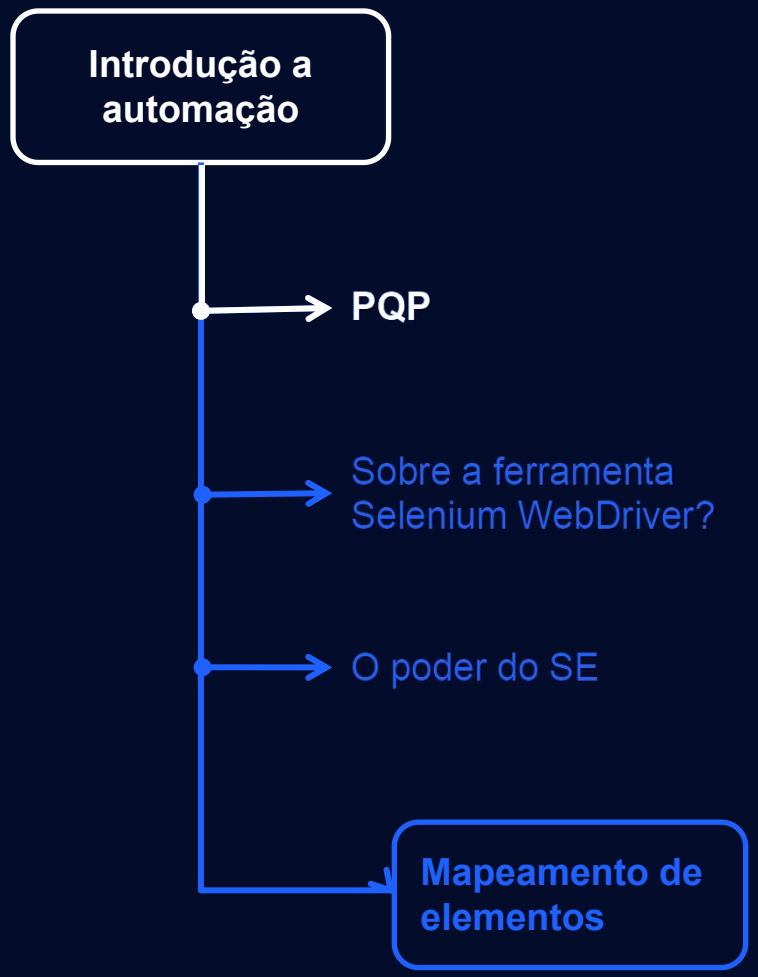
Depois os fluxos alternativos

Tecnologias de Automação Front End



- Por onde começar ?

Tecnologias de Automação Front End



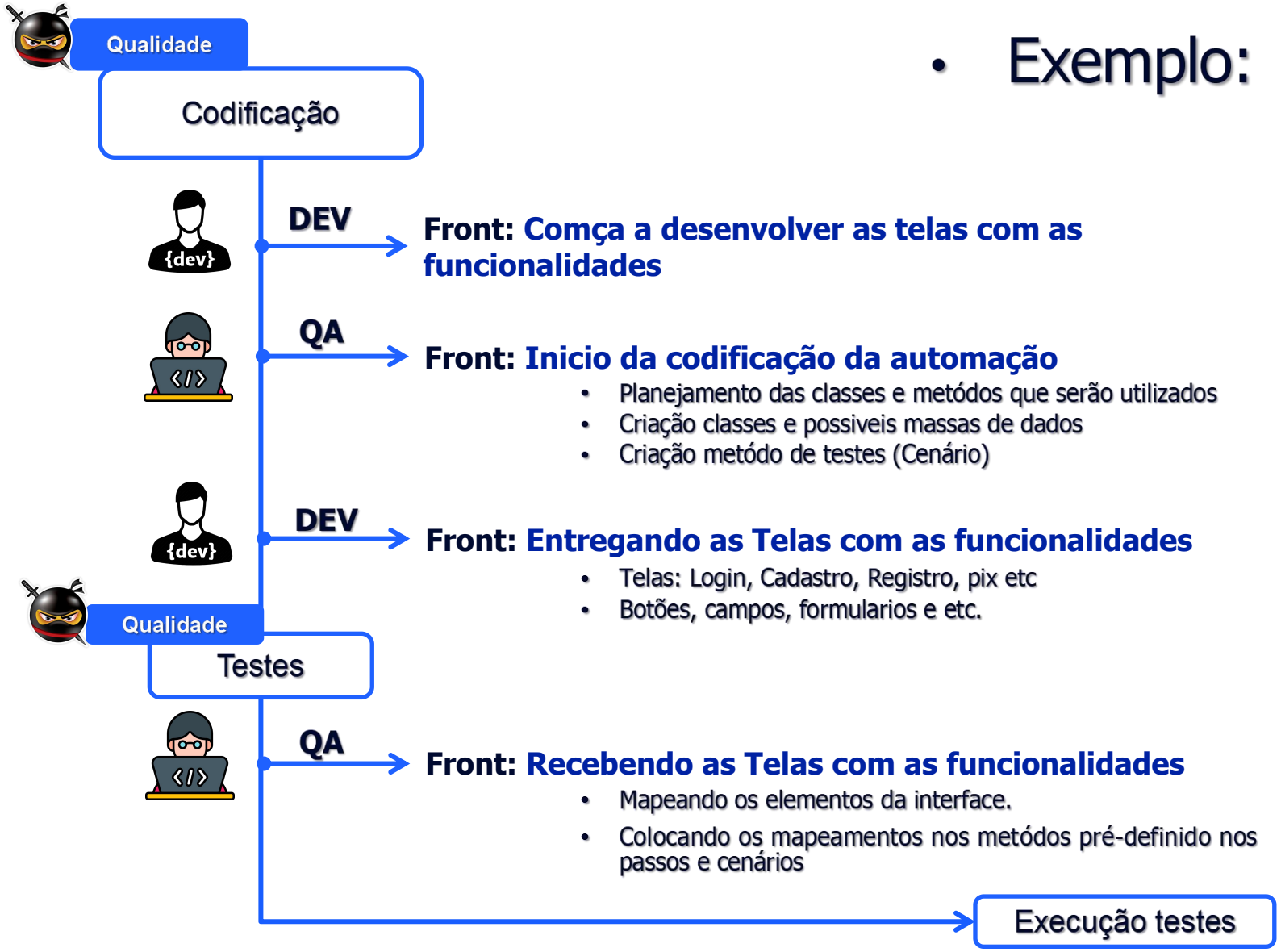
- Por onde começar ?

1. Identificação dos fluxos prinpal (is);
2. Identificação dos fluxos alternativos.
3. Sempre adiantando a automação o maximo possivel

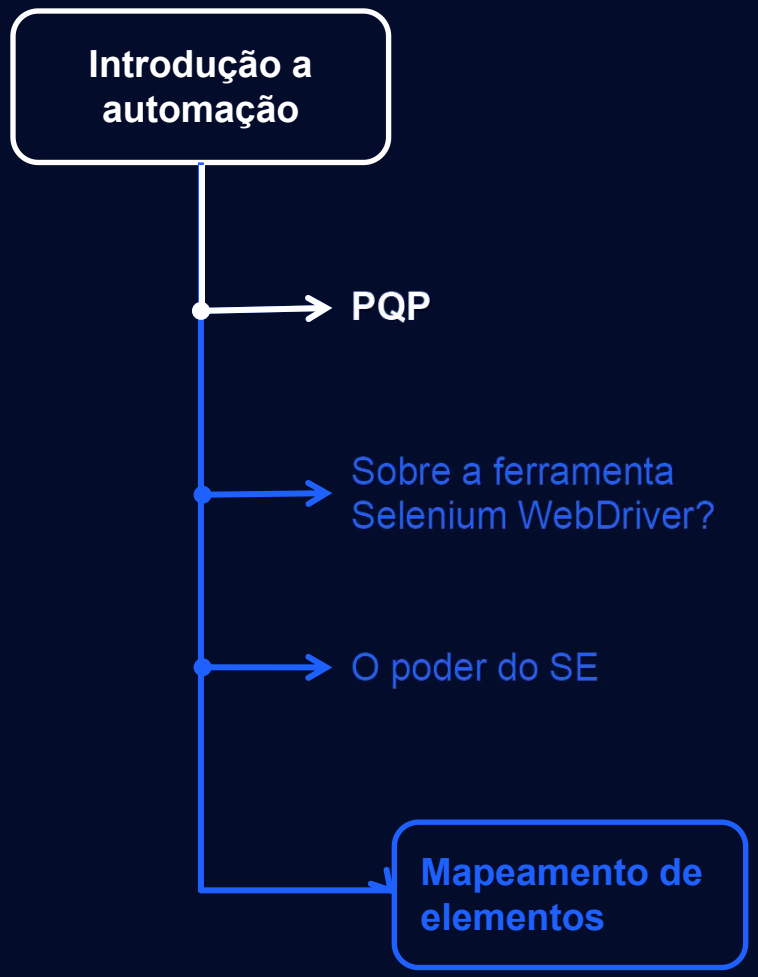
Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- **Mapeamento de elementos**

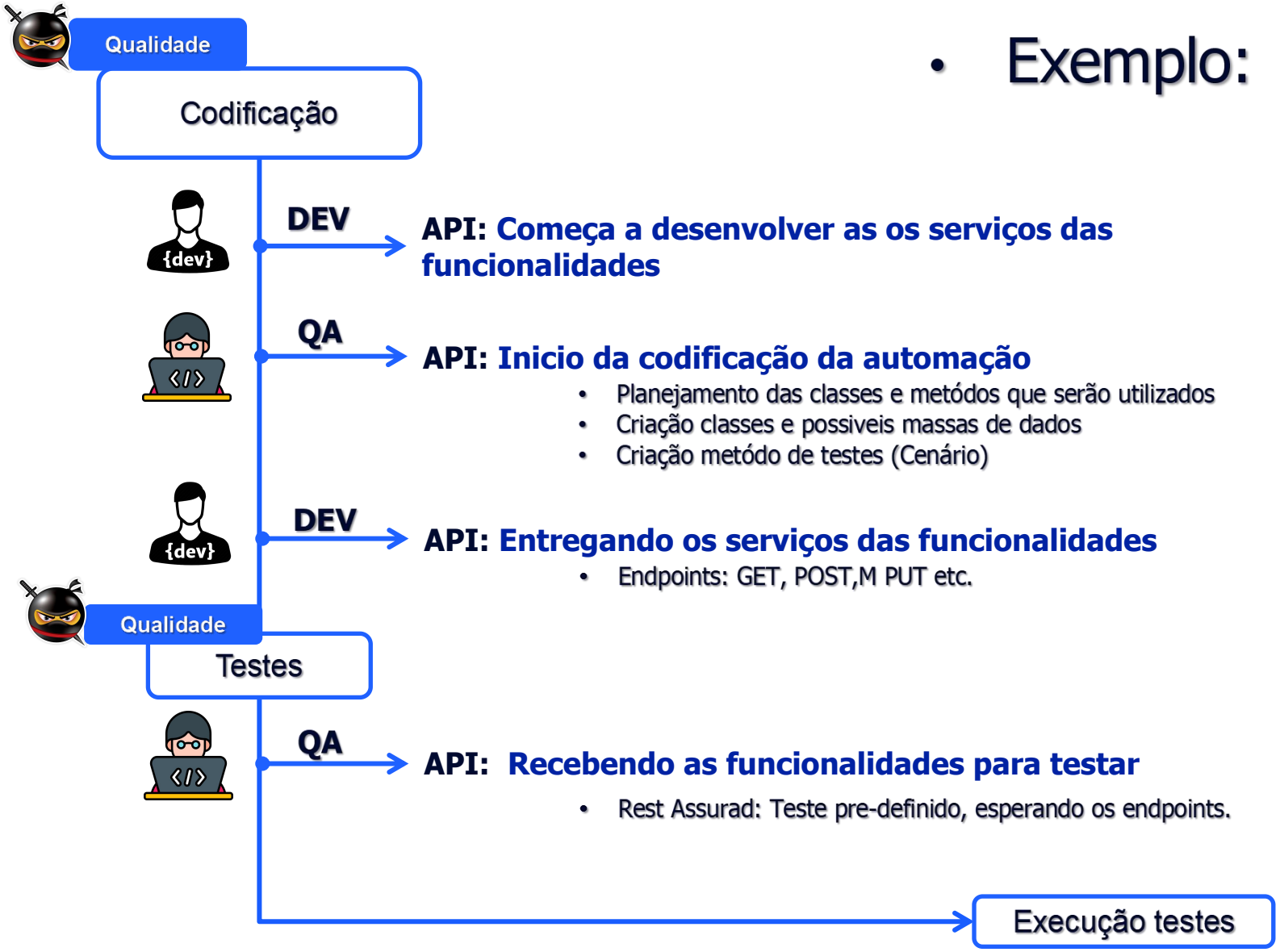
- Exemplo:



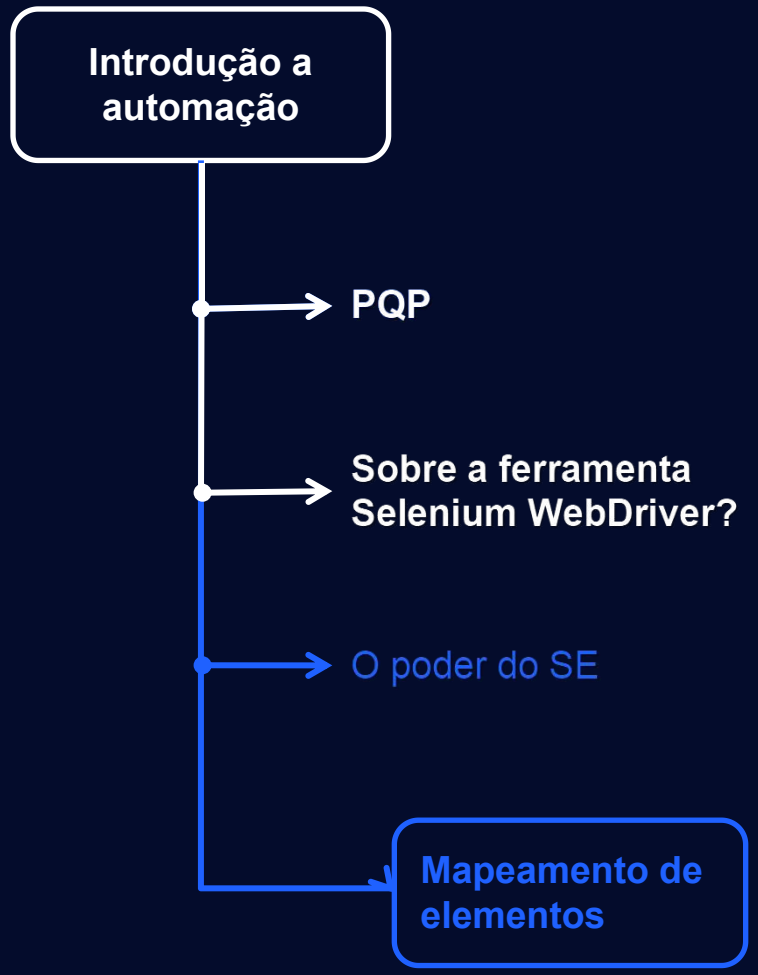
Tecnologias de Automação Front End



- Exemplo:



Tecnologias de Automação Front End



- Sobre a ferramenta
- Selenium WebDriver

Tecnologias de Automação Front End

- Introdução a automação
  - PQP
  - Sobre a ferramenta Selenium WebDriver?
  - O poder do SE
  - Mapeamento de elementos

Criada pela

Empresa TW em 2004

Novs dias atuais

É considerada a ferramenta essencial pros QAs e pedidas nas vagas de Qualidade

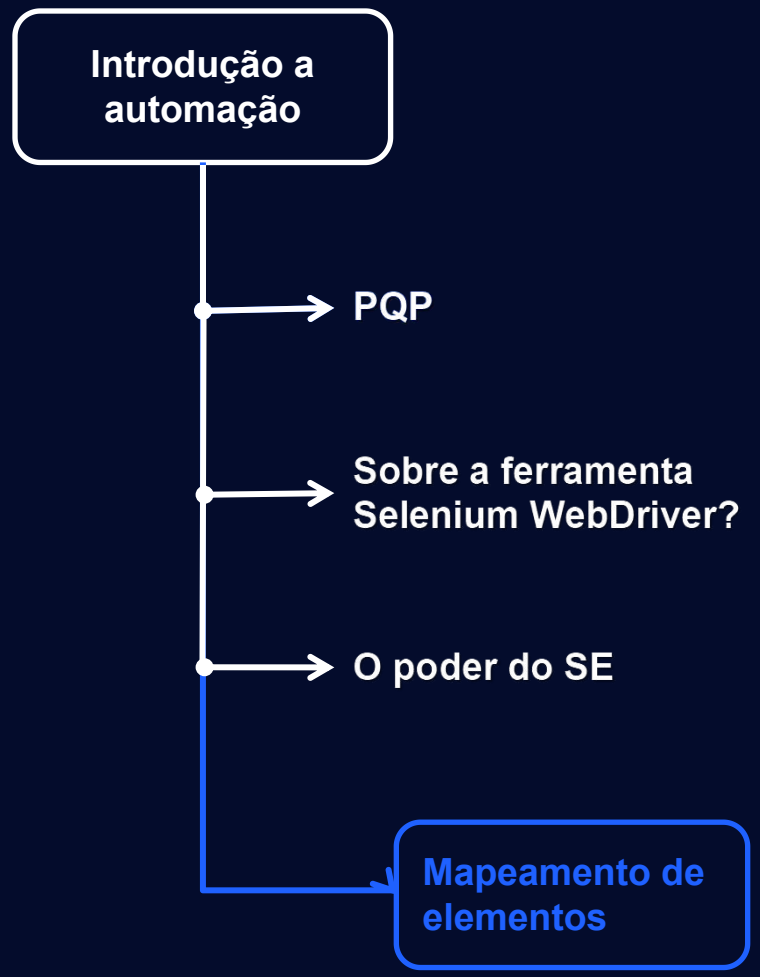
Para que serve ?

Testes de front-end (Interface grafica)

Pode ser utilizados em diversas linguagem de Programação tais como

- Java
- Python
- C#
- Ruby
- Tiutieby maleketo baiou, lactui peidi baiou.

Tecnologias de Automação Front End



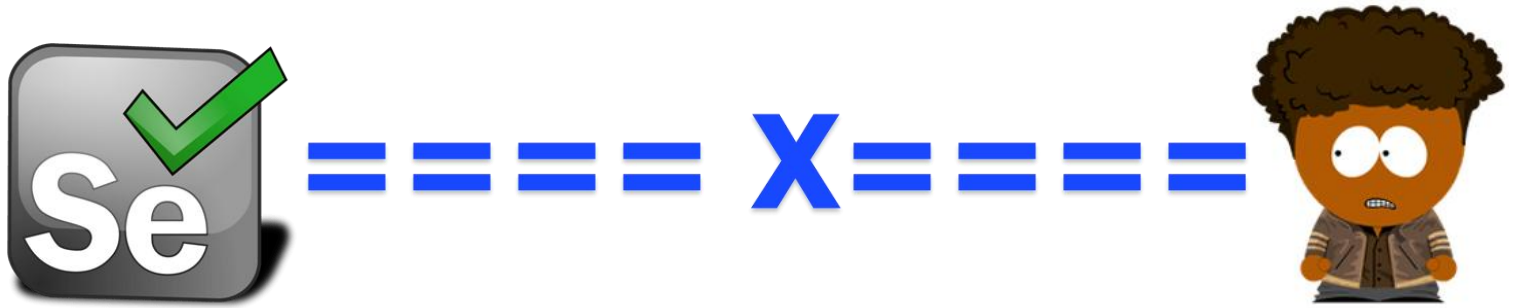
O poder do SE

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

Selenium testando

- Usando métodos de interações com o sistema
  Ex: Clicar, Preencher campo, Esperar, Abrir navegador
- É preciso mapear cada elemento do sistema Front-End
  Ex: Botão, campo, textos



Uma pessoa testando

- Abrir o navegador
- Precisa entrar e visualizar o sistema
- Usar o teclado para digitar e preecher campos
- Usar mouse para clicar em campos ou botões

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

Selenium interagindo

- Métodos de interações:

Ex:

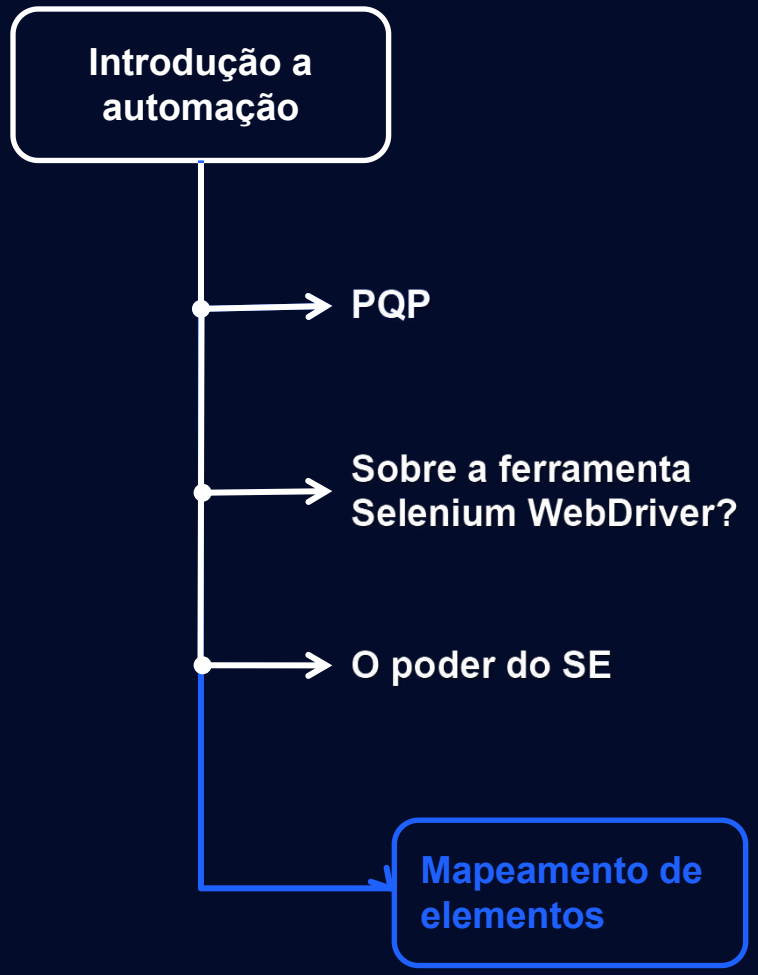
- Abrir navegador:
  .get("www.google.com");

- Clicar:
  .click();

- Preencher campo,
  .sendKeys("Escrevendo um texto");

- Esperar.
  .wait();

Tecnologias de Automação Front End



Conhecendo os principais métodos do SE

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.get()

Este comando tem como objetivo abrir o browser e acessar o endereço do sistema.

Exemplo:

```
driver.get("https://www.google.com");
```

Comando para acessar a url

Endereço do sistema

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.findElement()

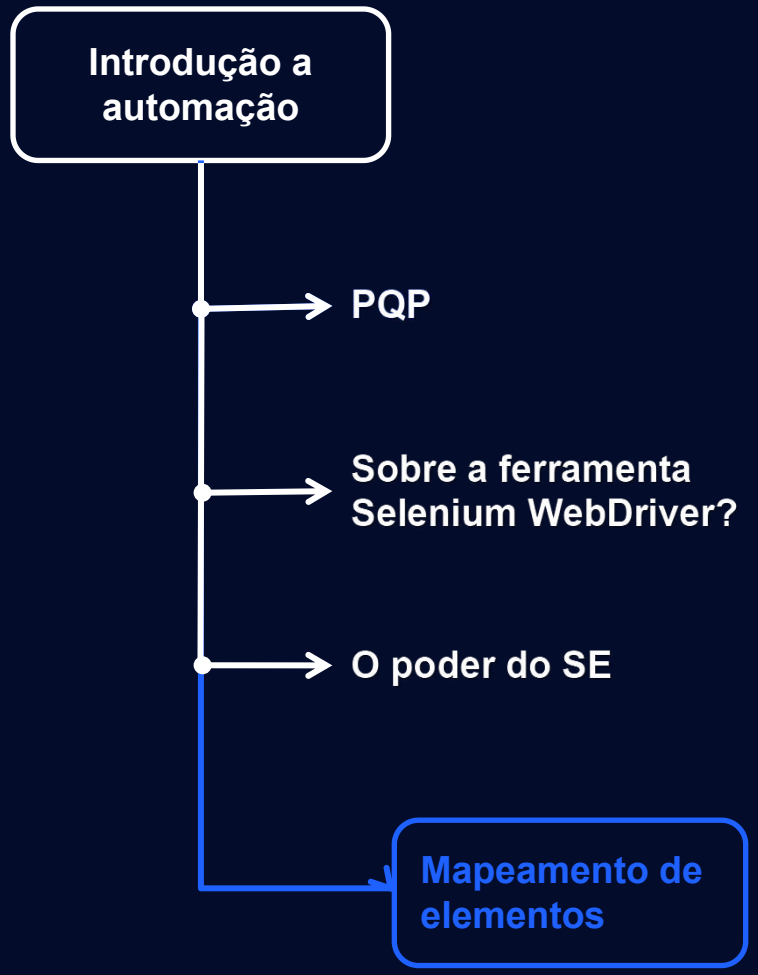
Comando para encontrar o elemento.

Exemplo:

```
driver.findElement(cssSelector("Mapeamento"));
```

- Procurar elemento
- Tipo do Seletor
- Mapeamento do elemento

Tecnologias de Automação Front End



.findElement()

Comando para encontrar o elemento.

Exemplo:

```
driver.findElement(
                    By.
                      cssSelector("Mapeamento")
                      id("Mapeamento")
                      className("Mapeamento")
                      xpath("Mapeamento")
                      name("Mapeamento")
                      linkText("Mapeamento")
                                              );
```

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.click()

Comando para clicar em elemento botão.

Exemplo:

```
driver.findElement(cssSelector("Mapeamento")).click();
```

- Procurar elemento
- Tipo do Seletor
- Mapeamento do elemento
- Metódo de clicar

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.getText()

Comando para ler um texto na tela

Exemplo:

```
driver.findElement(cssSelector("Mapeamento")).getText();
```

Lêr

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.sendKeys("Texto")

Comando para escrever em elemento do tipo campo

Exemplo:

```
driver.findElement(
      cssSelector("Mapeamento")).sendkeys("Texto que deseja escrever");
```

Procurar elemento

Metódo de escrever

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.clear()

Comando limpar campo

Exemplo:

```
driver.findElement(cssSelector("Mapeamento")).clear();
```

Procurar elemento

Limpa campo

Tecnologias de Automação Front End

- Introdução a automação
- PQP
- Sobre a ferramenta Selenium WebDriver?
- O poder do SE
- Mapeamento de elementos

.quit()

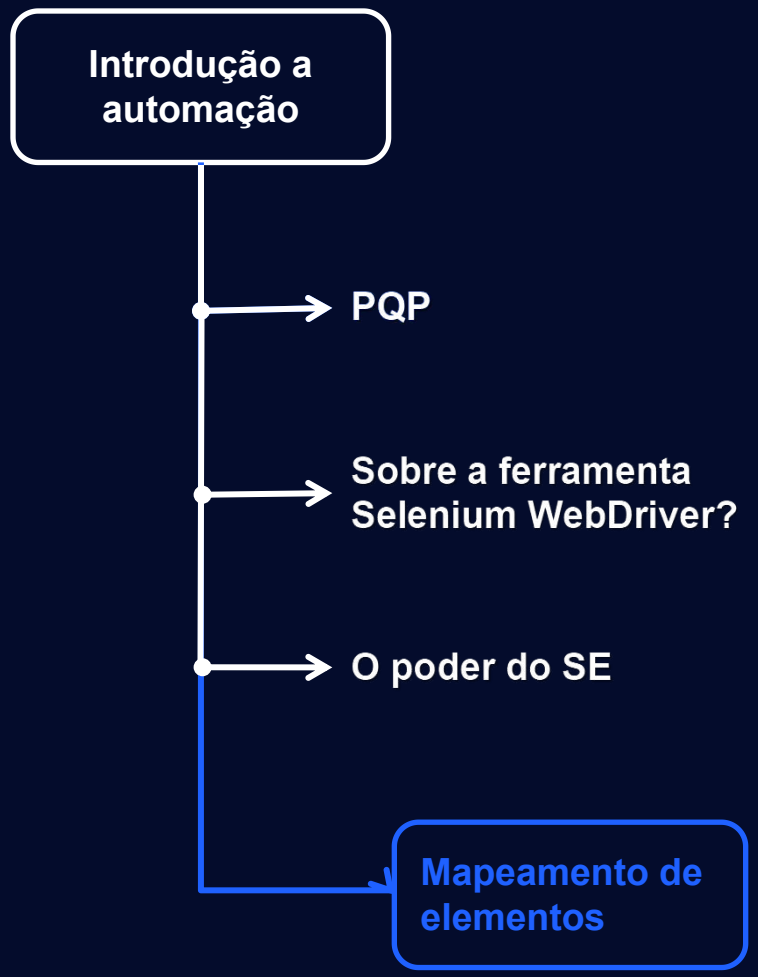
Comando para fechar o navegador

Exemplo:

```
driver.quit();
```

Fechar navegador

Tecnologias de Automação Front End



.manage() .window().maximize()

Comando para maximizar a tela do navegador

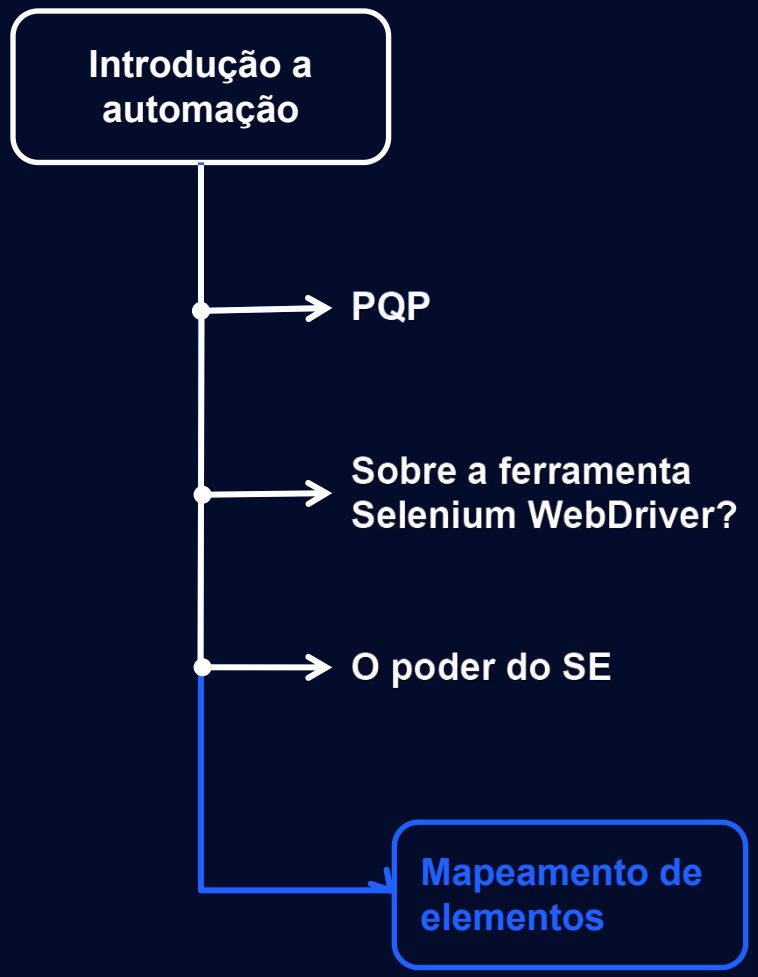
Exemplo:

```
driver.manage().window().maximize();
```

Maximimar

Tela navegador

Tecnologias de Automação Front End



.manage()
.timeouts().implicityWait(tempo, tempo);

Comando para maximizar a tela do navegador

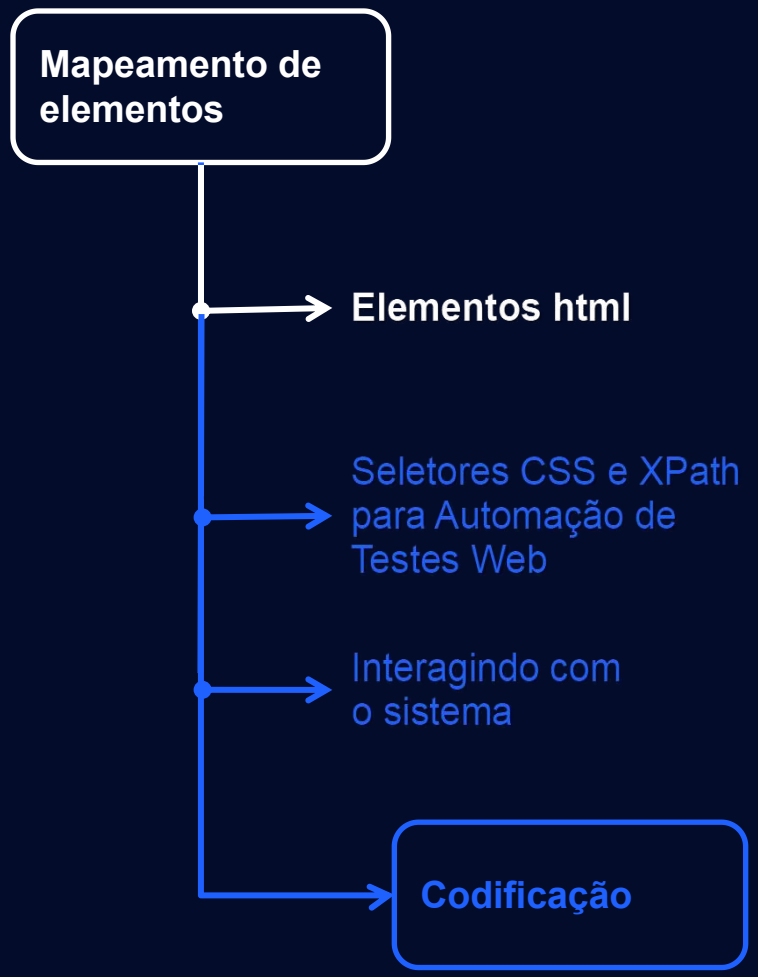
Exemplo:

driver.manage().timeouts().implicityWait(tempo, tempo);

Método de espera implicita

Mapeamento de Elementos

Tecnologias de Automação Front End



- ## HTML - HyperText Markup Language

O HTML é a linguagem de marcação padrão usada para criar páginas da web.

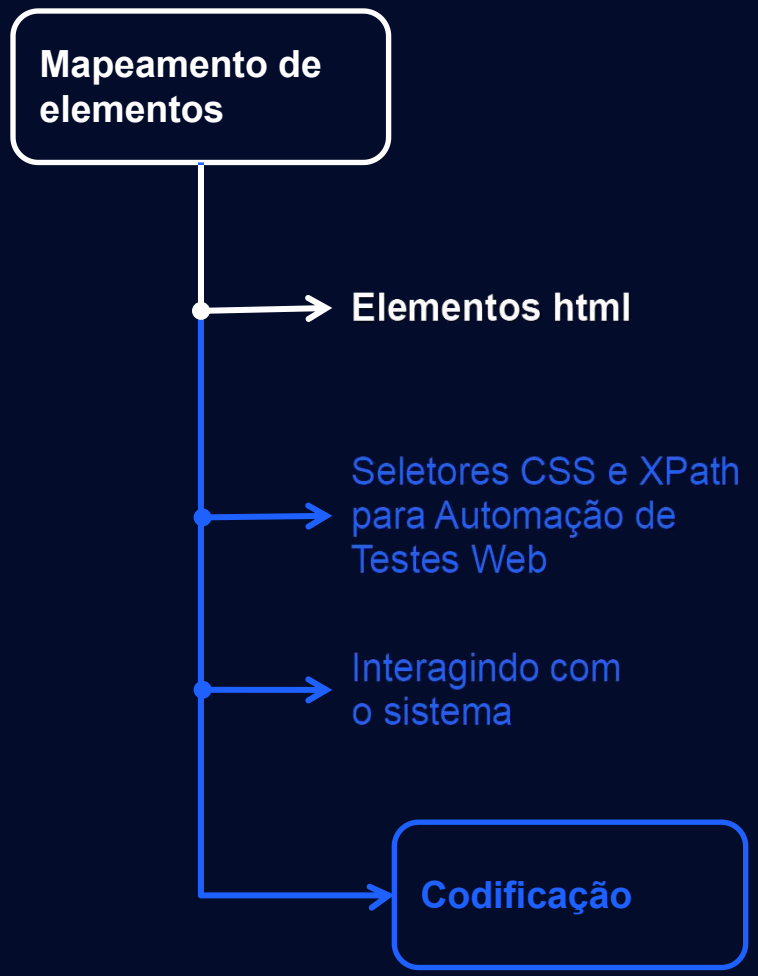
- ## Tags HTML

As tags HTML são elementos fundamentais da linguagem. Elas são usadas para definir a estrutura e o significado do conteúdo em uma página.

Cada tag é representada por um nome entre colchetes angulares (<>)

```
<p>Isto é um parágrafo.<p>
<button> Login <button>
<input>
```

Tecnologias de Automação Front End



- ## Atributos

As tags HTML também podem ter atributos, que fornecem informações adicionais sobre a tag ou modificam seu comportamento.

```
<button type="submit" data-qa="login-button"> Login </button>
<input type="email" data-qa="login-email" placeholder="Email Address">
```

Os atributos são especificados dentro das tags de abertura e geralmente têm um nome e um valor.

```
<a href="https://www.google.com">Visite a google</a>
```

- ## ID

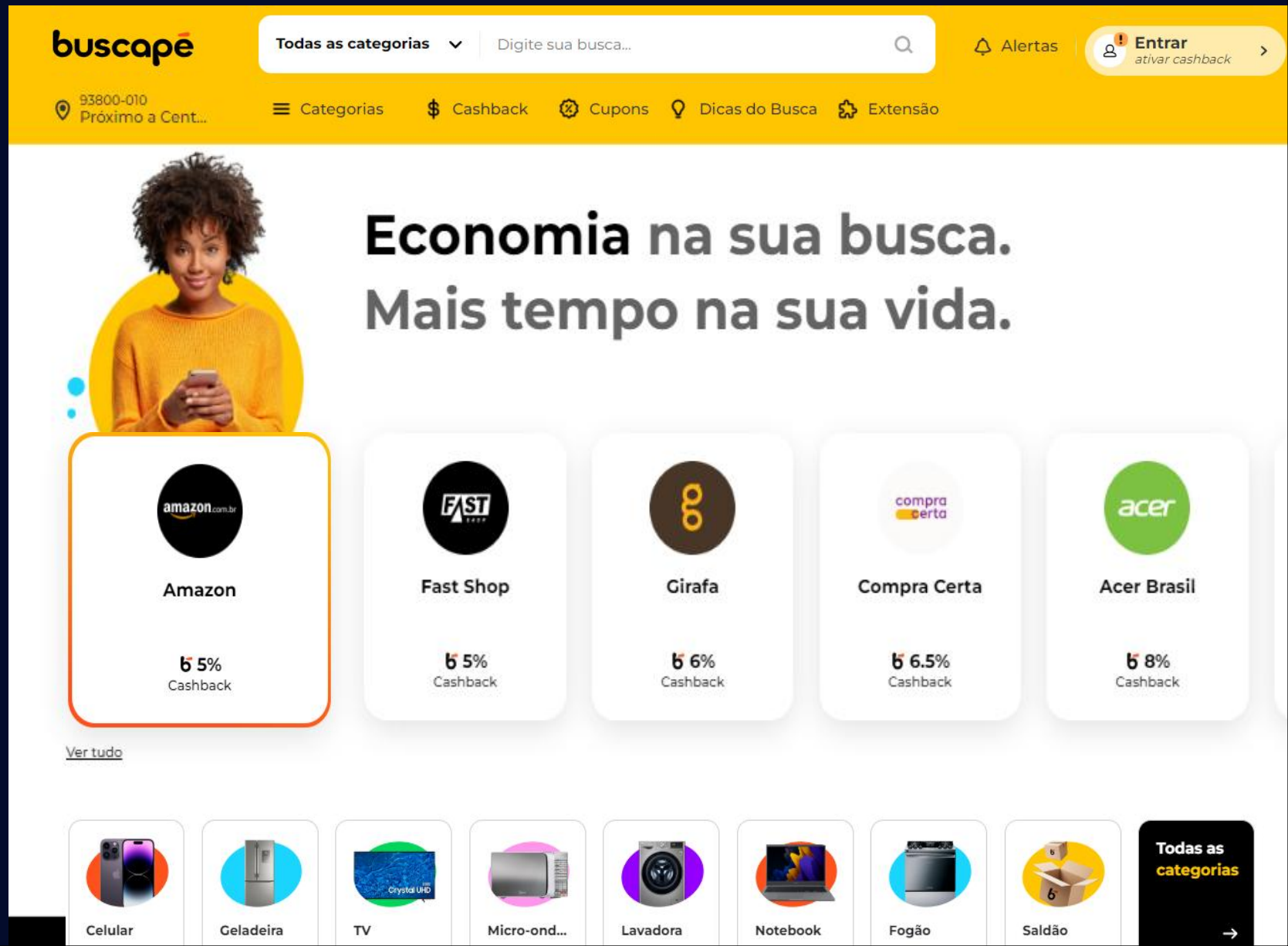
O atributo id é especial e usado para identificar exclusivamente um elemento em uma página.

```
<div id="header”>Cabeçalho da Página</div>
```

Tecnologias de Automação Front End

Mapeamento de elementos

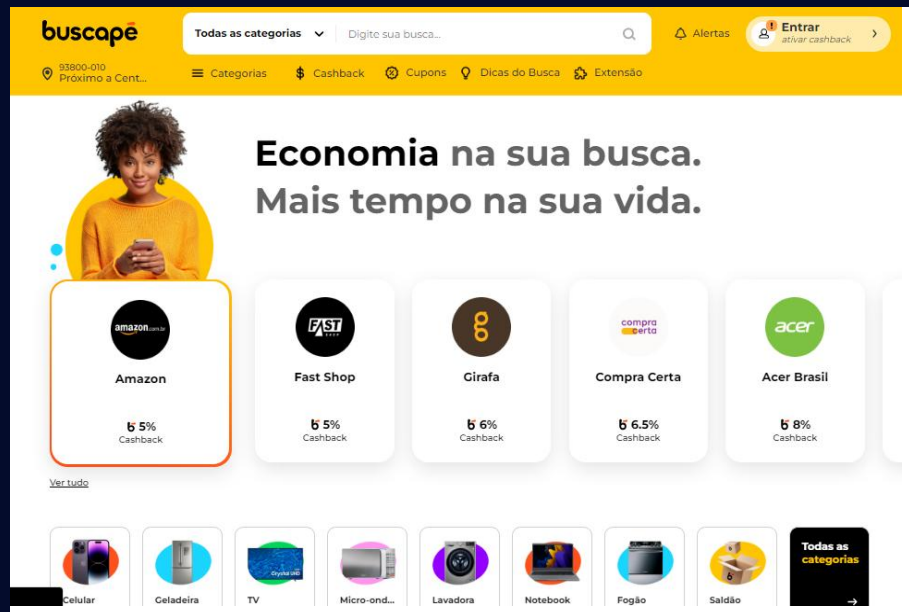
- Elementos html
- Seletores CSS e XPath para Automação de Testes Web
- Interagindo com o sistema
- **Codificação**



Tecnologias de Automação Front End

Mapeamento de elementos

- Elementos html
- Seletores CSS e XPath para Automação de Testes Web
- Interagindo com o sistema
- **Codificação**



Tecnologias de Automação Front End

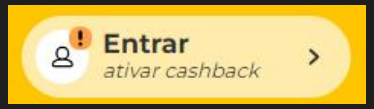
- **Mapeamento de elementos**
  - **Elementos html**
  - **Seletores CSS e XPath para Automação de Testes Web**
  - Interagindo com o sistema
  - **Codificação**

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Automation Exercise - Signup / Login</title>
6  </head>
7      <body>
8
9      <form>
10         <button type="submit" data-qa="login-button" class="btn btn-default">Entrar<button>
       </form>
11
12      # Mapeamento do botão
13          => [type="submit"]              => button[type="submit"]
14          => [data-qa="login-button"]     => button[data-qa="login-button"]
            => [class="btn btn-default"]    => button[class="btn btn-default"]
15
16
17     </body>
18 </html>
19
```

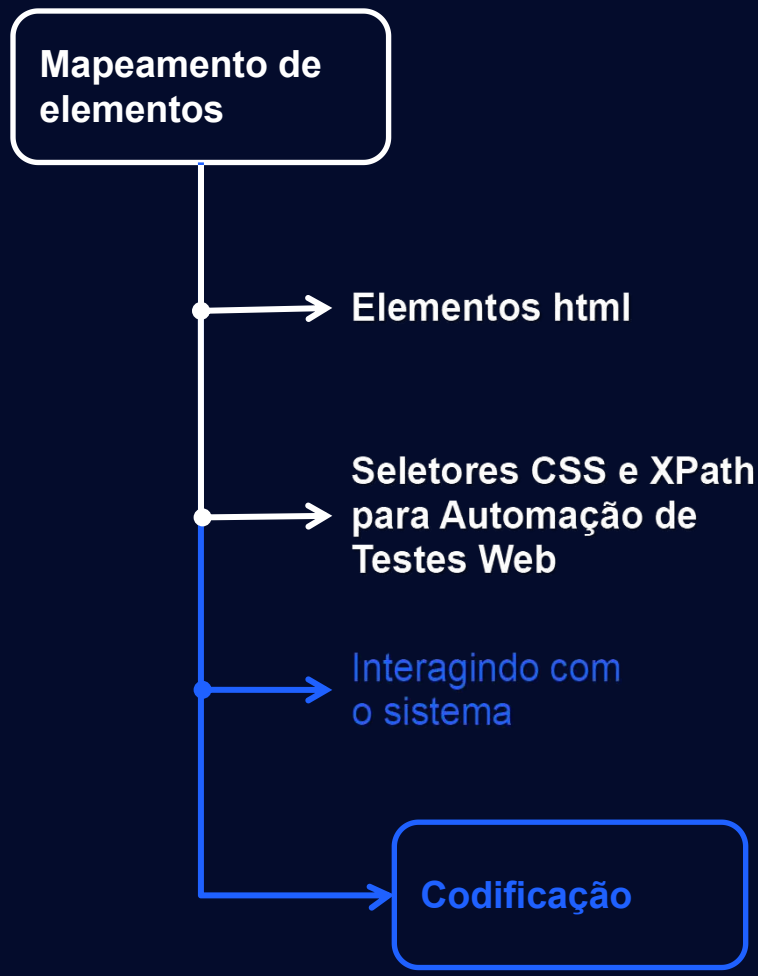
Tecnologias de Automação Front End

- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Automation Exercise - Signup / Login</title>
</head>
    <body>
    <form>
        <button type="submit" data-qa="login-button" class="btn btn-default">Entrar<button>
    </form>
    <form>
        <button type="submit" data-qa="login-button" class="btn btn-default">Entrar<button>
    </form>

   # Mapeamento do segundo botão

      => form:nth-child(2) [type="submit"]              => form:nth-child(2) button[type="submit"]
      => form:nth-child(2) [data-qa="login-button"]     => form:nth-child(2) button[data-qa="login-button"]
      => form:nth-child(2) [class="btn btn-default"]    => form:nth-child(2) button[class="btn btn-default"]


    </body>
</html>
```



Tecnologias de Automação Front End



Seletores CSS

Os seletores CSS (Cascading Style Sheets) são padrões usados para selecionar elementos HTML com base em suas propriedades visuais ou hierarquia.

Alguns exemplos de seletores CSS:

Seleção por tag: **p** seleciona todos os parágrafos.
Seleção por classe: **.botao** seleciona elementos com a classe "botao".
Seleção por ID: **#header** seleciona o elemento com o ID "header".

XPath

XPath é uma linguagem de consulta utilizada para navegar e selecionar elementos em documentos XML, como páginas HTML. Ele fornece uma maneira precisa e flexível de selecionar elementos.

Seleção por tag: **/html/body/p** seleciona todos os parágrafos no corpo do documento.

Seleção por atributo: **//input[@name='username']** seleciona o elemento input com o atributo name igual a "username".

Seleção por posição: **(//div)[2]** seleciona o segundo elemento div na página.

Tecnologias de Automação Front End

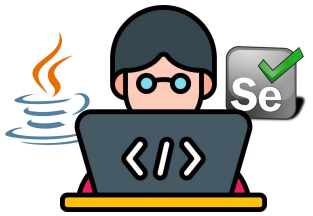
- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação

Como fazemos a interação com o sistema ??

Tecnologias de Automação Front End

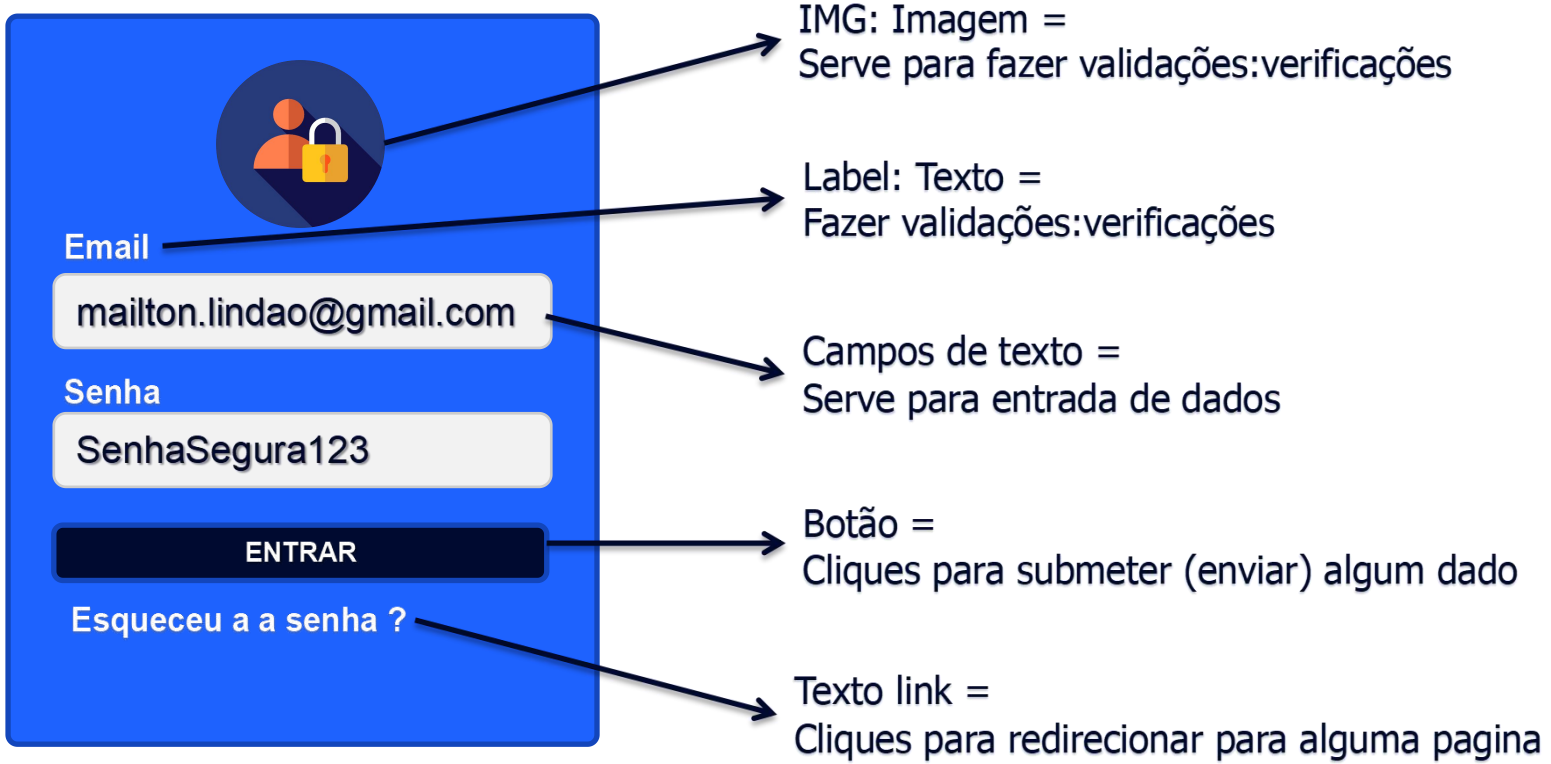
- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação

Mapeamento de elemento



Tecnologias de Automação Front End

- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação



Mapeamento de elemento

Tecnologias de Automação Front End

- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação

Mapeamento de elemento

IMG: Imagem =
Serve para fazer validações:verificações

Código Selenium:

```
findElement(By.id("id="imgLogin"")).click();
```

Explicação da interação :

1. findElement = Função de busca
2. (= Aberta da função
3. By.id = Tipo do mapeamento
4. ("imgLogin"") = Mapeamento do elemento
5. . = Acessando outros metodos
6. click() = Método de interação clique
7. ; = Simbolo de finalização

Tecnologias de Automação Front End

- Mapeamento de elementos
  - Elementos html
  - Seletores CSS e XPath para Automação de Testes Web
  - Interagindo com o sistema
  - Codificação

Mapeamento de elemento

IMG: Imagem = Serve para fazer validações:verificações
findElement(By.id(“id="img"”));

Label: Texto = Fazer validações:verificações
findElement(By.CssSelector(“label="email"”));

Campos de texto = Serve para entrada de dados
findElement(By.CssSelector(“label="email"”));

Botão = Cliques para submeter (enviar) algum dado
findElement(By.className(“class="btnEntrar"”));

Texto link = Cliques para redirecionar para alguma pagina
findElement(By.linkText(“link="esqueceusenha”));

Tecnologias de Automação Front End

Mapeamento de elementos

- Elementos html
- Seletores CSS e XPath para Automação de Testes Web
- Interagindo com o sistema
- Codificação

Existe diversos tipos de mapeamento onde o automatizor pode usar para mapear o elemento:

Mapeamento de elemento

findElement(

```
By.id("id="img""));
By.CssSelector("label="email""));
By.CssSelector("label="email""));

By.className("class="btnEntrar""));
By.linkText("link="esqueceusenha"));
By.xpath("link="esqueceusenha"));
```

porém

Mais utilizados em boas praticas é o:

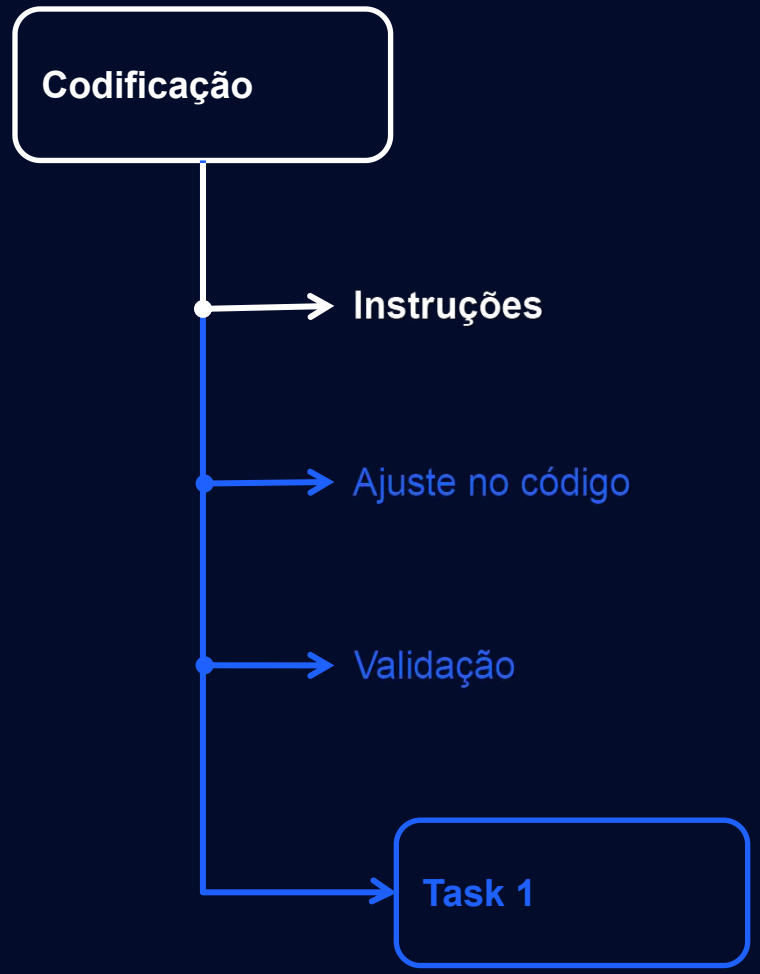
```
By.CssSelector("label="email""));
```

porque ?

Ele consegue mapear quase todos os tipos:

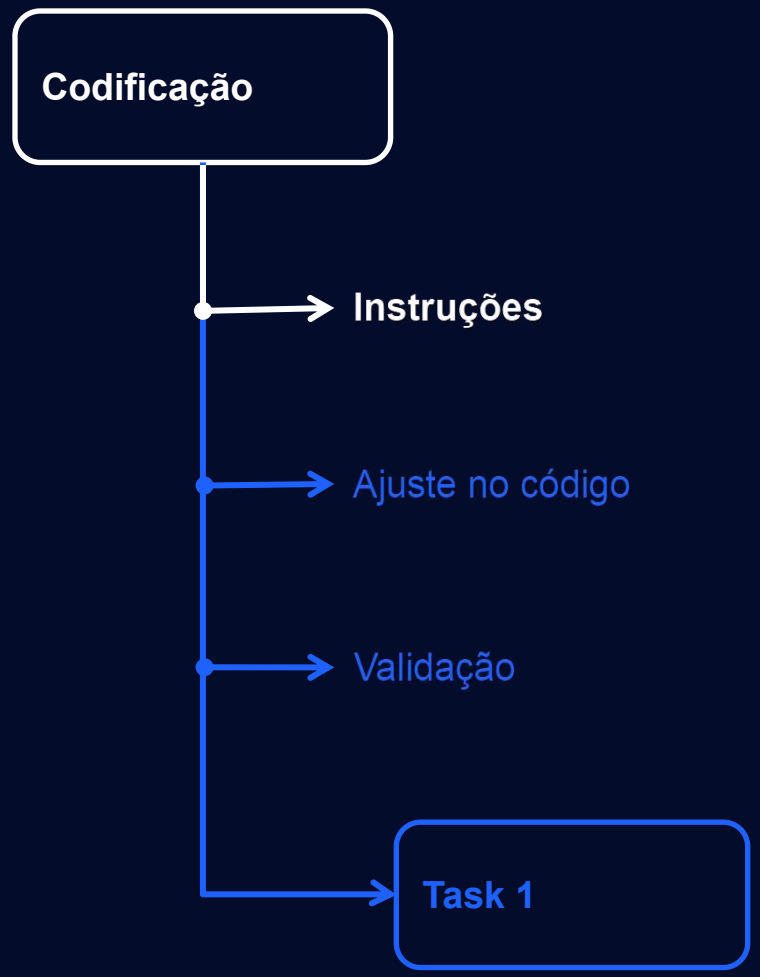
```
By.CssSelector("label="email""));
```

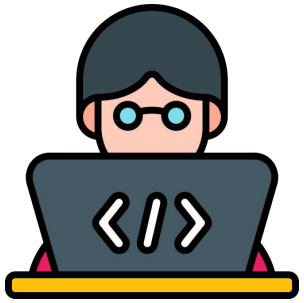
Tecnologias de Automação Front End



Instruções

Tecnologias de Automação Front End





Instruções

1. Siga as instruções do TechLeader
2. Copie cada linha de código apresentado na IDE
3. O resultado final deverá ser a execução com sucesso do cenário automatizado.

Tecnologias de
Automação
Front
End

Codificação

tafe_project1 > src > main > java > AulaA > LoginTest

```
1  package AulaA;
2
3  public class LoginTest {
4
5      // 1 - Declarar as bibliotecas em variáveis
6      public static WebDriver driver;
7      public static WebDriverWait wait;
8
9      // 2 - Criar primeiro método de testes
10     public void deveFazerLoginComSucesso(){
11
12         // 3 - Coloco o caminho do meu "ChromeDriver" em uma variável
13         String caminhoDriver = "driver/chromedriver.exe";
14         // 4 - Setar o recurso que vou utilizar passando o caminho do driver
15         System.setProperty("webdriver.chrome.driver", caminhoDriver);
16
17         // 5 - instancio "ChromeDriver" na variável "driver"
18         driver = new ChromeDriver();
19         /* 6 - Instancio "WebDriverWait" na variável "wait",
20         * passando o driver + um tempo de espera global */
21         wait = new WebDriverWait(driver, timeOutInSeconds: 40);
22         // 7 - Abrir o navegador
23         driver.get("https://www.automationexercise.com/login");
24         // 8 - Maximizar navegador
25         driver.manage().window().maximize();
26     }
```

Tecnologias de
Automação
Front
End

Codificação

tafe_project1 ▸ Version control ▸ Current File

Project

tafe_project1 C:\Users\mai
- .idea
- driver
- src
- main
  - java
    - AulaA
      - LoginTest
    - AulaAdesafio
    - AulaB
    - AulaBdesafio
    - AulaC
    - AulaCdesafio
  - resources
- test
  - java
    - pasta
      - LoginTest

LoginTest.java

```
1  package AulaA;
2
3  public class LoginTest {
4
5      public static WebDriver driver;
6      public static WebDriverWait wait;
7
8      public void deveFazerLoginComSucesso(){
9          String caminhoDriver = "driver/chromedriver.exe";
10         System.setProperty("webdriver.chrome.driver", caminhoDriver);
11         driver = new ChromeDriver();
12         wait = new WebDriverWait(driver, timeOutInSeconds: 40);
13         driver.get("https://www.automationexercise.com/login");
14         driver.manage().window().maximize();
15
16         // 1- Esperar o campo de "login" carregar na tela e depois escrever no campo.
17         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-qa=\"login-email\"]")));
18         driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...charSequences: "vs@gmail.com");
19
20         // 2- Esperar o campo de "senha" carregar na tela e depois escrever no campo.
21         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"login-password\"]")));
22         driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSequences: "123456");
```

tafe_project1 › src › main › java › AulaA › LoginTest

Tecnologias de
Automação
Front
End

Codificação

```
tafe_project1 > src > main > java > AulaA > LoginTest
```

```
Project
tafe_project1 C:\Users\mai
.idea
driver
src
main
java
AulaA
LoginTest
AulaAdesafio
AulaB
AulaBdesafio
AulaC
AulaCdesafio
resources
test
java
pasta
LoginTest
```

LoginTest.java

```
1  package AulaA;
2
3  public class LoginTest {
4      public static WebDriver driver;
5      public static WebDriverWait wait;
6      @Test
7      public void deveFazerLoginComSucesso(){
8          String caminhoDriver = "driver/chromedriver.exe";
9          System.setProperty("webdriver.chrome.driver", caminhoDriver);
10         driver = new ChromeDriver();
11         wait = new WebDriverWait(driver, timeOutInSeconds: 40);
12         driver.get("https://www.automationexercise.com/login");
13         driver.manage().window().maximize();
14         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-qa=\"login-email\"]")));
15         driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...charSequences: "vs@gmail.com");
16         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"login-password\"]")));
17         driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSequences: "123456");
18         // 3 - Esperar o botão login carregar na tela
19         String btnLogin = "#form  div div div.col-sm-4.col-sm-offset-1 div  form > button";
20         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogin)));
21         // 4 - Clicar no botão login
22         driver.findElement(By.cssSelector(btnLogin)).click();
23
24     }
```

Tecnologias de
Automação
Front
End

Codificação

tafe_project1

Project

tafe_project1 C:\Users\m

- .idea
- driver
- src
- main
  - java
    - AulaA
      - LoginTest
      - AulaAdesafio
      - AulaB
      - AulaBdesafio
      - AulaC
      - AulaCdesafio
  - resources
- test
  - java
    - pasta
      - LoginTest

tafe_project1 › src › main

New

Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V

Find Usages Alt+F7
Analyze

Refactor

Bookmarks

Browse Type Hierarchy Ctrl+H
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Excluir
Override File Type

Build Module 'tafe_project1'
Run 'LoginTest' Ctrl+Shift+F10
Debug 'LoginTest'
Run 'LoginTest' with Coverage
Modify Run Configuration...

Open in Right Split Shift+Enter
Open In

File

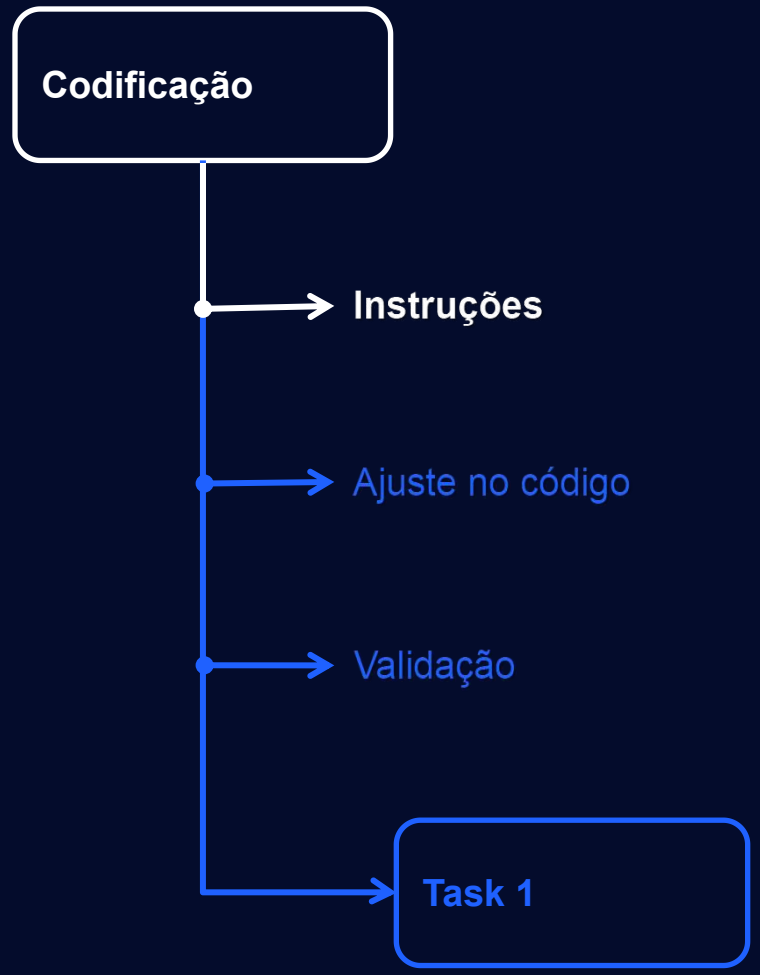
```
){
driver.exe";
driver", caminhoDriver);

OutInSeconds: 40);
rcise.com/login");

cated(By.cssSelector("input[data-qa=\"login-email\"]")));
("login-email\"])).sendKeys( ...charSequences: "vs@gmail.com");
cated(By.cssSelector("[data-qa=\"login-password\"]")));
in-password\"])).sendKeys( ...charSequences: "123456");

4.col-sm-offset-1 div  form > button";
ntLocated(By.cssSelector(btnLogin)));

.click();
```

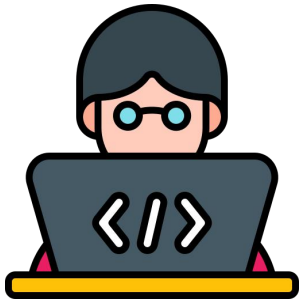
Tecnologias de
Automação
Front
End

Codificação

```
20      System.setProperty("webdriver.chrome.driver", caminhoDriver);
21      driver = new ChromeDriver();
22      wait = new WebDriverWait(driver, timeOutInSeconds: 40);
23      driver.get("https://www.automationexercise.com/login");
24      driver.manage().window().maximize();
25
26      wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-q
27      driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...char
28      wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"l
29      driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSec
30
31      // 5 - Esperar o botão login carregar na tela
32      String btnLogin = "#form  div div div.col-sm-4.col-sm-offset-1 div  form > button";
33      wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogin)));
34
35      // 6 - Clicar no botão login
36      driver.findElement(By.cssSelector(btnLogin)).click();
37
38
39
40
```

Tecnologias de Automação Front End



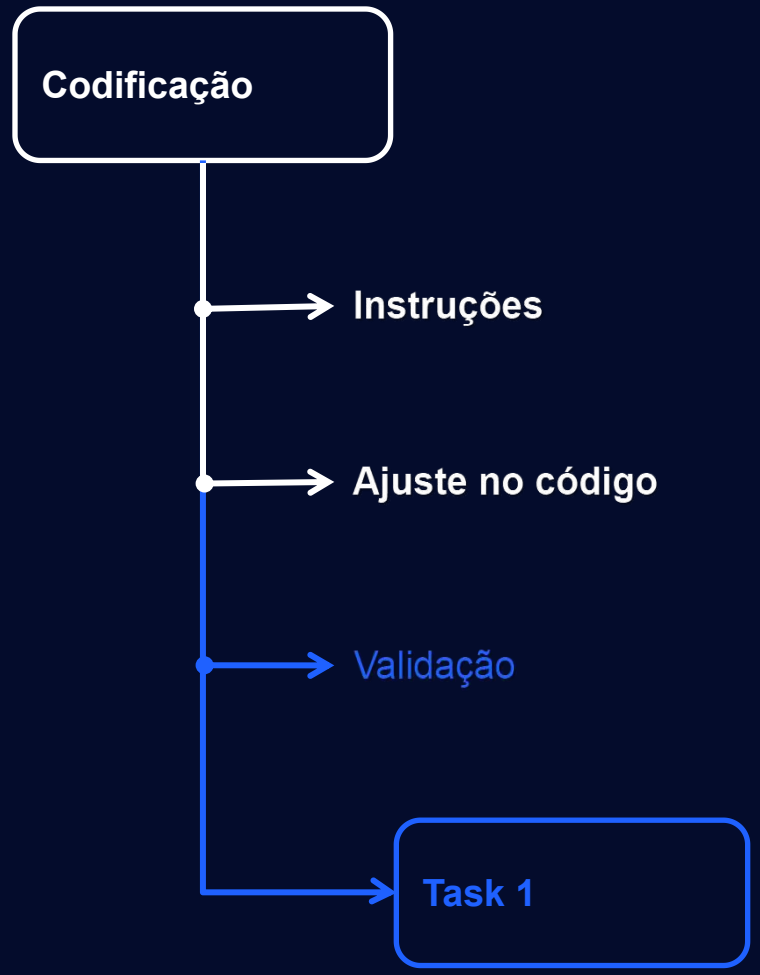


Instruções

Check list de automação:

- [x] Abrimos o navegador ?
- [x] Fizemos o cenário de login ?
- [ ] Código organizado ?
- [ ] Validamos se estamos na pagina certa após a conclusão do testes ?
- [ ] Fechamos o navegador ?

Tecnologias de Automação Front End



Ajuste no código

Tecnologias de
Automação
Front
End

Codificação

Esse código pode ser melhor !

Tecnologias de
Automação
Front
End

Codificação

tafe_project1 > src > main > java > AulaA > LoginTest

```
1  package AulaA;
2
3  public class LoginTest {
4      public static WebDriver driver;
5      public static WebDriverWait wait;
6      @BeforeTest
7      public void abrirNavegador(){
8          String caminhoDriver = "driver/chromedriver.exe";
9          System.setProperty("webdriver.chrome.driver", caminhoDriver);
10         driver = new ChromeDriver();
11         wait = new WebDriverWait(driver, timeOutInSeconds: 40);
12         driver.get("https://www.automationexercise.com/login");
13         driver.manage().window().maximize();
14     }
15     @Test
16     public void deveFazerLoginComSucesso(){
17
18         String btnLogin = "#form  div div div.col-sm-4.col-sm-offset-1 div  form > button";
19
20         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-qa=\"login-email\"]")));
21         driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...charSequences: "vs@gmail.com");
22         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"login-password\"]")));
23         driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSequences: "123456");
24         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogin)));
25         driver.findElement(By.cssSelector(btnLogin)).click();
26     }
27
```

Tecnologias de
Automação
Front
End

Codificação

tafe_project1 | Version control | Current File

Project

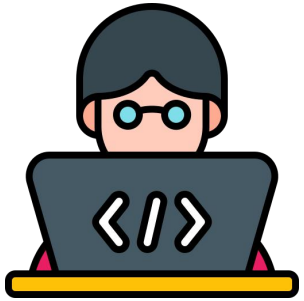
tafe_project1 C:\Users\mai
- .idea
- driver
- src
- main
  - java
    - AulaA
      - LoginTest
    - AulaAdesafio
    - AulaB
    - AulaBdesafio
    - AulaC
    - AulaCdesafio
  - resources
- test
  - java
    - pasta
      - LoginTest

LoginTest.java

```
1  package AulaA;
2
3  public class LoginTest {
4
5      public static WebDriver driver;
6      public static WebDriverWait wait;
7
8      @BeforeTest
9      public void abrirNavegador(){...}
10
11     @Test
12     public void deveFazerLoginComSucesso(){
13
14         String btnLogin = "#form  div div div.col-sm-4.col-sm-offset-1 div  form > button";
15
16         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-qa=\"login-email\"]")));
17         driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...charSequences: "vs@gmail.com");
18         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"login-password\"]")));
19         driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSequences: "123456");
20         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogin)));
21         driver.findElement(By.cssSelector(btnLogin)).click();
22     }
23
24     @AfterTest
25     public void finalizarNavegador(){
26
27         driver.quit();
28
29     }
30
31
```

tafe_project1 > src > main > java > AulaA > LoginTest

Tecnologias de Automação Front End



Ajuste no código

Check list de automação:

- [x] Abrimos o navegador ?
- [x] Fizemos o cenário de login ?
- [x] Código organizado ?
- [ ] Validamos se estamos na pagina certa após a conclusão do testes ?
- [x] Fechamos o navegador ?

Tecnologias de Automação Front End

- Codificação
  - Instruções
  - Ajuste no código
  - Validação
    - Task 1

Tecnologias de
Automação
Front
End

Codificação

tafe_project1 ▾ Version control ▾ Current File ▾

Project ▾

- tafe_project1 C:\Users\mai
  - .idea
  - driver
  - src
    - main
      - java
        - AulaA
          - LoginTest
        - AulaAdesafio
        - AulaB
        - AulaBdesafio
        - AulaC
        - AulaCdesafio
      - resources
    - test
      - java
        - pasta
          - LoginTest

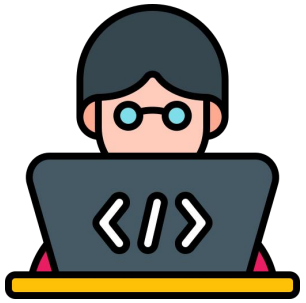
LoginTest.java

```
1  package AulaA;
2
3  public class LoginTest {
4
5      public static WebDriver driver;
6      public static WebDriverWait wait;
7
8      @BeforeTest
9      public void abrirNavegador(){...}
10
11     @Test
12     public void deveFazerLoginComSucesso(){
13
14         String btnLogin = "#form  div div div.col-sm-4.col-sm-offset-1 div  form > button";
15         String btnLogout = "#header > div > div > div > div.col-sm-8 > div > ul > li:nth-child(4) > a";
16
17         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("input[data-qa=\"login-email\"]")));
18         driver.findElement(By.cssSelector("input[data-qa=\"login-email\"]")).sendKeys( ...charSequences: "vs@gmail.com");
19         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("[data-qa=\"login-password\"]")));
20         driver.findElement(By.cssSelector("[data-qa=\"login-password\"]")).sendKeys( ...charSequences: "123456");
21         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogin)));
22         driver.findElement(By.cssSelector(btnLogin)).click();
23
24         // 1 - Esperar o "btnLogout" aparecer na tela e depois ler o texto do botão adicionando na variável
25         wait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector(btnLogout)));
26         String textBtnLogout = driver.findElement(By.cssSelector(btnLogout)).getText();
27
28         // 2 - Validando o "Atual" do "Esperar" => utilizando o TestNg
29         Assert.assertEquals(textBtnLogout, expected: "Logout");
30     }
31
32     @AfterTest
33     public void finalizarNavegador(){...}
34
35 }
```

tafe_project1 › src › main › java › AulaA › LoginTest

Tecnologias de Automação Front End

- Codificação
  - Instruções
  - Ajuste no código
  - Validação
    - Task 1

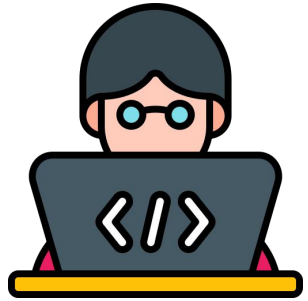


Validação

Check list de automação:

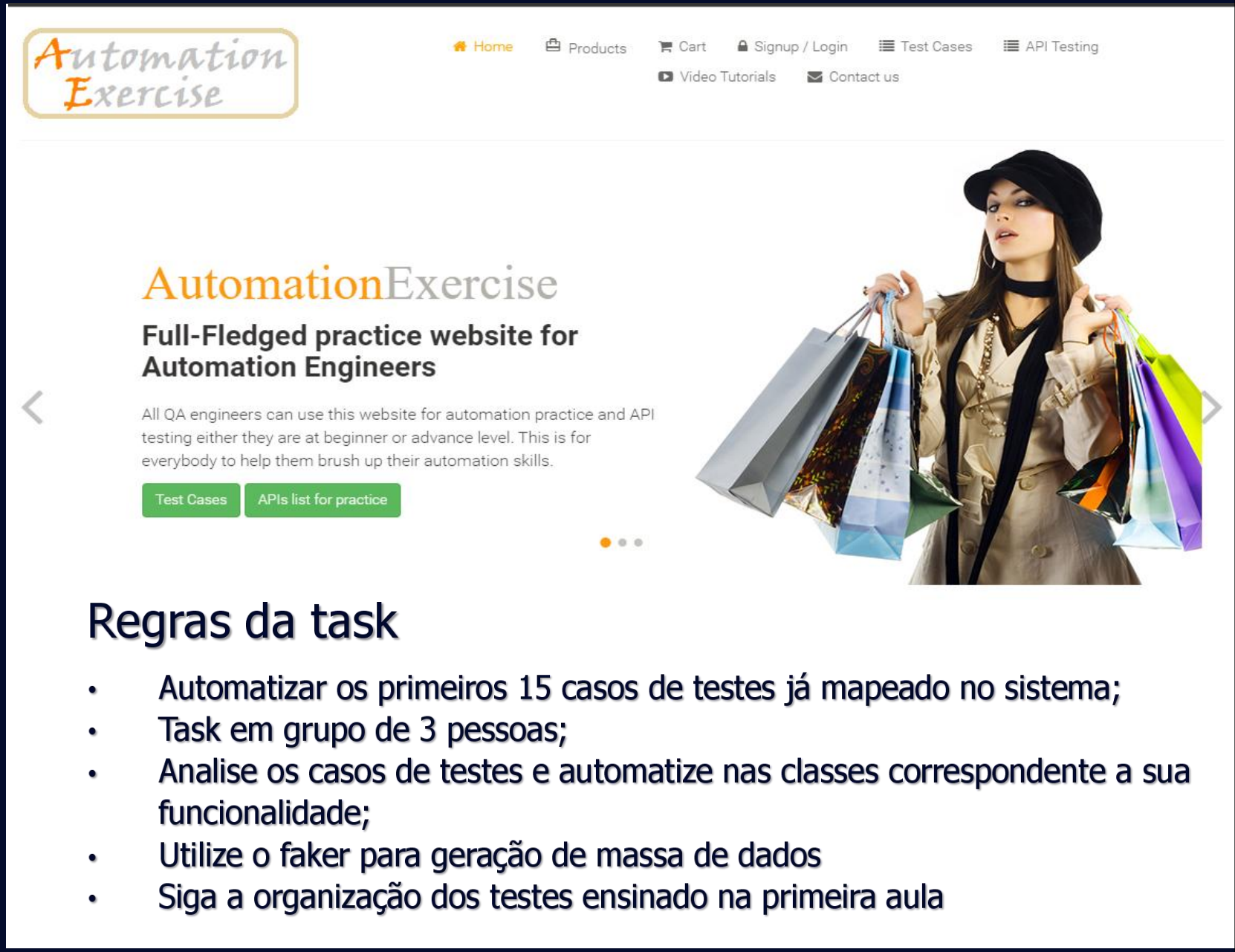
- [x] Abrimos o navegador ?
- [x] Fizemos o cenário de login ?
- [x] Código organizado ?
- [x] Validamos se estamos na pagina certa após a conclusão do testes ?
- [x] Fechamos o navegador ?

Tecnologias de Automação Front End



Task 1

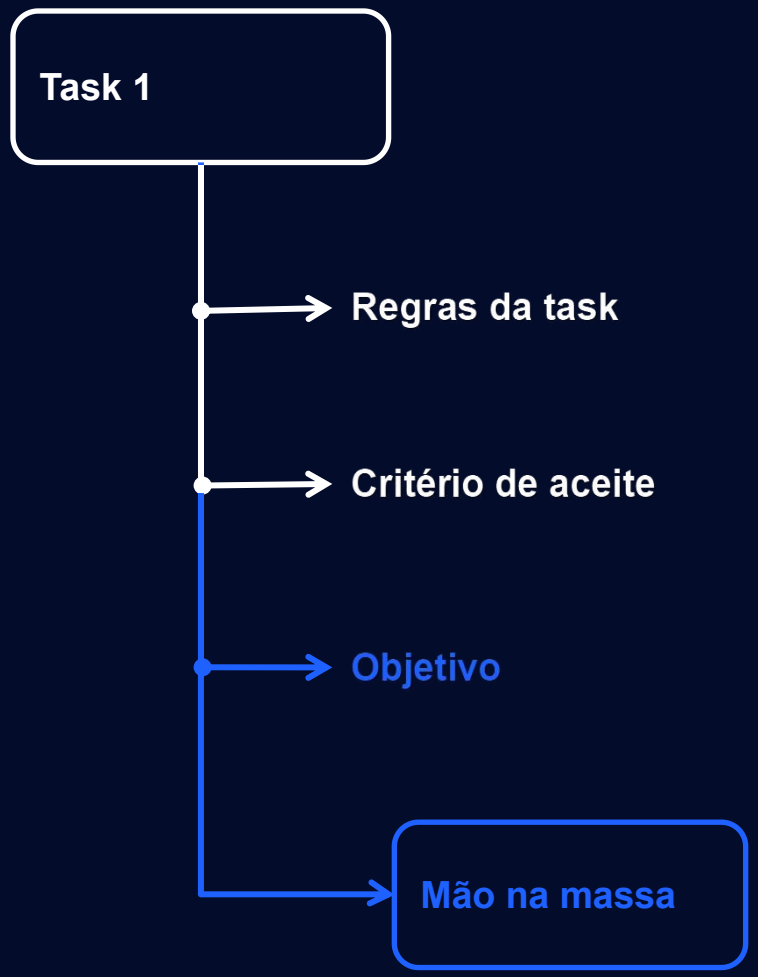
Tecnologias de Automação Front End

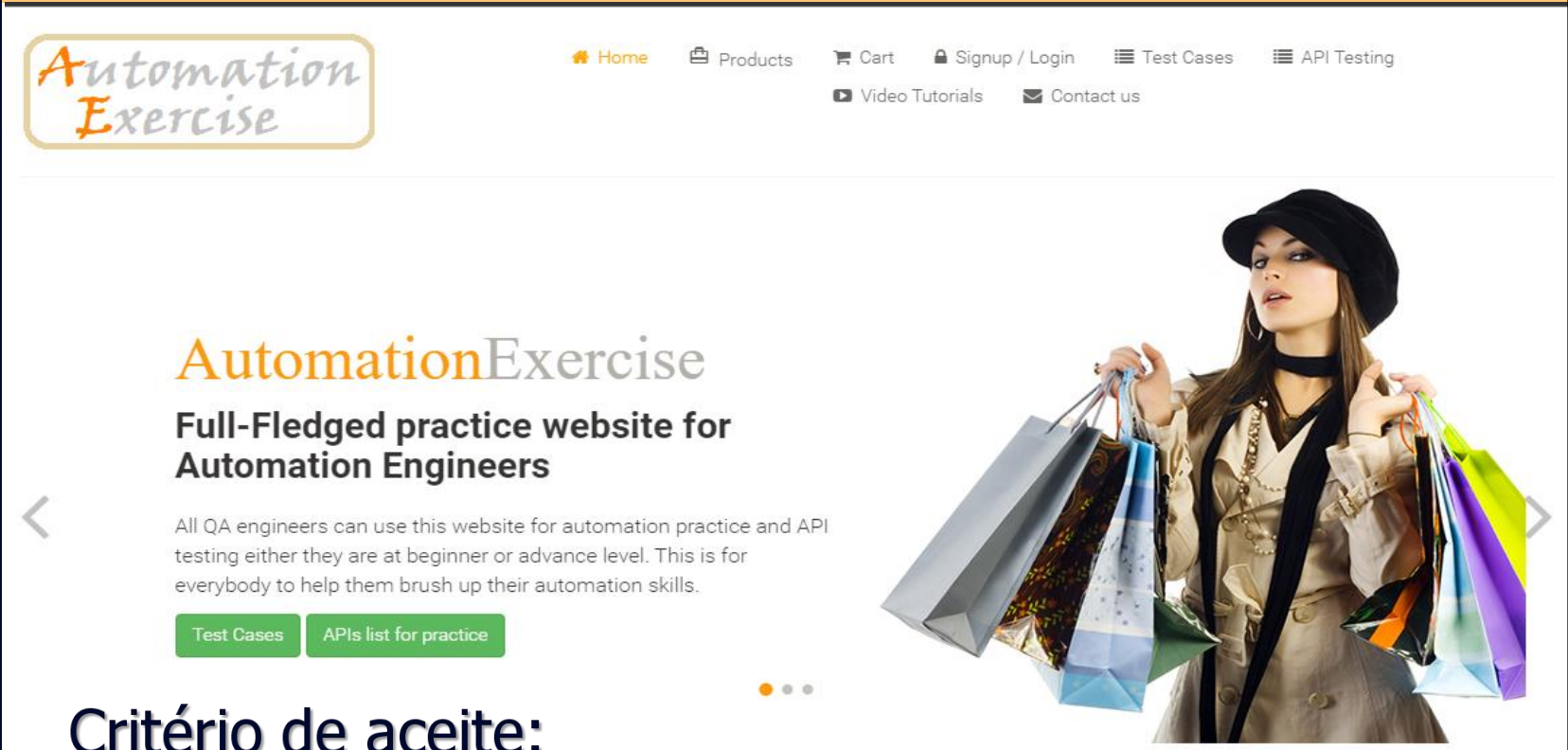


Regras da task

- Automatizar os primeiros 15 casos de testes já mapeado no sistema;
- Task em grupo de 3 pessoas;
- Analise os casos de testes e automatize nas classes correspondente a sua funcionalidade;
- Utilize o faker para geração de massa de dados
- Siga a organização dos testes ensinado na primeira aula

Tecnologias de Automação Front End

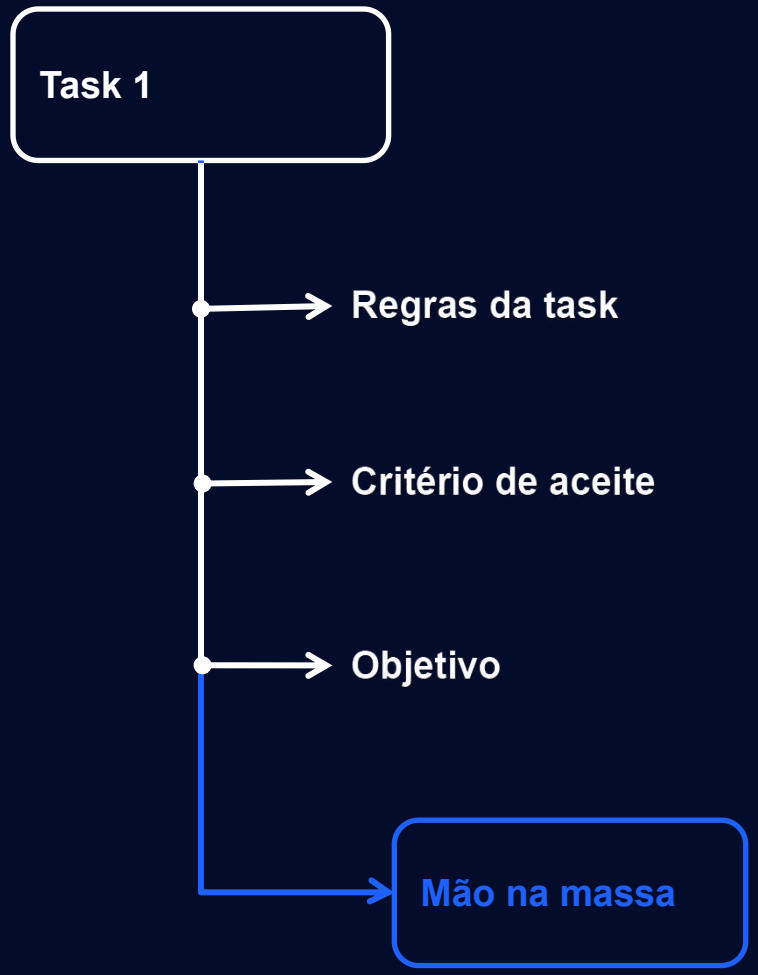


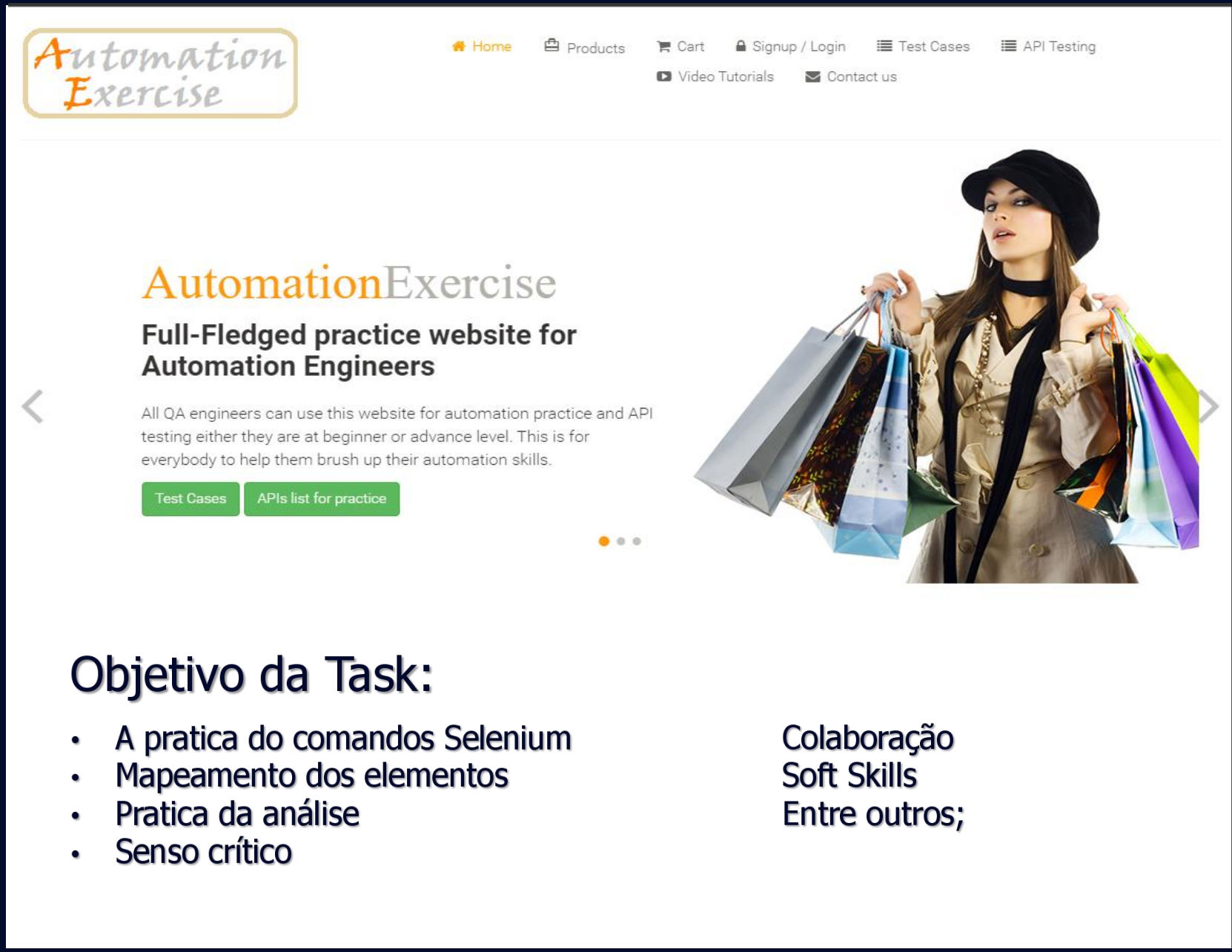


Critério de aceite:

- Automação rodando sem problemas
- Casos de testes mapeados automatizados
- Titulo do cenário deve está conforme caso de testes mapeado
- Documentação "readme" deve conter informações de projeto tais como versão do Chrome;
- Utilização de dados "faker"
- Commits no git por "cenários cocluídos"

Tecnologias de Automação Front End



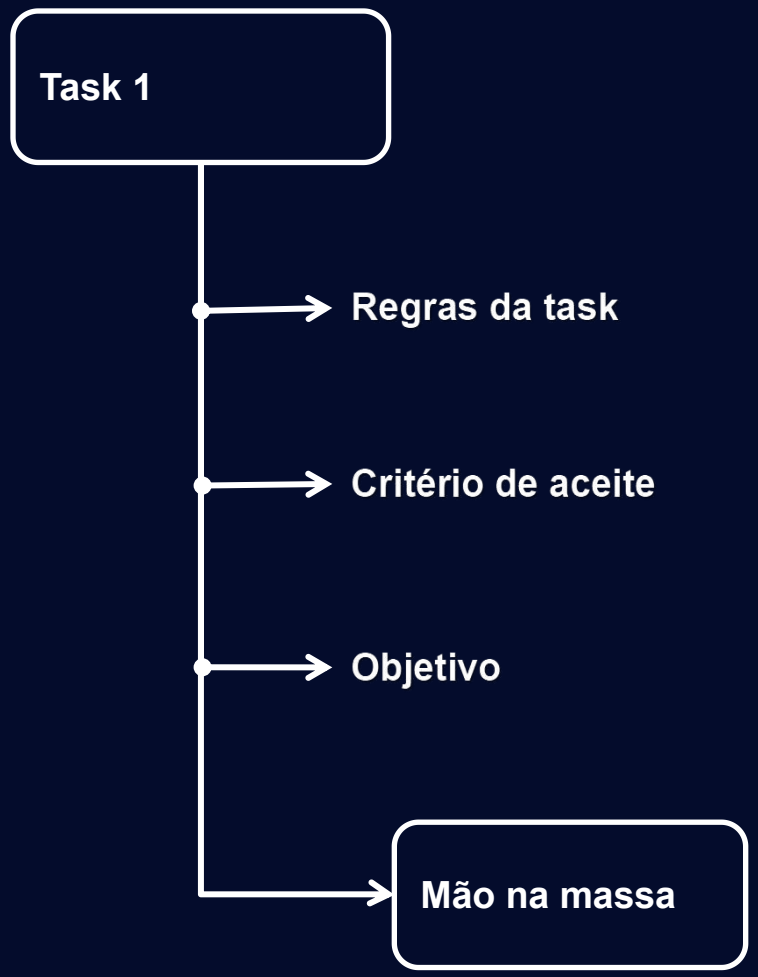


Objetivo da Task:

- A pratica do comandos Selenium
- Mapeamento dos elementos
- Pratica da análise
- Senso crítico

Colaboração
Soft Skills
Entre outros;

Tecnologias de Automação Front End




Let's Tech Up Together