

AUTOMAÇÃO DE TESTES – SELENIUM WEBDRIVER

Testes de Software

Equipe:

Fernanda

Rogério

Lucas

ROTEIRO

- Introdução
- Download da ferramenta
- Configuração
- Conteúdo teórico
- Exemplos

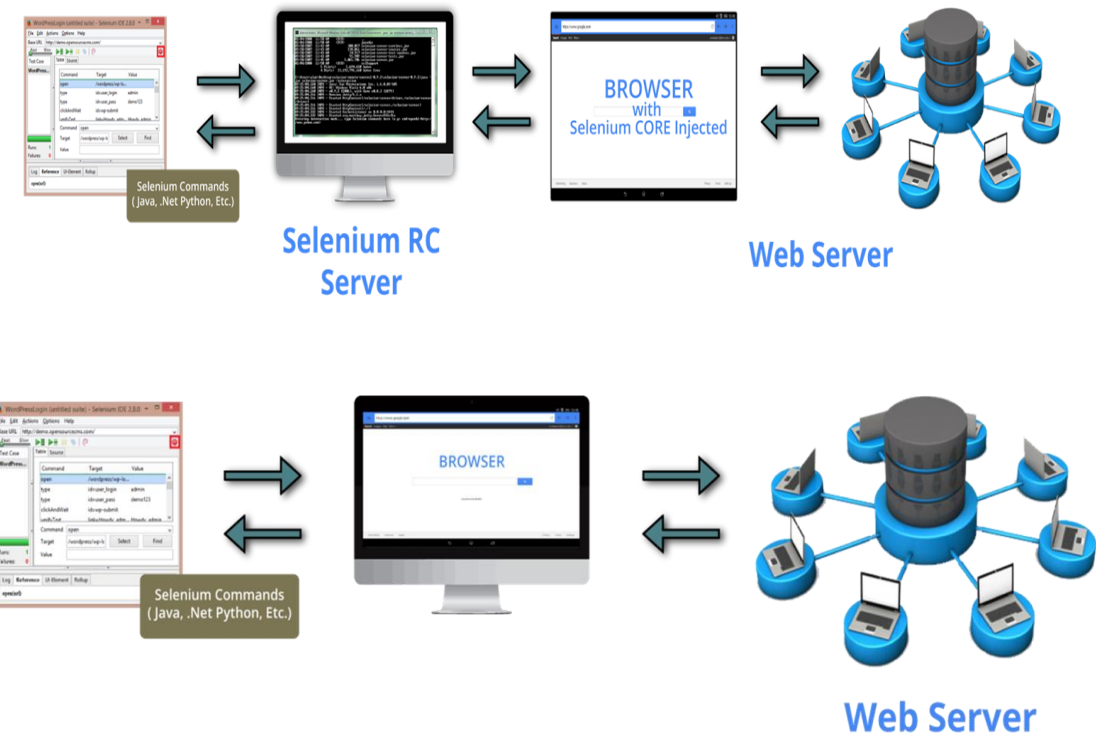
INTRODUÇÃO

- O Selenium Webdriver é uma ferramenta de código aberto usada para realizar automação de testes em aplicativos web.
- O Selenium refere ao Acceptance Testing (ou funcional testing) que envolve realizar testes em um sistema finalizado
- Conduz um navegador de forma nativa (como um usuário).
- Foi projetado para fornecer uma interface de programação mais simples e concisa.



INTRODUÇÃO

- A arquitetura do WebDriver é mais simples que a do Selenium-RC.
- O WebDriver é mais rápido que o Selenium-RC.
- Ele interage com os elementos da página de maneira mais realista.



INTRODUÇÃO

- Faz chamadas diretas ao navegador e todo o script de teste é executado dessa maneira, utilizando o suporte e os recursos dos próprios navegadores para automação.

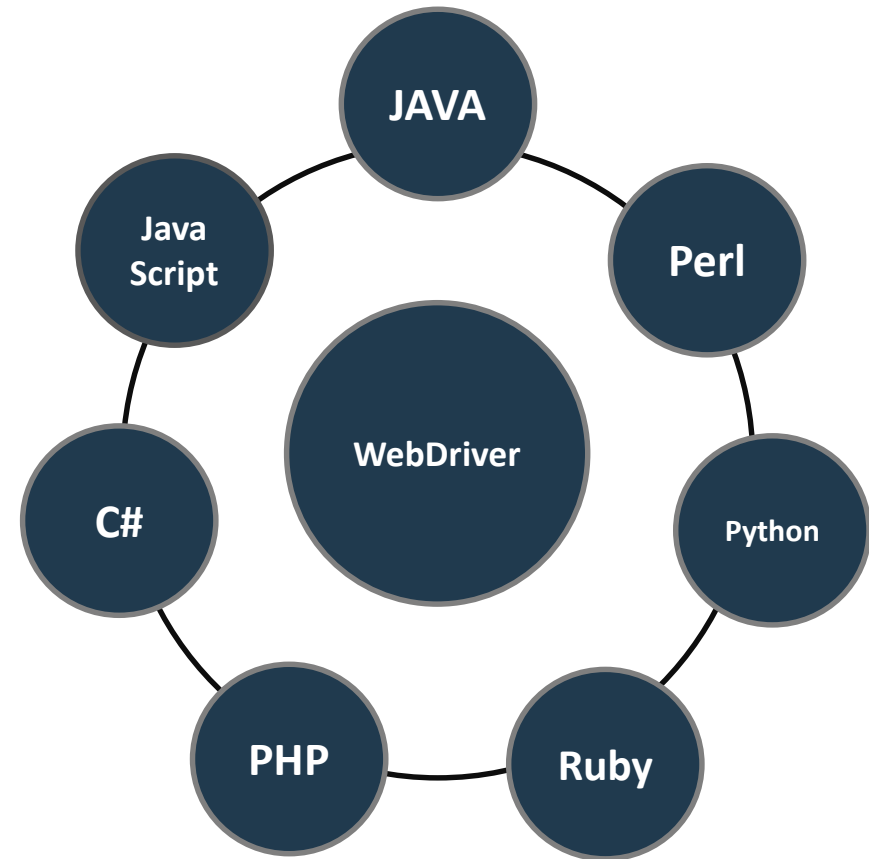


INTRODUÇÃO

Suporte:



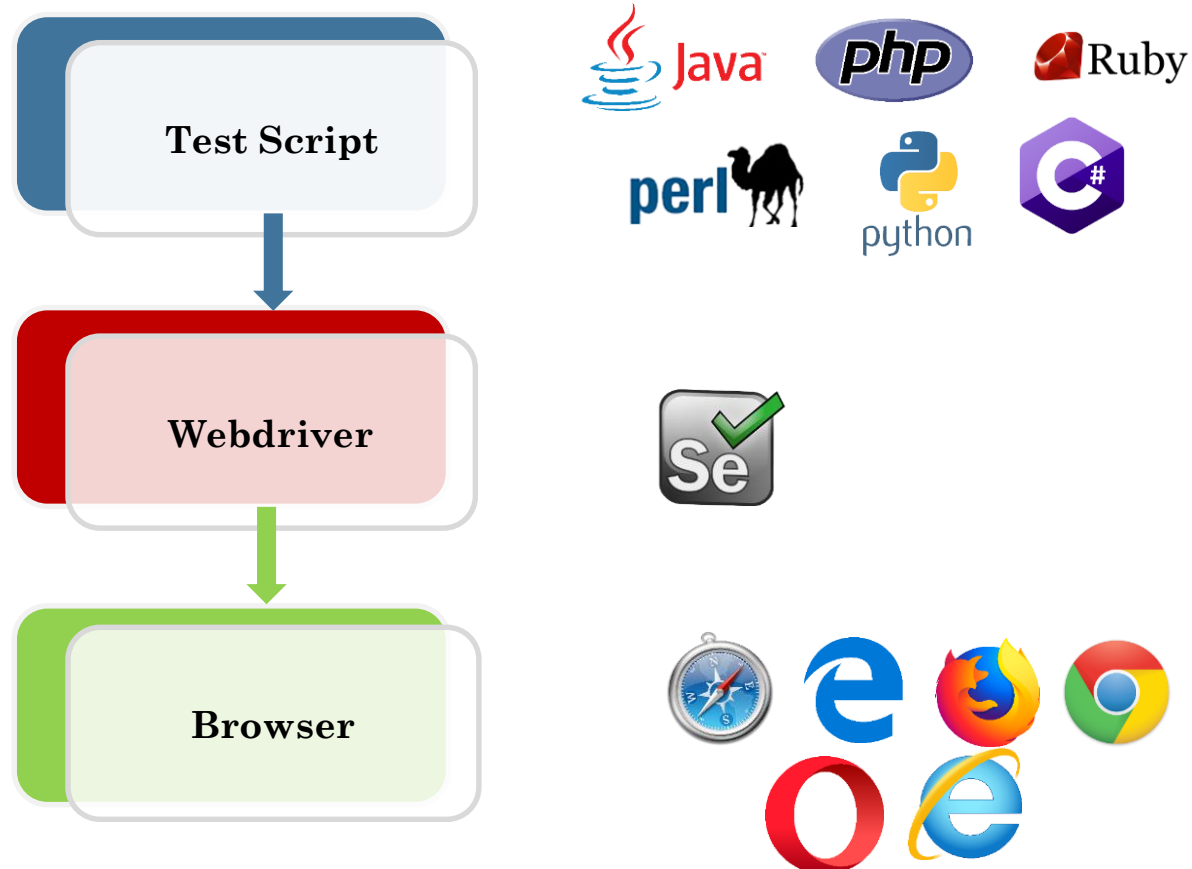
Browsers



Linguagens

INTRODUÇÃO

Funcionamento:



Para realização dos testes, cada uma dessas três partes necessitam uma da outra.

DOWNLOAD DA FERRAMENTA

Há duas maneiras de Obter a biblioteca da ferramenta:

- 1 – Baixando a biblioteca pelo site do Selenium, e adicionando ao seu projeto.
- 2 – Adicionando a biblioteca Maven do Selenium WebDriver.

Estão a mostra logo
a frente



DOWNLOAD DA FERRAMENTA

1 - Baixando biblioteca

<https://www.seleniumhq.org/download/>

Language	Client Version	Release Date			
Java	3.14.0	2018-08-02	<u>Download</u>	<u>Change log</u>	<u>Javadoc</u>
C#	3.14.0	2018-08-02	<u>Download</u>	<u>Change log</u>	<u>API docs</u>
Ruby	3.14.0	2018-08-03	<u>Download</u>	<u>Change log</u>	<u>API docs</u>
Python	3.14.0	2018-08-02	<u>Download</u>	<u>Change log</u>	<u>API docs</u>
Javascript (Node)	4.0.0-alpha.1	2018-01-13	<u>Download</u>	<u>Change log</u>	<u>API docs</u>

DOWNLOAD DA FERRAMENTA

2 - Adicionando biblioteca maven do Selenium Webdriver:

- De acordo com o site do Selenium, a maneira mais fácil de configurar um projeto Java do Selenium 2.0 é usar o Maven.
- O Maven baixará as ligações do Java e todas as suas dependências, e criará o projeto para você, usando um arquivo pom.xml.

```
<dependency>  
  <groupId>org.seleniumhq.selenium</groupId>  
  <artifactId>selenium-java</artifactId>  
  <version>3.11.0</version>  
</dependency>
```

DOWNLOAD DA FERRAMENTA

Adicionando biblioteca Maven do Junit:

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.12</version>  
  <scope>test</scope>  
</dependency>
```

DOWNLOAD DA FERRAMENTA

Baixaremos o driver do navegador Chrome:






<https://www.seleniumhq.org/download/>

Third Party Browser Drivers **NOT DEVELOPED** by seleniumhq

Browser

Mozilla GeckoDriver	0.20.0	change log	issue tracker	Implementation Status	Released 2018-03-08
Google Chrome Driver	2.36	change log	issue tracker	selenium wiki page	Released 2018-03-02
Opera	2.29		issue tracker	selenium wiki page	Released 2017-06-27

Index of /2.36/

	Name	Last modified
	Parent Directory	
	chromedriver_linux64.zip	2018-03-02 10:19:20
	chromedriver_mac64.zip	2018-03-02 09:17:32
	chromedriver_win32.zip	2018-03-02 11:24:21
	notes.txt	2018-03-14 22:10:46







DOWNLOAD DA FERRAMENTA

Baixaremos o driver do navegador Mozilla Firefox:

<https://www.seleniumhq.org/download/>

Third Party Browser Drivers **NOT DEVELOPED** by seleniumhq

Browser						
Mozilla GeckoDriver	0.20.0	change log	issue tracker	Implementation Status	Released 2018-03-08	 geckodriver-v0.20.0-linux64.tar.gz
Google Chrome Driver	2.36	change log	issue tracker	selenium wiki page	Released 2018-03-02	 geckodriver-v0.20.0-macos.tar.gz
Opera	2.29		issue tracker	selenium wiki page	Released 2017-06-27	 geckodriver-v0.20.0-win32.zip
						 geckodriver-v0.20.0-win64.zip

CONTEÚDO TEÓRICO

API:

- A API Selenium WebDriver tem como principal objetivo **automatizar ações do navegador**, tais como:
 - Preenchimento de campos de Texto;



Abra uma conta

É gratuito e sempre será.

Francisgleydson Jacinto

francisglaydsonjacinto@gmail.com

francisglaydsonjacinto@gmail.com

••••••••

CONTEÚDO TEÓRICO

API:

- Seleções em menus dropdown;



A screenshot of a web form titled "Data de nascimento" (Birth date). It features three dropdown menus for selecting the day, month, and year. The first dropdown shows the number "9", the second shows the month "Abr" (April), and the third shows the year "1996". Each dropdown has a small downward arrow icon to its right.

- Submits de formulários;



A screenshot of a green rectangular button with rounded corners and a white border. The button contains the text "Criar conta" (Create account) in white, centered.

- Varredura de dados em elementos;
- HTML etc.

CONTEÚDO TEÓRICO

Interfaces:

- **Interface WebDriver:** Podemos dizer que a interface WebDriver é a mais importante de um projeto Selenium WebDriver. Nela, temos os métodos que controlam o navegador, selecionam elementos de páginas HTML, etc.
- **Interface WebElement:** Como o próprio nome já sugere, o tipo WebElement serve para armazenar um elemento web, ou seja, um componente HTML de uma tela.

CONTEÚDO TEÓRICO

Interface WebElement:

- A WebElement trabalha na maioria das vezes basicamente com a utilização da tag **id** do html e é através dela que a mesma obtém todas os dados necessários para a realização de suas responsabilidades.
- Além da tag **id** também podemos utilizar as tags: **className**, **cssSelector**, **linkText**, **name**, **partialLinkText**, **tagName** e **xpath**.

CONTEÚDO TEÓRICO

Interface WebElement:

- **id** - Especifica um identificador único para o elemento.
- **className** - Especifica o nome da classe de um elemento.
- **cssSelector** - Seleciona o css do elemento.
- **linkText** - Especifica o link do elemento.
- **name** - Especifica o nome do elemento.

CONTEÚDO TEÓRICO

Interface WebElement:

- **partialLinkText** - Especifica o link parcial do elemento.
- **tagName** - Especifica o nome da tag do elemento.
- **xpath** - Especifica o caminho do elemento.
- Essa interface possui vários métodos que a possibilitam manusear o objeto e seus atributos. Mais adiante veremos os que são considerados mais utilizados.

CONTEÚDO TEÓRICO

Métodos do WebDriver:

- Os dois métodos mais comuns e importantes da interface WebDriver são o **get()** e o **findElement()**. Veja esses e outros métodos logo abaixo:
 - **driver.close();**
Fecha a janela corrente.
 - **driver.findElement(By by);**
Encontra o primeiro elemento de uma tela HTML através de um dado argumento.
 - **driver.findElements(By by);**
Encontra todos os elementos de uma tela HTML através de um dado argumento.

CONTEÚDO TEÓRICO

Métodos do WebDriver:

- **driver.get();**

Abre uma nova URL no navegador.

- **driver.getCurrentUrl();**

Retorna uma string que contém a URL aberta pelo navegador.

- **driver.getPageSource();**

Retorna o código fonte da última página aberta pelo navegador.

CONTEÚDO TEÓRICO

Métodos do WebDriver:

- **driver.getTitle();**

Retorna o título da página aberta pelo navegador.

- **driver.getWindowHandle();**

Retorna um identificador da janela em questão.

- **driver.getWindowHandles();**

Retorna identificadores que podem ser utilizados para movimentação entre janelas.

CONTEÚDO TEÓRICO

Métodos do WebDriver:

- **driver.switchTo();**

Envia comandos futuros para uma janela (ou frame) diferente.

- **driver.manage();**

Permite gerenciar cookies do navegador, logs, timeouts etc.

- **driver.navigate();**

Abstração que permite acessar o histórico e navegar para uma determinada URL.

CONTEÚDO TEÓRICO

Métodos do WebDriver:

- **driver.quit();**

Fecha a instância do Selenium WebDriver e todas os navegadores associados.

CONTEÚDO TEÓRICO

Principais métodos do WebElement:

- **elemento.clear();**

Limpa o conteúdo do elemento.

- **elemento.click();**

Clica e, conseqüentemente, muda o foco da tela para o elemento.

- **elemento.getAttribute();**

Retorna o valor do atributo passado como parâmetro de dado elemento.

CONTEÚDO TEÓRICO

Principais métodos do WebElement:

- **elemento.getCssValue();**

Retorna o valor de uma propriedade CSS passada como parâmetro de dado elemento.

- **elemento.getLocation();**

Retorna o ponto da tela do canto superior esquerdo de dado elemento web.

- **elemento.getSize();**

Retorna a dimensão do elemento (largura e altura).

CONTEÚDO TEÓRICO

Principais métodos do WebElement:

- **elemento.tagName();**

Retorna o nome da tag HTML de dado elemento.

- **elemento.getText();**

Retorna o texto presente dentro do elemento.

- **elemento.isDisplayed();**

Retorna verdadeiro ou falso se dado elemento estiver visível ou não na tela.

CONTEÚDO TEÓRICO

Principais métodos do WebElement:

- **elemento.isEnabled();**

Retorna verdadeiro ou falso se dado elemento estiver ativo ou não na tela.

- **elemento.isSelected();**

Retorna verdadeiro ou falso se dado elemento estiver selecionado ou não na tela.

- **elemento.sendKeys();**

Insere caracteres num determinado elemento da tela.

CONTEÚDO TEÓRICO

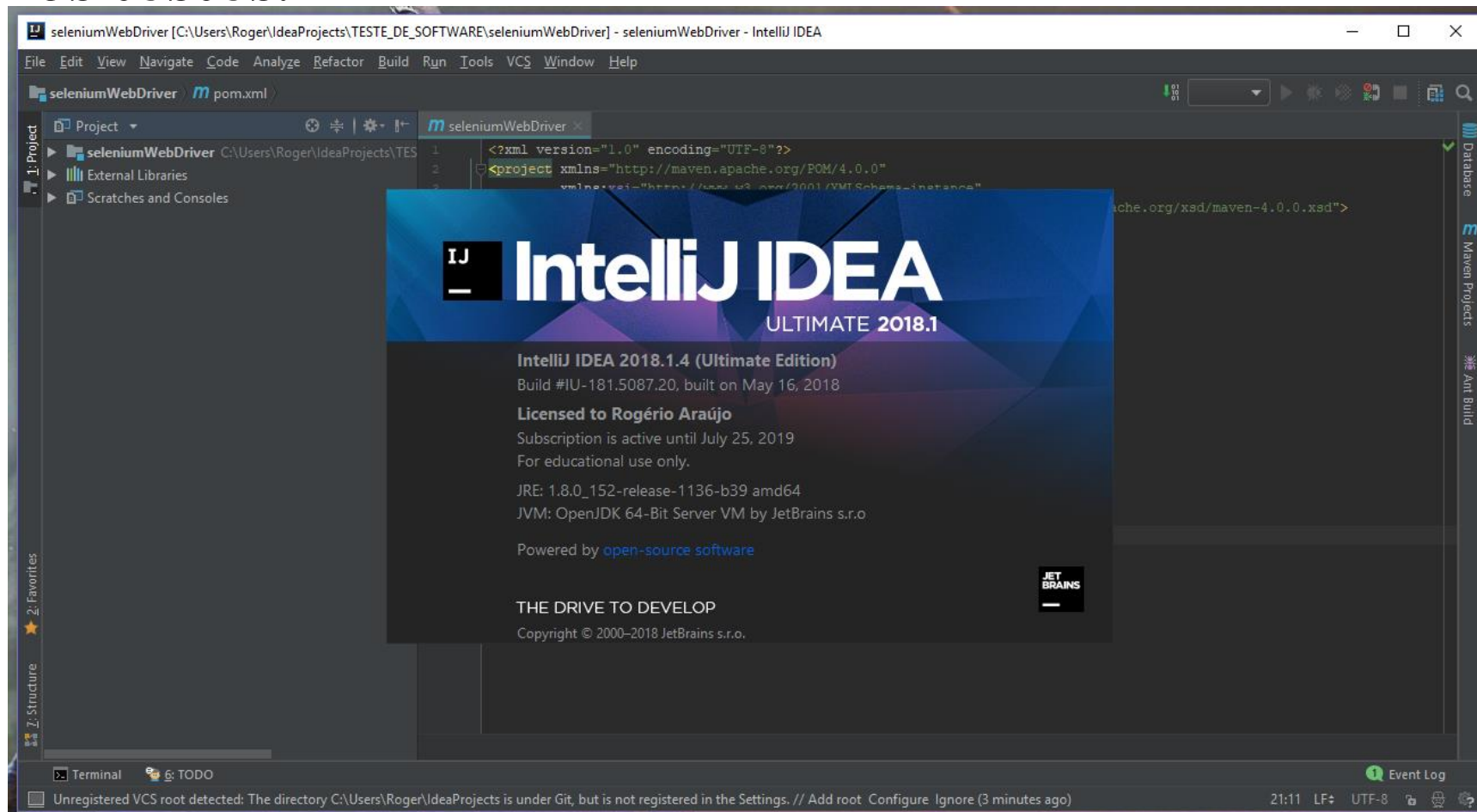
Principais métodos do WebElement:

- **elemento.submit();**

Envia dados para o servidor se o elemento em questão for um formulário.

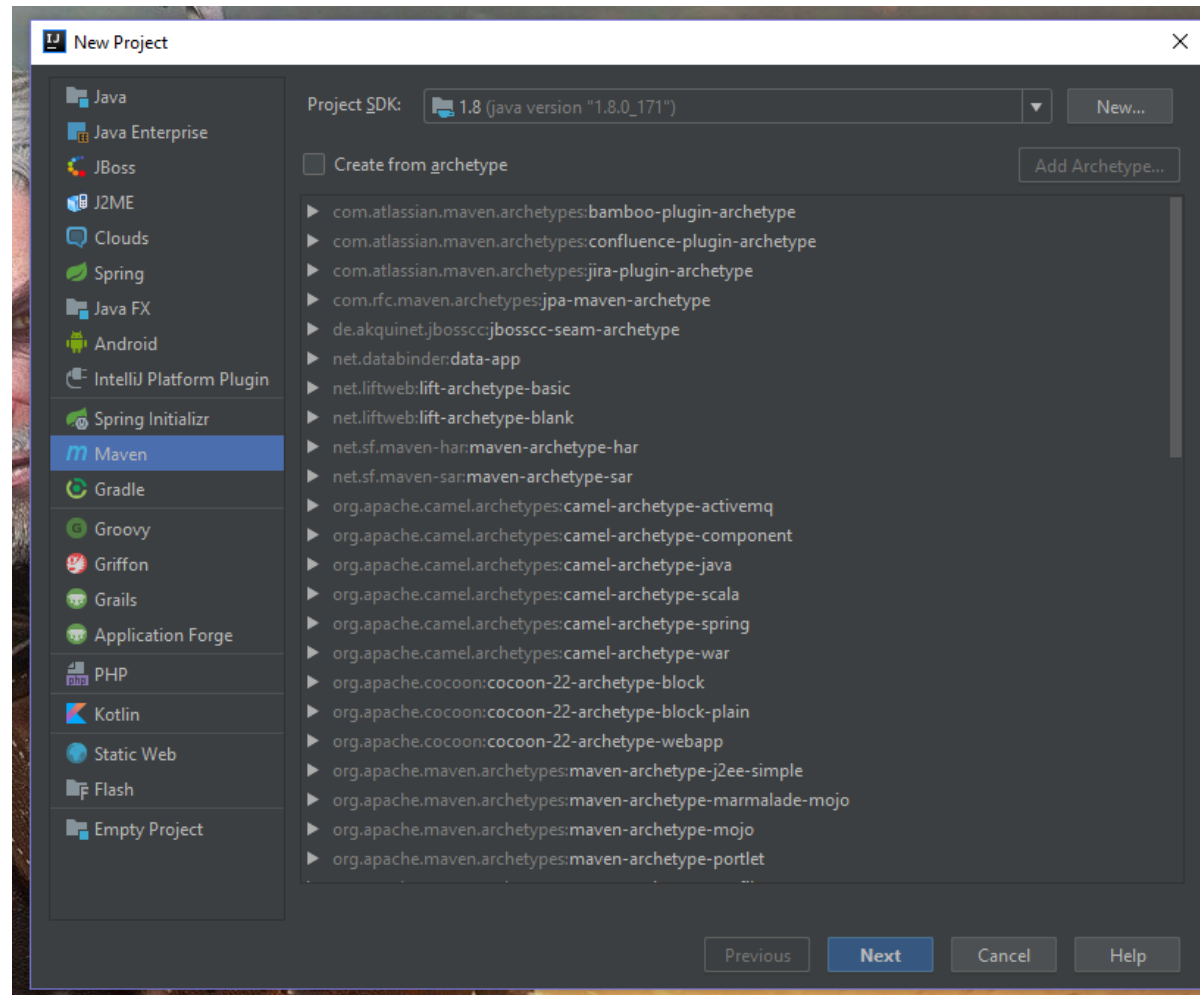
EXEMPLOS

Será utilizada a IDE, IntelliJ IDEA Ultimate 2018.1 para realizar os testes.



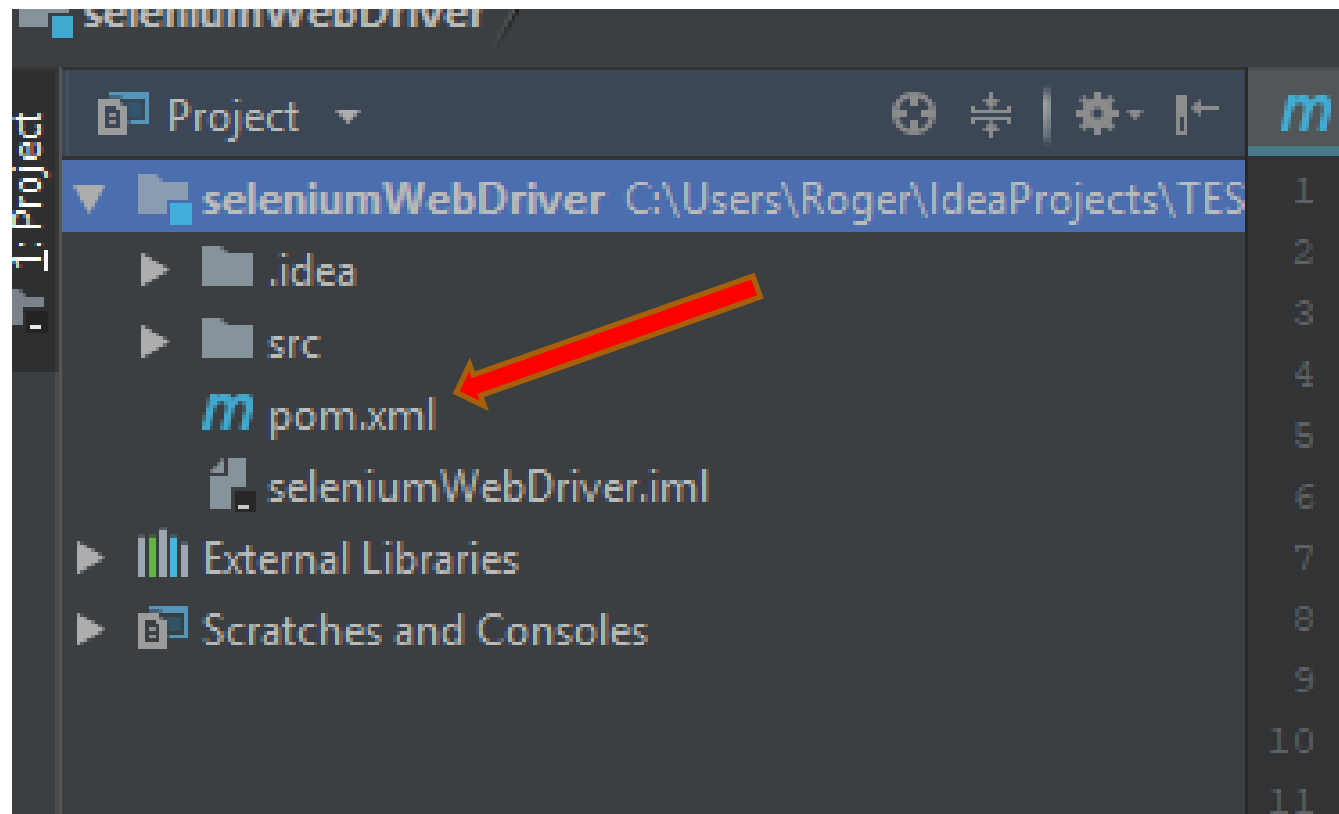
EXEMPLOS

Vamos criar um novo projeto Maven:



EXEMPLOS

Vamos modificar o arquivo pom.xml:



EXEMPLOS

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.14.0</version>
  </dependency>
</dependencies>
```

Adicionando as dependências do Selenium e JUnit

EXEMPLO

○ Exemplo 1:

Faremos um teste para verificar se o logo da Google está visível:

- ✓ CRIAR UM PROJETO MAVEN
- ✓ IMPORTAR AS DUAS BIBLIOTECAS (JUnit e Selenium WebDriver)
- ✓ GUARDAR O ARQUIVO **chromedriver.exe** EM ALGUM LUGAR DE SUA PREFERENCIA
- ✓ CRIAR SCRIPT DE TESTE **TesteLogoGoogle.java** OU UM NOME DE SUA PREFERENCIA

EXEMPLO

○ Exemplo 1:

```
m seleniumWebDriver x TesteLogoGoogle.java x
13 ▶ public class TesteLogoGoogle {
14     //Cria uma variável do tipo WebDriver
15     private static WebDriver driver;
16
17     public TesteLogoGoogle() {
18     }
19
20     @Before
21     public void Conectar() {
22
23         //Editando as propriedades da localização do driver
24         System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
25         //Inicializando a variável
26         driver = new ChromeDriver();
27
28     }
29
30     @After
31     public void Encerrar() {
32
33         driver.close();
34     }
35
36
37     @Test
38     ▶ public void testeLogoGoogle() {
39
40         //Acessar a página do google
41         driver.get("https://www.google.com.br/");
42         //Encontrando elemento pelo Id
43         WebElement logo = driver.findElement(By.id("hplogo"));
44         //Verificando se o logo da google está visível no display
45         assertTrue(logo.isDisplayed());
46     }
47 }
```

EXEMPLO

○ Exemplo 2:

Nesse segundo teste vamos realizar uma soma em uma calculadora online e verificar se o resultado está correto.

LINK DO SITE PARA O TESTE

<https://web2.0calc.com/>

EXEMPLO

○ Exemplo 2:

```
m seleniumWebDriver x testeCalculadora.java x
16 public class testeCalculadora {
17     //Declara uma variavel statica do tipo "WebDriver"
18     private static WebDriver driver;
19
20     public testeCalculadora() {
21     }
22
23     @BeforeClass
24     public static void Conectar() {
25         //Especificando tipo e caminho do driver
26         System.setProperty("webdriver.gecko.driver", "geckodriver.exe");
27         //Instanciando o driver de conexão
28         driver = new FirefoxDriver();
29     }
30
31     @AfterClass
32     public static void Encerrar() {driver.close();}
33
34     @Test
35     public void testeSoma() {
36         //Passando endereço para acessar o site
37         driver.get("https://web2.0calc.com/");
38         WebElement numero5 = driver.findElement(By.id("Btn5"));
39         numero5.click();
40         WebElement soma = driver.findElement(By.id("BtnPlus"));
41         soma.click();
42         WebElement numero3 = driver.findElement(By.id("Btn3"));
43         numero3.click();
44         WebElement igualdade = driver.findElement(By.id("BtnCalc"));
45         igualdade.click();
46         String esperado = "8";
47         String atual = driver.findElement(By.id("input")).getAttribute("value");
48         assertEquals(atual, esperado);
49     }
50 }
```

EXEMPLO

○ Exemplo 3:

Nesse terceiro exemplo faremos a verificação do login no Facebook:

- ✓ PREENCHER OS DADOS DE **EMAIL E SENHA**
- ✓ VERIFICAR SE O USUÁRIO ESTÁ LOGADO

EXEMPLO

○ Exemplo 3:

```
m seleniumWebDriver x LoginSUAP.java x TesteLoginFacebook.java x
13 public class TesteLoginFacebook {
14
15     private static WebDriver driver;
16
17     public TesteLoginFacebook() {}
18
19     @BeforeClass
20     public static void Conectar() {
21
22         System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
23         driver = new ChromeDriver();
24
25     }
26
27     @AfterClass
28     public static void Encerrar() {driver.close();}
29
30     @Test
31     public void testeLogin() {
32
33         driver.get("https://pt-br.facebook.com/");
34
35         WebElement email = driver.findElement(By.id("email"));
36         email.sendKeys("seleniumteste8@gmail.com");
37
38         WebElement senha = driver.findElement(By.id("pass"));
39         senha.sendKeys("teste010203");
40
41         driver.findElement(By.id("u_0_2")).submit();
42
43         String atual = driver.findElement(By.xpath("//*[@id=\"u_0_a\"]/div[1]/div[1]/div/a/span/span")).getText();
44
45         assertEquals("Selenium", atual);
46     }
47 }
```

EXEMPLO

○ Exemplo 4:

Nesse quarto exemplo será realizado um cadastro no mercado livre e será verificado a mensagem de conta existente:

- ✓ PREENCHER OS DADOS DE CADASTRO
- ✓ CLICAR NO BOTÃO DE CADASTRO
- ✓ VERIFICAR MENSAGEM

EXEMPLO

○ Exemplo 4:

```
m seleniumWebDriver x CadastroMercadoLivre.java x
14
15 public class CadastroMercadoLivre {
16
17     private static WebDriver driver;
18
19     public CadastroMercadoLivre() {
20
21     }
22
23     @Before
24     public void setUpClass() {
25
26         System.setProperty("webdriver.gecko.driver", "geckodriver.exe");
27         driver = new FirefoxDriver();
28
29     }
30
31     @After
32     public void tearDownClass() {
33
34         driver.close();
35
36     }
37
38     @Test
39     public void testeCadastro() {
40
41         driver.get("https://www.mercadolivre.com.br/");
42
43         WebElement btCadastro = driver.findElement(By.xpath("//*[@id=\"nav-header-menu\"]/a[1]"));
44         btCadastro.click();
45
46         WebElement nome = driver.findElement(By.id("signupFirstName"));
47         nome.sendKeys("teste");
48
49     }
50 }
```

EXEMPLO

○ Exemplo 4:

```
seleniumWebDriver x CadastroMercadoLivre.java x
36 }
37
38 @Test
39 public void testeCadastro() {
40
41     driver.get("https://www.mercadolivre.com.br/");
42
43     WebElement btCadastro = driver.findElement(By.xpath("//*[@id=\"nav-header-menu\"]/a[1]"));
44     btCadastro.click();
45
46     WebElement nome = driver.findElement(By.id("signupFirstName"));
47     nome.sendKeys("teste");
48
49     WebElement sobrenome = driver.findElement(By.id("signupLastName"));
50     sobrenome.sendKeys("selenium");
51
52     WebElement email = driver.findElement(By.id("signupEmail"));
53     email.sendKeys("selinumteste8@gmail.com");
54
55     WebElement senha = driver.findElement(By.id("signupPassword"));
56     senha.sendKeys("teste010203");
57
58     WebElement entrar = driver.findElement(By.xpath("//*[@id=\"signup\"]/div/div[2]/button"));
59     entrar.click();
60
61     String aviso = (driver.findElement(By.xpath("//*[@id=\"use-other-email-modal\"]/\"
62         + \"div[1]/div/div/div[2]/h3\"))).getText();
63     System.out.println(aviso);
64     assertEquals(aviso, atual: "Você já fez um cadastro com selinumteste8@gmail.com.");
65
66 }
67
68 }
```

EXEMPLO

○ Exemplo 5:

Verifica login no SUAP, e verificar se o título da página é : “- SUAP: Sistema Unificado de Administração Pública”.

- ✓ FAZER LOGIN
- ✓ VERIFICAR O TÍTULO DA PÁGINA

EXEMPLO

○ Exemplo 5:

```
m seleniumWebDriver x LoginSUAP.java x
11
12 public class LoginSUAP {
13
14     private static WebDriver driver;
15
16     public LoginSUAP() {
17     }
18
19     @BeforeClass
20     public static void setUpClass() {
21
22         System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
23         driver = new ChromeDriver();
24
25     }
26
27     @Test
28     public void login() {
29         driver.get("https://suap.ifpb.edu.br");
30
31         driver.findElement(By.name("username")).sendKeys(User.getUsername());
32         driver.findElement(By.name("password")).sendKeys(User.getPassword());
33         driver.findElement(By.tagName("form")).submit();
34
35         assertEquals("expected: " - SUAP: Sistema Unificado de Administração Pública",
36                     driver.getTitle());
37
38     }
39
40 }
41
```

REFERÊNCIAS

- <https://www.trabalhosgratuitos.com/Exatas/Inform%C3%A1tica/Testes-Web-com-Selenium-823657.html>
- <https://www.devmedia.com.br/introducao-aos-testes-funcionais-automatizados-com-junit-e-selenium-webdriver/28037>
- <http://taketest.take.net/2016/02/05/selenium-webdriver-como-interagir-com-o-browser/>
- <https://pt.slideshare.net/Qualister/automacao-de-testes-funcionais-com-selenium-webdriver>
- http://www.w3ii.com/pt/selenium/selenium_webdriver.html