RECRUTA

B2

TREINAMENTO RECRUTA BASE 2

Maio/2021

Quem sou?

BASE2

José Mário Pedrosa de Sousa (Zé Mário ou só Zé)

- **Formação:**
  - Grad. Sistemas de Informação (UNA) e Pós Grad. Engenharia de Software Centrada em Metodologias Ágeis (UNA);
- **Certificações:**
  - CTFL, CTFL-AT, CTFL-PT, CTAL-TA e CTAL-TAE;
- **Atuação:**
  - Arquiteto de Automação de Testes na Base2.

Quem sou?

BASE2

Áldrich Khálled Santos Bezerra

- **Formação:**
  - Tec. Eletrotécnica, Grad. Engenharia Elétrica e Pós Grad. Engenharia de Software;

- **Certificações:**
  - CTFL, CTFL-PT;

- **Atuação:**
  - Arquiteto de Testes na Base2;

Tópicos do treinamento

BASE2

1. Automação de testes
2. Sobre o Selenium WebDriver
3. Revisão pré-requisitos
4. Criando um novo projeto de testes automatizados
5. JUnit
6. “Hello World”
7. Navegação
8. Fechando o browser
9. Obtendo dados do driver
10. Localizando elementos
11. Realizando ações nos elementos
12. Obtendo dados dos elementos
13. Sincronização
14. Prática

1

Automação de testes

BASE2

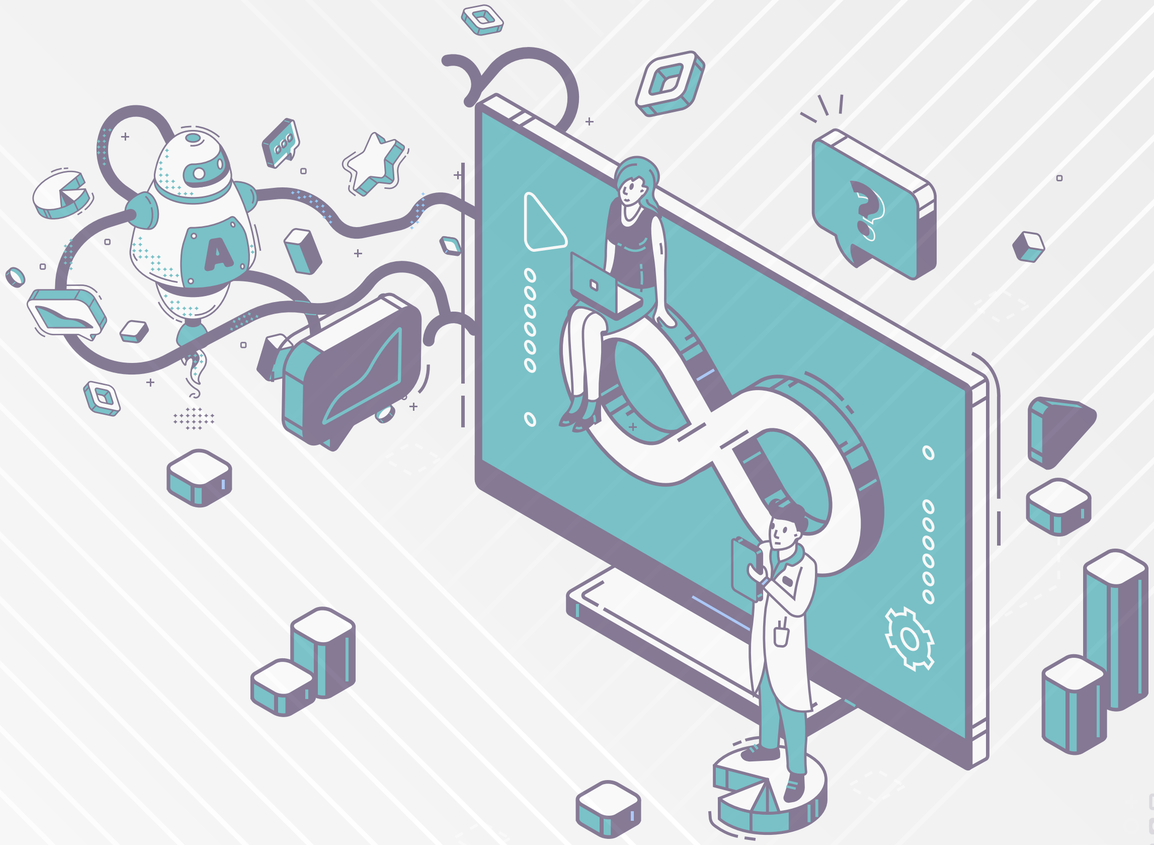
Automação de testes

BASE2

Testes automatizados é processo de implementar scripts dos fluxos de testes manuais para serem executados de maneira automatizada por meio de ferramentas e scripts.

Benefícios da automação de testes:

- Garantia de testes de regressão
- Feedback imediato
- Liberação para o mercado mais rápida
- Maior cobertura geral de testes
- Melhor organização do time
- Redução do esforço manual



2

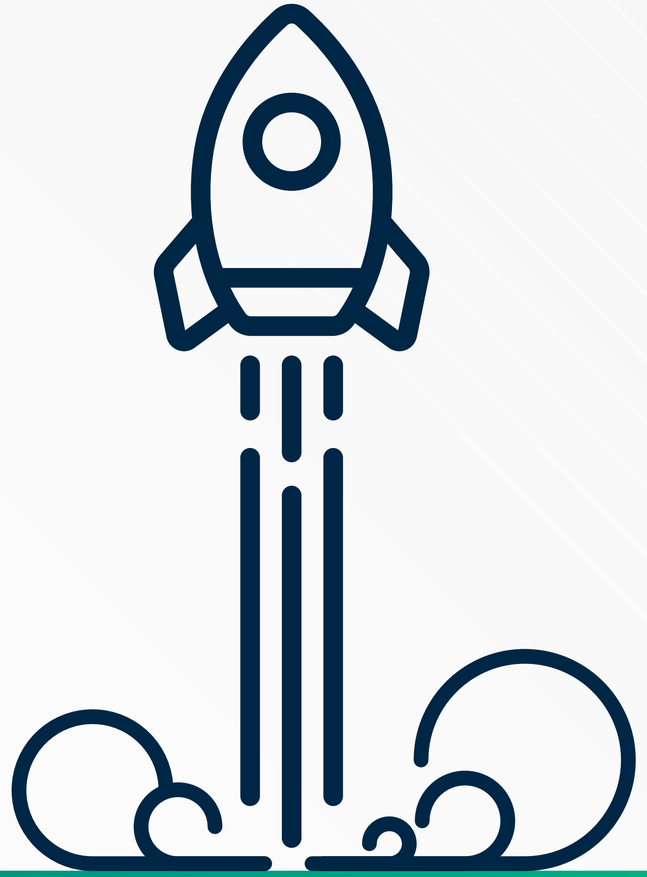
Sobre o Selenium WebDriver

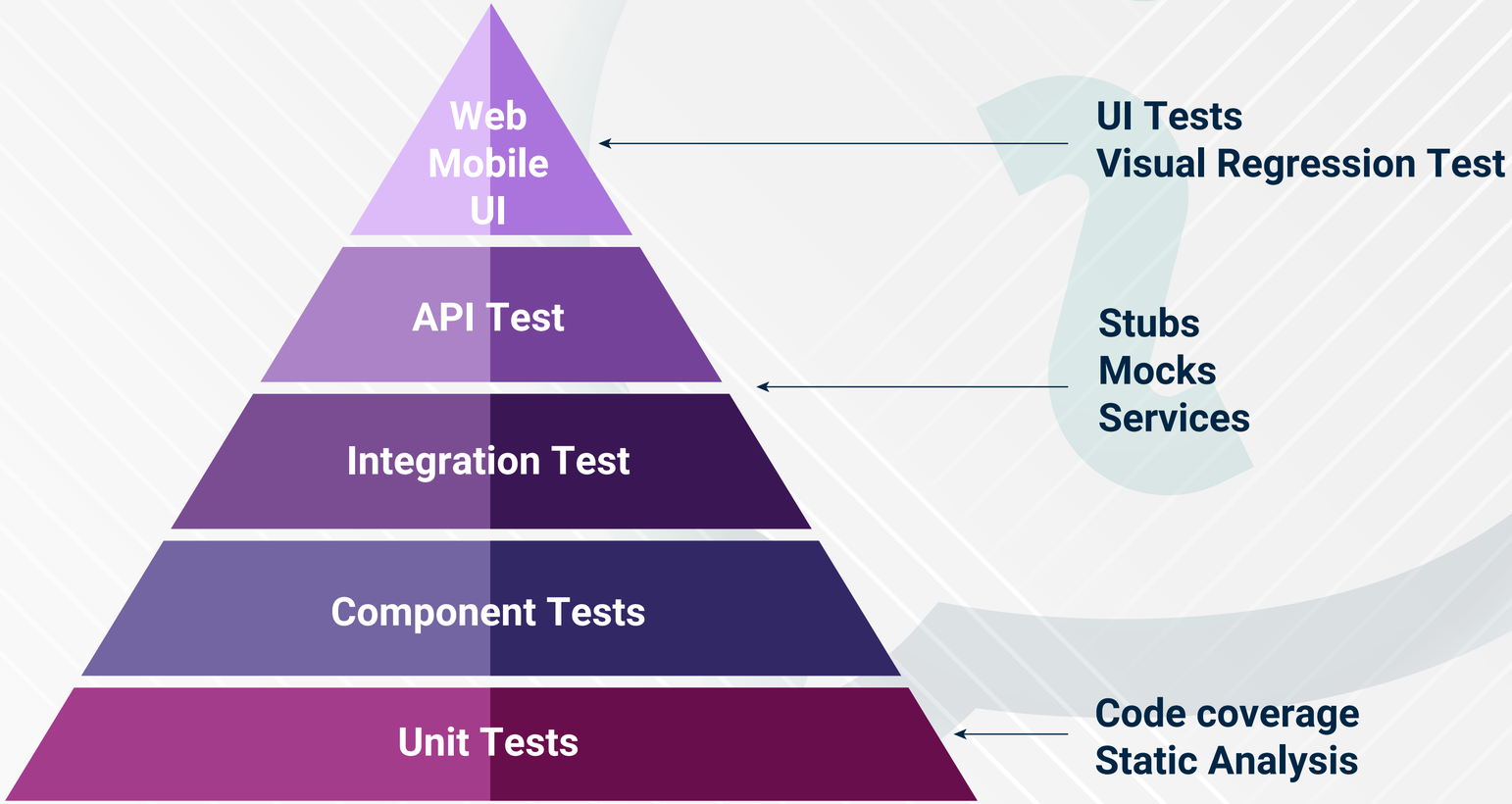
BASE2

BASE2

Different Test Levels

Test Pyramid - Focus on Automated Testing

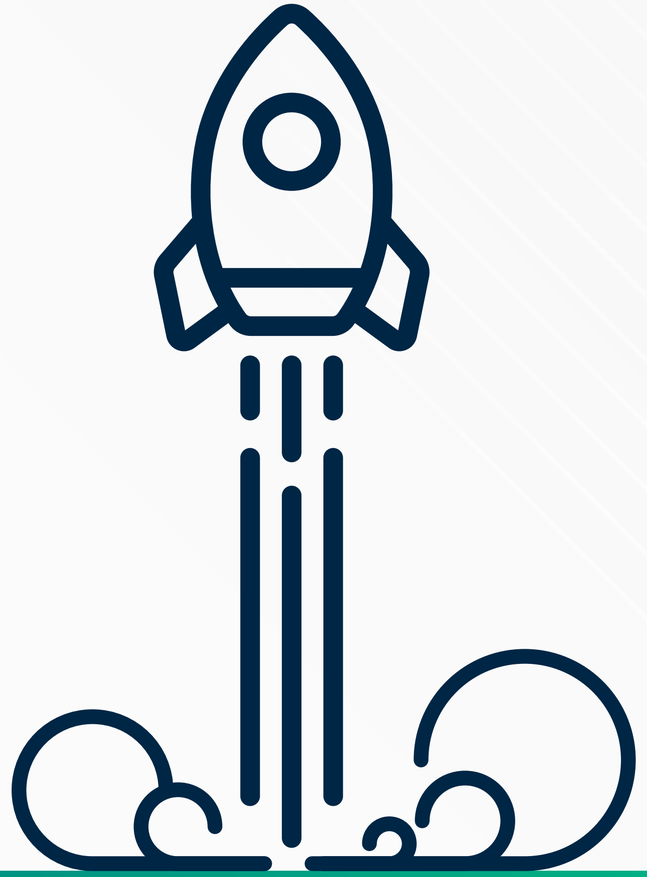


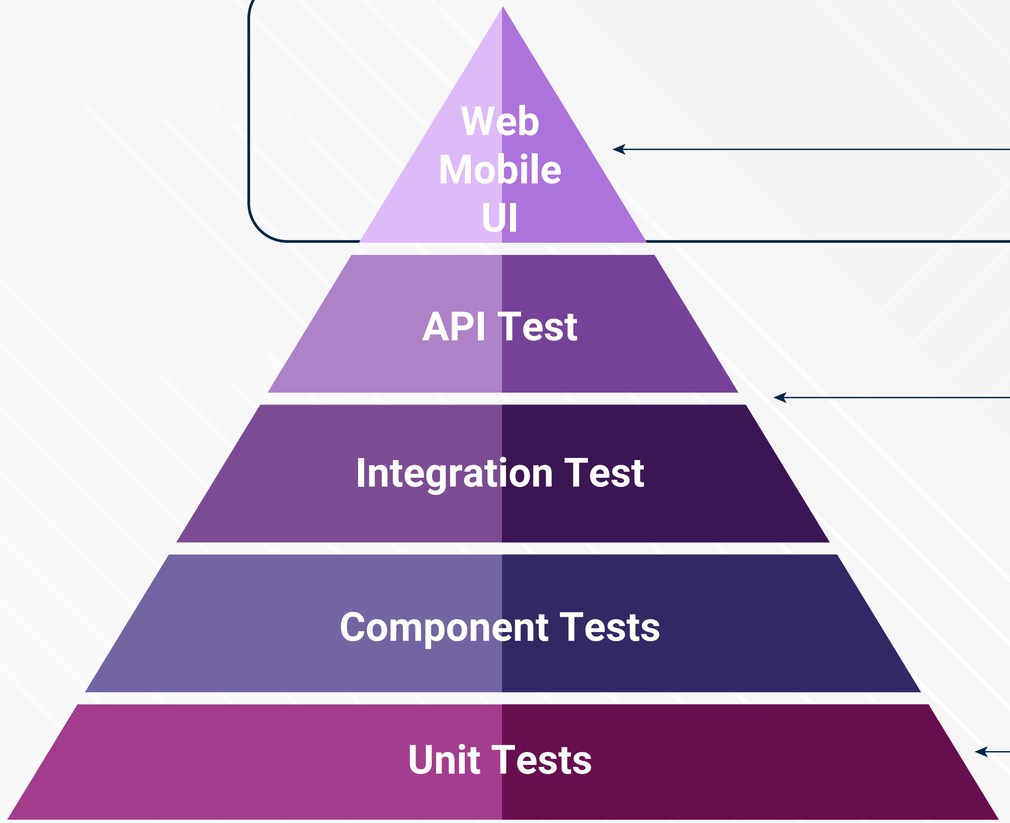


BASE2

Different Test Levels

Test Pyramid - Focus on Automated Testing





UI Tests
Visual Regression Test

Stubs
Mocks
Services

Code coverage
Static Analysis

Sobre o Selenium WebDriver

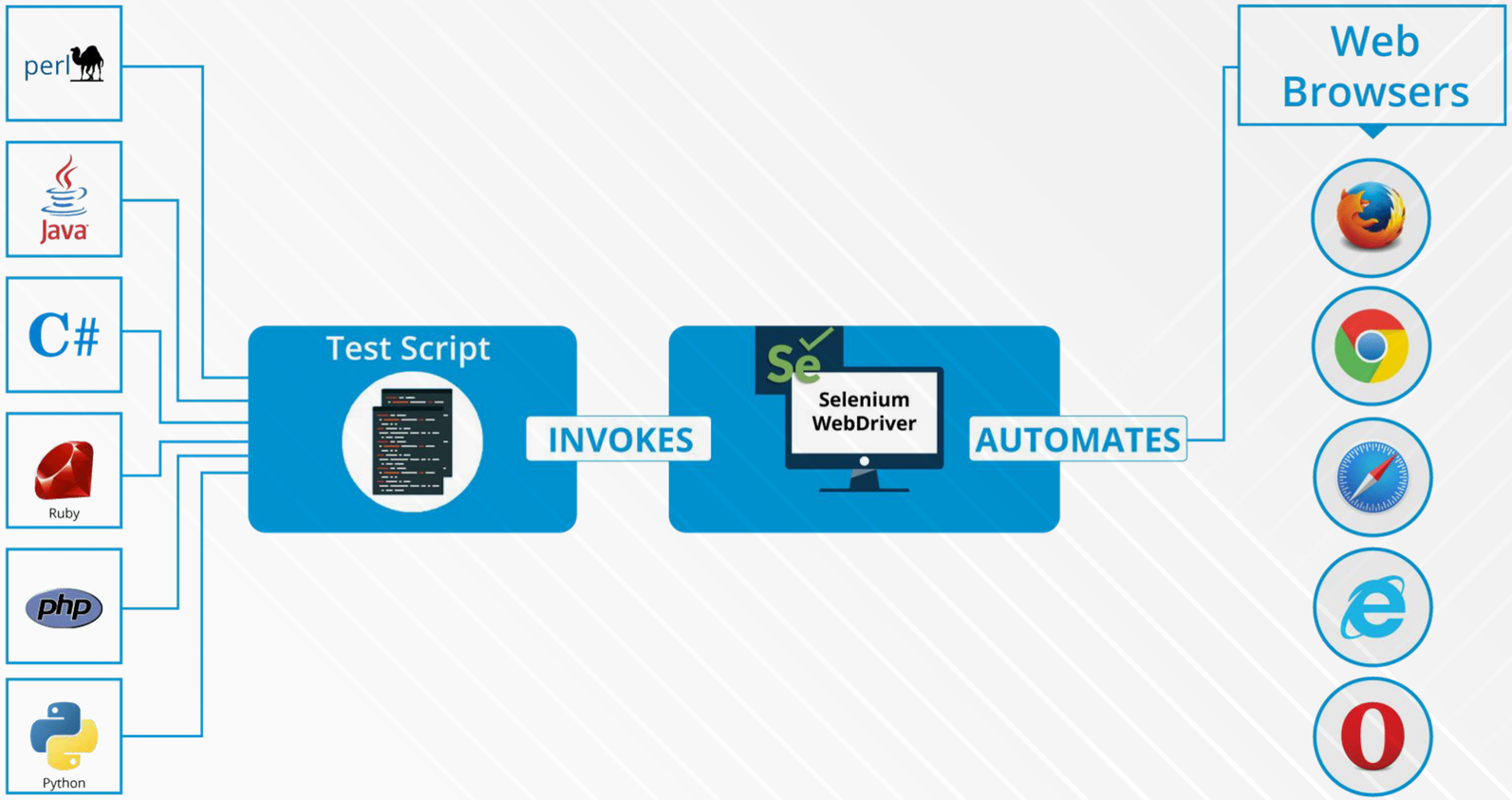
BASE2

- É uma biblioteca **gratuita** e de **código aberto** para **automação de testes funcionais** de **aplicações web** acessadas via browser.

- Disponível para as linguagens: Java, C#, Python, Ruby, JavaScript, PHP e Perl;

Sobre o Selenium WebDriver

É um padrão World Wide Web Consortium (W3C).



4

Criando um novo projeto de testes automatizados

BASE2

Criando um novo projeto de testes automatizados

BASE2

- Abrir o IntelliJ;
- Clicar em “Create New Project”;
- Marcar a opção “Maven”;
- Marcar o check-box “Create from archetype”;
- Marcar uma das opções:
  org.apache.maven.archetypes:maven-apache-quickstart
  org.apache.maven.archetypes:maven-archetype-quickstart
- Clicar em “Next”;
- Preencher “GroupId” e “ArtifactId” com “TreinamentoSelenium”;
- Em “Project Location” selecionar onde ficará a pasta com o projeto;
- Clicar em “Finish”;

Criando um novo projeto de testes automatizados

BASE2

A. Aguardar até que o IntelliJ acabe de carregar o projeto;

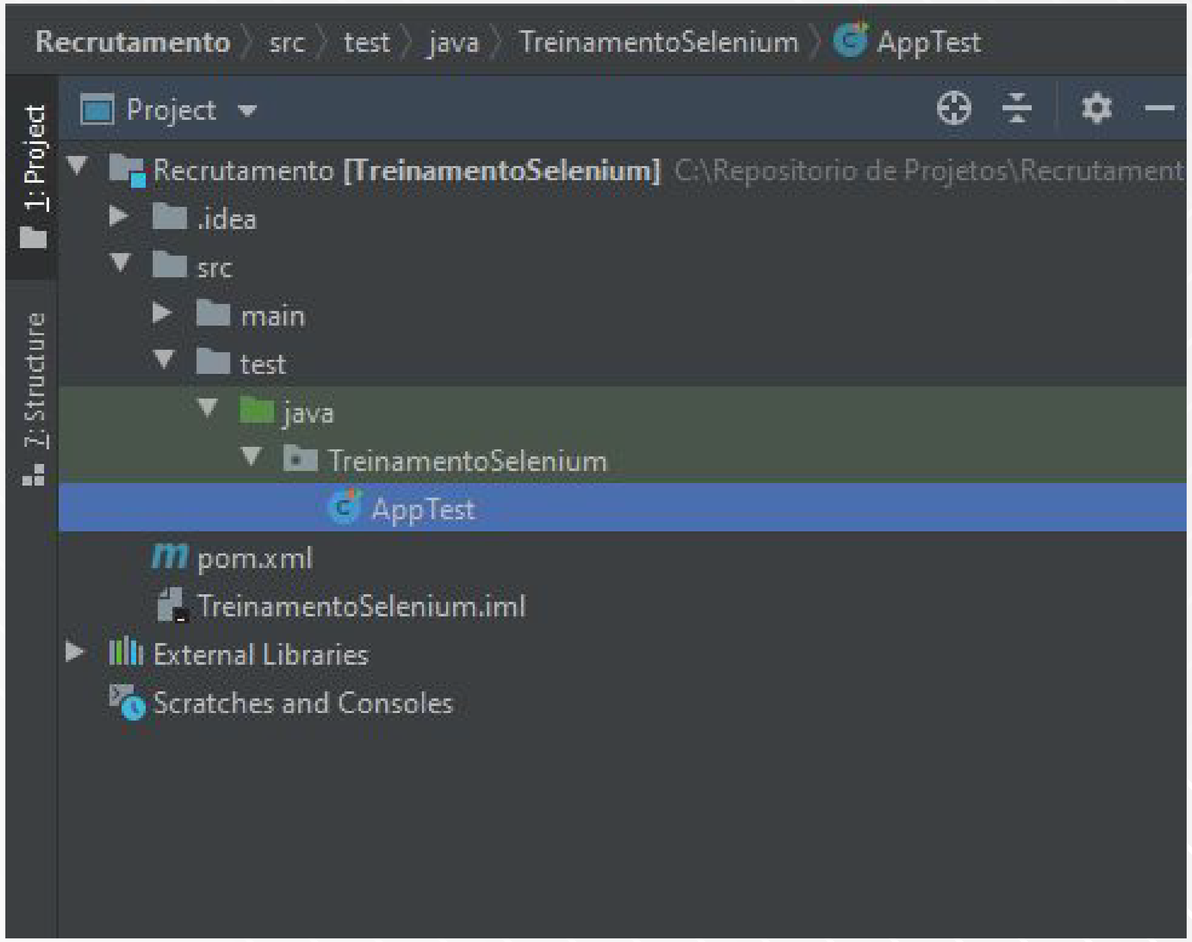
B. Expandir a árvore: TreinamentoSelenium -> src -> test -> java;

C. Clicar com o botão direito do mouse na pasta “java”;

D. Navegar até a opção: Mark Directory as -> Test Sources Root;

Criando um novo projeto de testes automatizados

Após configuração o projeto apresentará a seguinte estrutura:



BASE2

Selenium WebDriver - ChromeDriver

Existem vários navegadores que podem ser utilizados para realizar a automação de testes de interface, iremos escolher o **Google Chrome** e seu respectivo Driver, o Chrome Driver para execução do navegador.

Chrome Driver → **Google Chrome**

Baixar o chromedriver e copiar/arrastar para a pasta raiz do projeto (verificar versão do Google Chrome instalado no seu computador);

https://chromedriver.chromium.org/downloads

Maven:

- Gerenciador de dependências
- Arquivo pom.xml

Configurando dependências:

- **JUnit:** Inserir a referência de dependência da biblioteca do JUnit no arquivo “pom.xml” do Maven;
- **Selenium Java:** Inserir a referência de dependência da biblioteca do Selenium Java no arquivo “pom.xml” do Maven.

5

JUnit

BASE2

Sobre o JUnit

- É um framework de testes para o Java;

- Será usado em conjunto com o Selenium para facilitar a execução dos testes e também aferir os resultados esperados (orquestrador de testes);

- Traz mecanismos visuais para informar se o teste passou ou qual problema de validação foi encontrado;

- OBS: Não é obrigatório o uso do JUnit para implementação dos testes com Selenium Webdriver.

Estrutura básica de um teste do JUnit

BASE2

```
public class JUnitTeste {

    @Test
    public void meuPrimeiroTeste() {

    }

}
```

anotação @Test

método publico

Asserções do JUnit

BASE2

Asserções(assertions):

São métodos de aferição usados para comparar os resultados esperados dos testes com os resultados encontrados nas execuções;

Os tipos mais comuns de asserções usadas em testes automáticos funcionais são:

- **assertTrue:** função que valida o retorno como true
- **assertEquals:** função que compara o resultado esperado com o resultado atual (nesta ordem)

Asserções do JUnit

Exemplo de “assertTrue”:

Criamos uma *string* chamada fruta com o valor “Pera”

```
@Test
public void meuPrimeiroTeste() {
    String fruta = "Pera";

    assertTrue(fruta.equals("Pera"));
}
```

Garantimos que a *string* fruta é uma “Pera”

Asserções do JUnit

BASE2

- AssertTrue:

A String empresa tem o valor “Base2”?
A String cidade tem o valor “Belo Horizonte”?

- No contexto do Selenium WebDriver:

O botão está visível na tela?
O botão está clicável na tela?

Criamos uma *string* chamada fruta com o valor “Pera”

```
@Test
public void meuPrimeiroTeste() {
    String fruta = "Pera";

    assertTrue(fruta.equals("Pera"));
}
```

Garantimos que a *string* fruta é uma “Pera”

Asserções do JUnit

Exemplo de “assertEquals”:

```
@Test
public void meuTerceiroTeste() {
    assertEquals(4, 2+2);
}
```

Resultado Esperado

Resultado Obtido (soma de 2 + 2)

Asserções do JUnit

BASE2

- **Exemplos de assertEquals:**

A soma de 4 e 6 é igual a 10?
A string carro tem o valor Gol?
A String cidade tem o valor “Belo Horizonte”?

```
@Test
public void meuTerceiroTeste() {
    assertEquals(4, 2+2);
}
```

Resultado Esperado

Resultado Obtido (soma de 2 + 2)

- **No contexto do Selenium WebDriver:**

A mensagem de sucesso exibida na tela é igual a mensagem esperada?
O cálculo obteve o resuldo esperado?

6

Hello World!

BASE2

“Hello World”

BASE2

Criar teste “Hello World”

1- Instanciar um novo driver
2- Navegar para a URL
3- Verificar se o título é o esperado
4- Encerrar o driver

```
@Test
public void helloWorld(){
    WebDriver driver = new ChromeDriver();

    driver.navigate().to( s: "http://google.com");

    Assert.assertEquals( expected: "Google", driver.getTitle());

    driver.quit();
}
```

7

Navegação

BASE2

Navegações

Os métodos disponíveis para realizarmos navegação no browser são:

- **navigate().to(string):** navega para uma URL contida em uma string
- **navigate().to(URL):** navega para uma URL do tipo “java.net.URL”
- **navigate().back():** volta para a página anterior do histórico de navegação
- **navigate().forward():** avança para a próxima página do histórico de navegação
- **navigate().refresh():** atualiza a página

8

Fechando o browser

BASE2

Fechando o browser

Os métodos disponíveis para fechar o browser são:

- **driver.close():** fecha a aba atual, se for única fecha o browser;

- **driver.quit():** fecha todas as abas junto com o browser.

9

Obtendo dados do driver

BASE2

Obtendo dados do driver

Os métodos disponíveis para obter dados do driver são:

- **driver.getTitle():** retorna o título da página;
- **driver.getCurrentURL():** retorna URL atual;
- **driver.getPageSource():** retorna código fonte da página (HTML).

10

Localizando elementos

BASE2

Localizando elementos da tela

Os elementos da tela são definidos como “WebElements”

Os métodos que localizam os elementos são:

- **driver.findElement():** encontra um único elemento
  Exemplos: um campo específico, um botão;

- **driver.findElements():** encontra mais de um elemento
  Exemplos: links de uma página, linhas de uma tabela etc;

Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

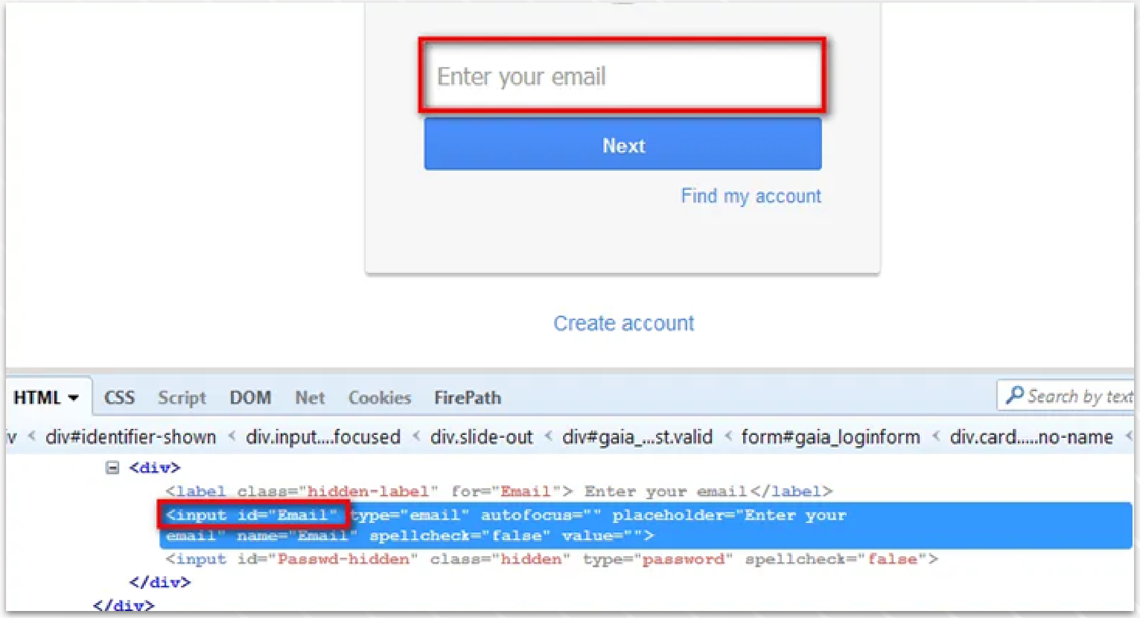
- **By.id()**
- **By.name()**
- **By.tagName()**
- **By.linkText()**
- **By.partialLinkText()**
- **By.className()**
- **By.cssSelector()**
- **By.xpath()**

Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.id():** localiza o elemento pelo atributo “id”

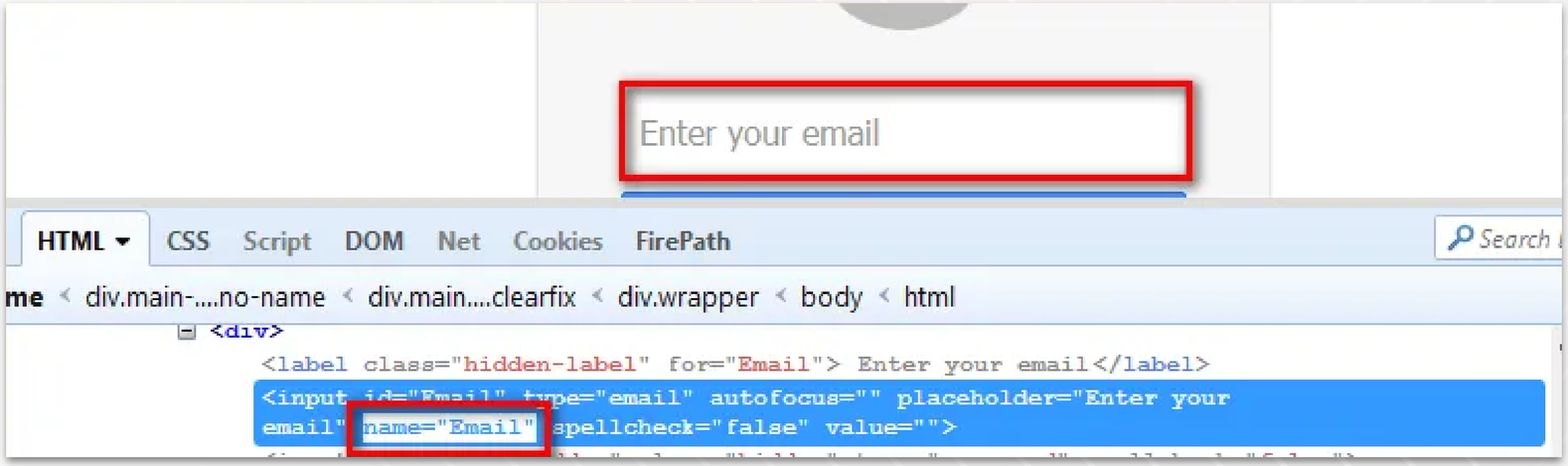


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.name():** localiza o elemento pelo atributo “name”

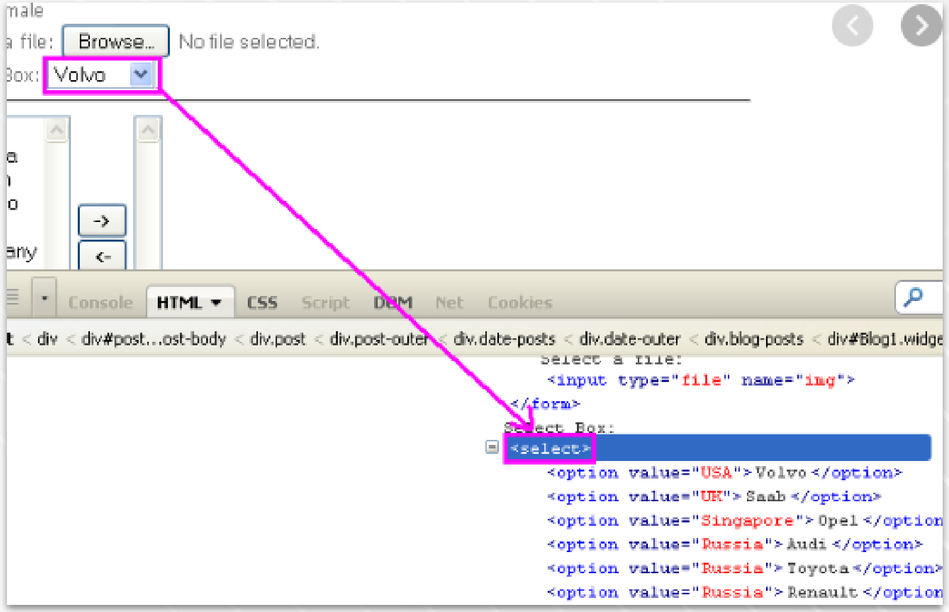


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.tagName():** localiza o elemento pelo nome de uma tag HTML

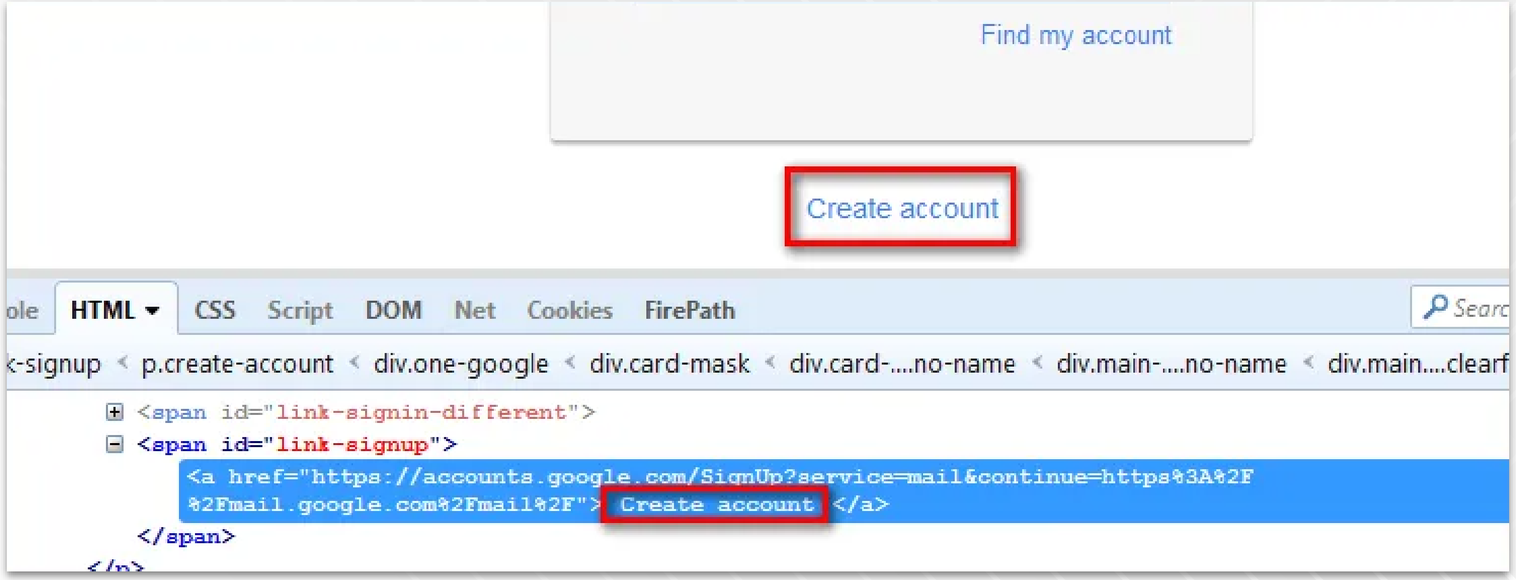


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.linkText():** localiza o elemento pelo texto do link (no caso de elementos do tipo link “<a>”)

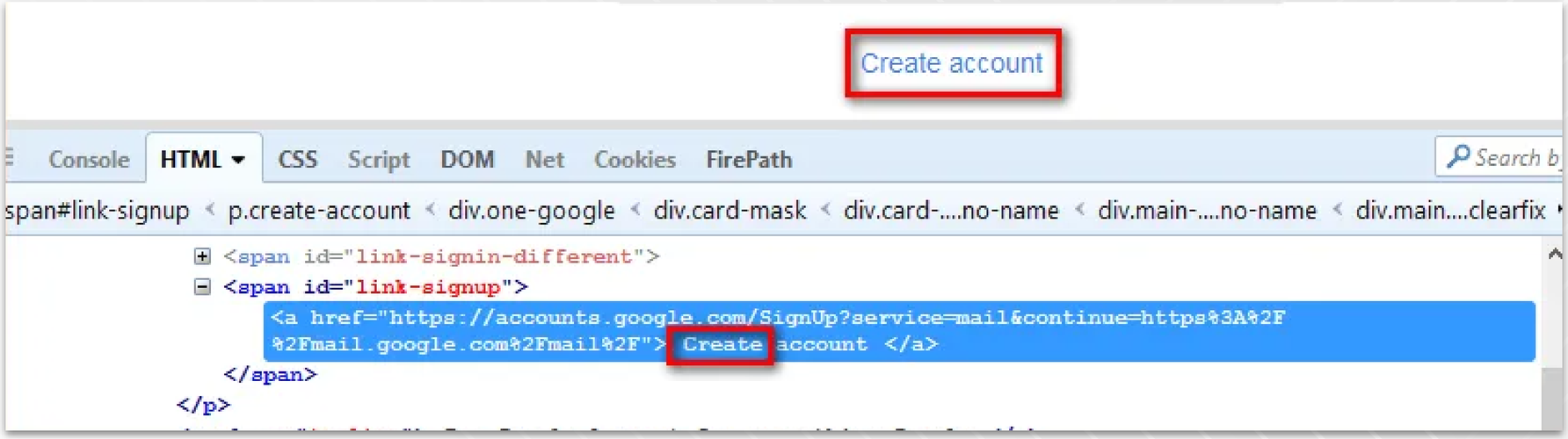


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.partialLinkText():** localiza o elemento por parte do texto do link (elementos do tipo link “<a>”)

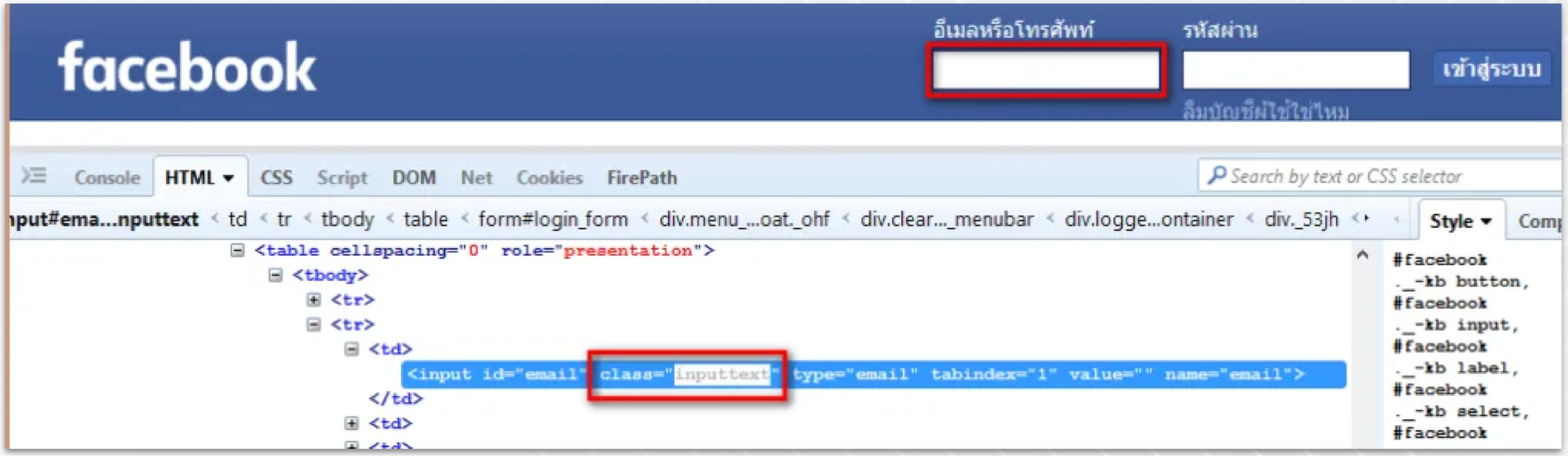


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.className():** localiza o elemento pelo nome da classe do CSS

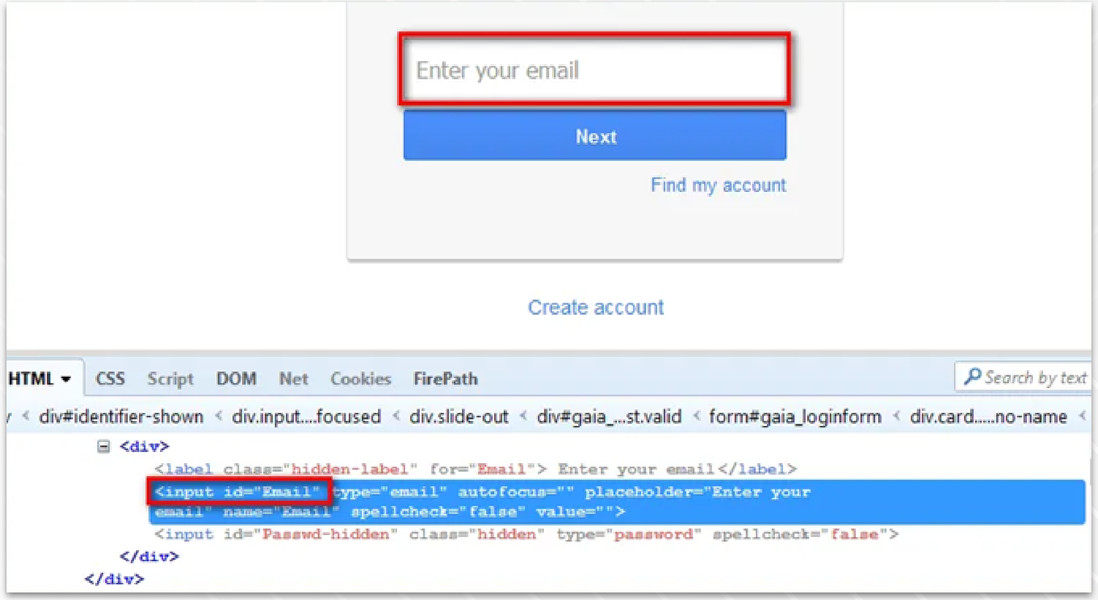


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.cssSelector():** localiza o elemento através de uma expressão de cssSelector

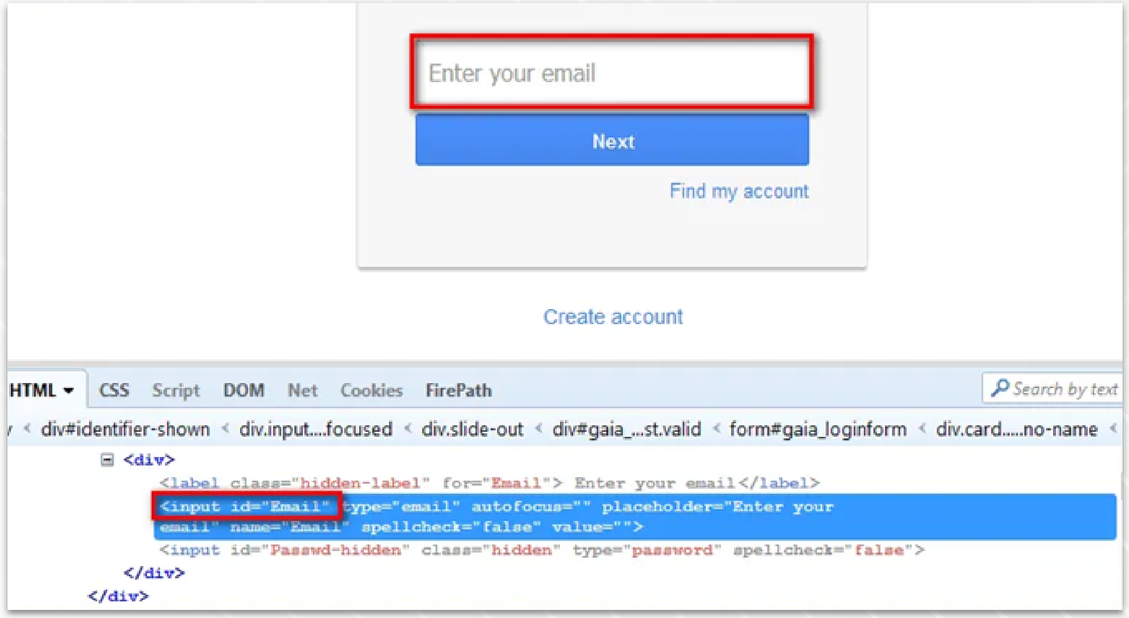


Localizando elementos da tela

Para encontrar um elemento é necessário informar a estratégia de localização (By) para os métodos de localização.

Os tipos de localizadores de elementos são:

- **By.xpath():** localiza o elemento através de uma expressão XPath



Localizando elementos da tela

BASE2

Formas de localização de elementos:

- Inspecionar o html (ctrl+shift+c) + ctrl+ f - xpath ou CssSelector
- Selenium IDE
- Katalon

Sites para treinar:

- https://the-internet.herokuapp.com/

Localizando elementos da tela

BASE2

Localizar elemento → **Gerar o elemento**

Locator (By)

WebElement

```
driver.findElement(By.id("username"));
```

Localizando elementos da tela

BASE2

Localizar elemento → **Gerar o elemento**

Locator (By)

WebElement

```
driver.findElement(By.id("username"));
```

Locator

Localizando elementos da tela

BASE2

Localizar elemento → **Gerar o elemento**

Locator (By)

WebElement

```
driver.findElement(By.id("username"));
```

Estratégia de localização e locator

Localizando elementos da tela

BASE2

Localizar elemento → **Gerar o elemento**

Locator (By)

WebElement

```
driver.findElement(By.id("username"));
```

WebElement

Localizando elementos da tela – CSS Seletores

BASE2

Exemplos:

- driver.findElement(By.**id**(“id”));
- driver.findElement(By.**name**(“name”));
- driver.findElement(By.**tagName**(“div”));
- driver.findElement(By.**linkText**(“Relatorio Mensal”));
- driver.findElement(By.**partialLinkText**(“Mensal”));
- driver.findElement(By.**cssSelector**(“div > .even”));
- driver.findElement(By.**className**(“error”));
- driver.findElement(By.**xpath**(“//input[@class=‘required’]”));

Localizando elementos da tela – CSS

BASE2

- Localiza o elemento através do seu estilo;

- Alguns selectors não são suportados pelos Internet Explorers 6, 7 e 8;

- Referência de CSS Selectors:
https://www.w3schools.com/cssref/css_selectors.asp

Expressões mais comuns - Site Google

- **elemento**: Localiza o elemento. Ex: div
- **#id:** Localiza o elemento através do seu id. Ex: #footcnt
- **classe:** Localiza o elemento através da(s) classe na tag class. Ex: .content
- **elemento[atributo='valor']:** Localiza um elemento pelo valor de um atributo. Ex: div[id='main']
- **elemento > elemento:** Localiza o próximo elemento baseado no anterior. Ex: div[id='main']>span
- **elemento[atributo^='valor']:** Localiza o elemento que inicia com o valor de um atributo. Ex: div[class^='jhp']
- **elemento[atributo$='valor']:** Localiza o elemento que termina com o valor de um atributo. Ex: div[class$='big']
- **elemento[atributo*='valor']:** Localiza o elemento que contém o valor de um atributo. Ex: input[id*='mae']

Localizando elementos da tela - XPaths

BASE2

Exemplos:

- driver.findElement(By.cssSelector(“input”))
- driver.findElement(By.cssSelector(“#email”));
- driver.findElement(By.cssSelector(“.error”));
- driver.findElement(By.cssSelector(“input[value=‘Submeter’]”));
- driver.findElement(By.cssSelector(“div > p”));
- driver.findElement(By.cssSelector(“a[href^=‘https’]”));
- driver.findElement(By.cssSelector(“a[href$=‘.pdf’]”));
- driver.findElement(By.cssSelector(“a[href*=‘docs’]”));

Localizando elementos da tela - XPaths

BASE2

- É uma linguagem de navegação em arquivos XML;

- HTML é estruturado da mesma forma que um XML, por isso também podemos XPaths para localizar seus elementos;

- Localiza o elemento através do “caminho” para chegar até ele dentro da árvore do HTML;

- Aprenda mais sobre XPath:
https://www.w3schools.com/xml/xpath_intro.asp

https://www.guru99.com/using-contains-sbiling-ancestor-to-find-element-in-selenium.html

Localizando elementos da tela - XPaths

BASE2

Expressões mais comuns:

- **//elemento:** Localiza o elemento. Ex: //div
- **//elemento[@atributo]:** Localiza o elemento que tenha o atributo descrito
- **//elemento[@atributo='valor']:** Localiza o elemento que tenha o seguinte valor do atributo descrito
- **//elemento1/elemento2:** Localiza o elemento 2 através do elemento 1 (é filho)
- **//elemento[numero]:** Localiza o elemento pelo seu numero/posição
- **(//elemento)[numero]:** Localiza o elemento baseando na sua expressão e seu número
- **//elemento[contains(text(),'valor')]:** Localiza o elemento pelo texto que um elemento possui

Localizando elementos da tela

BASE2

Exemplos

- driver.findElement(By.xpath(“//input”));
- driver.findElement(By.xpath(“//input[@placeholder]”));
- ~~driver.findElement(By.xpath(“/html/body/div[3]/form/table/tbody/tr[2]/td[2]/input”));~~
- ~~driver.findElement(By.xpath(“//div/ul/li/a”));~~
- ~~driver.findElement(By.xpath(“//input[2]”));~~
- driver.findElement(By.xpath(“//input[starts-with(@id,’email’)]”));
- driver.findElement(By.xpath(“//input[contains(text(),’Clique aqui’)]”));

11

Realizando ações em elementos

BASE2

Realizando ações em elementos

BASE2

Um WebElement possui métodos para sua manipulação:

- **click():** simula a ação de um clique;
- **clear():** limpa o conteúdo de um elemento;
- **sendKeys(string):** preenche um campo de texto ou envia eventos do teclado.

driver.findElement(By.id("username")).ACTION;

Manipulação

Realizando ações em elementos

Para trabalhar com combobox precisamos utilizar a classe “Select”

driver.findElement(By.id("exemplo"))

Select combo = new **Select(WebElement);**

Os principais métodos para sua manipulação são:

combo.selectByIndex(index): Seleciona um item pelo índice;
combo.selectByValue(value): Seleciona um item pelo valor ;
combo.selectByVisibleText(text): Seleciona um item pelo texto visível.

Realizando ações em elementos

BASE2

Valor da combo

```
<select id="estados" name="estados">
0 <option value="AC">Acre</option>
1 <option value="AL">Alagoas</option>
2 <option value="AM">Amazonas</option>
3 <option value="AP">Amapá</option>
4 <option value="BA">Bahia</option>
5 <option value="CE">Ceará</option>
6 <option value="DF">Distrito Federal</option>
7 <option value="ES">Espirito Santo</option>
7 <option value="GO">Goiás</option>
8
9
```

Índice da combo

Texto visível da combo

12

Obtendo dados dos elementos

BASE2

Obtendo dados dos elementos

BASE2

Os métodos disponíveis para obter dados dos elementos (WebElement) são:

getText(): retorna o texto do elemento. Este texto está entre as tags do próprio elemento. Ex: texto
getAttribute(): Retorna o valor de um atributo de um elemento.
getTagName(): Retorna o nome da tag HTML do elemento
isEnable(): Retorna se o elemento está habilitado (boolean: true ou false)
isSelected(): Retorna se o elemento está selecionado (boolean: true ou false)
isDisplayed(): Retorna se o elemento é exibido (boolean: true ou false)
getSize(): Retorna o tamanho do elemento
getLocation(): Retorna a posição do elemento na tela (x e y)
getCSSValue(): Retorna o valor de alguma propriedade CSS do elemento

driver.findElement(By.id("exemplo")).ACTION;

13

Sincronização

BASE2

Sincronização

Quando uma página é carregada não é necessário implementar uma espera (Requisição HTTP Browser).

Por default o Selenium só executa a próxima ação após o carregamento total da página;

Exemplo de uso em Single Page Application ou Aplicativo de página única (SPA).

Existem duas formas básicas de sincronização (esperas): Explícitas e Implícitas;

Sincronização

Esperas implícitas?

- Define um "tempo limite implícito de espera"
- Se um elemento não puder ser encontrado instantaneamente, o servidor continuará tentando encontrá-lo até o tempo limite especificado.
- driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS)

Esperas explícitas?

- Espera até determinada condição ser atendida, atendeu? Segue o teste!
- O localizador está presente?
- Elemento é clicável?
- Elemento está ativo?

Então devemos usar qual?

BASE2

Explícitas!!!

A forma mais usada e recomendada para se tratar esperas por chamadas assíncronas são as esperas explícitas! A documentação do Selenium recomenda ela.

Obtemos um controle muito mais refinado e somos capazes de ensinar nosso script de teste como aguardar a condição correta em nosso aplicativo antes de prosseguir.

Sincronização

Para as esperas explícitas fazemos uso da classe “WebDriverWait”:

```
WebDriverWait wait = new WebDriverWait(driver, 30);
```

Necessita do driver/browser atual

Tempo de espera

Sincronização

Os métodos disponíveis do WebDriverWait são:

- **until(boolean):** Espera o até que determinada condição retorne “true”;
- **pollingEvery(duração, unidade):** Configura de quanto em quanto tempo a condição será verificada;
- **withMessage(string):** Define uma mensagem para ser exibida no log caso ocorra timeout;
- **ignoring(tipo exceção):** Ignora uma exceção que possa ocorrer durante a espera;
- **withTimeout(tiemout):** Sobrescreve o timeout;

Sincronização

BASE2

Para um elemento ser clicável ele tem que ter no mínimo 1px de altura, 1px de largura, estar habilitado e visível.

O método de uso mais comum é o **until();**

Existe uma classe estática chamada "*ExpectedConditions*" que contém algumas condições pré-determinadas e podem ser usadas como parâmetro do método until();

Os métodos da classe "ExpectedConditions" podem receber como parâmetro um elemento (WebElement) ou um localizador (By) .

```
wait.until(ExpectedConditions.presenceOfElementLocated(By);
```

O localizador está presente?

```
wait.until(ExpectedConditions.visibilityOf(WebElement));
```

Elemento está visivel?

```
wait.until(ExpectedConditions.elementToBeClickable(WebElement));
```

Elemento está clicável?

14

Prática

BASE2

Prática

Vamos praticar o que aprendemos

- Iremos enviar a apresentação em pdf para consulta
- Iremos praticar com o sistema Mantis - https://mantis-prova.base2.com.br/
- Cada participante terá seu usuário
- Em caso de dúvidas abra seu microfone ou envie no chat uma msg
- Vamos dividir em duas salas
- Sala principal
- Sala dúvidas
- Ao final da prática, cada um deverá enviar o código para o instrutor.

Prática

Implementar o seguinte caso de teste para o Mantis

- Cadastrar nova ocorrência com sucesso:
    Efetuar login no sistema
    Clicar em “criar tarefa”
    Selecionar “Categoria”
    Selecionar “Frequência”
    Selecionar “Gravidade”
    Preencher “Resumo”
    Preencher ‘Descrição”
    Preencher “Passos para reproduzir”
    Clicar em “Criar nova tarefa”
    Verificar se é exibida a mensagem “Operação realizada com sucesso”
    Verificar se a ocorrência foi salva com os dados informados

Dúvidas

Muito Obrigado.
Boa sorte!

BASE2