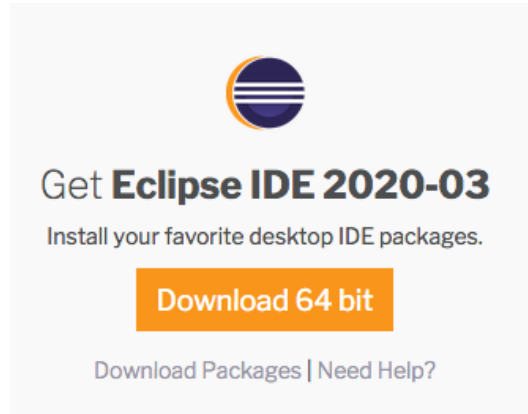


Instalação

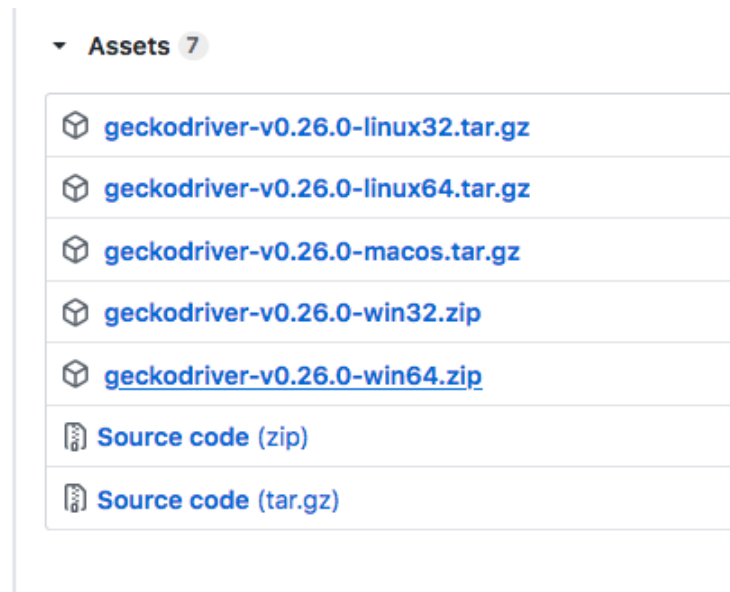
1. Fazer download do eclipse acessando <http://www.eclipse.org/downloads/> e clicando em *download packages*



2. Fazer download do Selenium em <http://www.seleniumhq.org/download/>

Selenium Client & WebDriver Language Bindings					
In order to create scripts that interact with the Selenium Server (Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers.					
While language bindings for other languages exist , these are the core ones that are supported by the main project hosted on GitHub.					
LANGUAGE	STABLE VERSION	RELEASE DATE	ALPHA VERSION	ALPHA RELEASE DATE	LINKS
Ruby	3.142.6	October 04, 2019	4.0.0alpha5	March 18, 2020	Download Alpha Download Changelog API Docs
Java	3.141.59	November 14, 2018	4.0.0-alpha-5	March 18, 2020	Download Alpha Download Changelog API Docs
Python	3.141.0	November 01, 2018	4.0.0a5	March 18, 2020	Download Alpha Download Changelog API Docs
C#	3.14.0	August 02, 2018	4.0.0-alpha05	March 18, 2020	Download Alpha Download Changelog API Docs
JavaScript	3.6.0	October 06, 2017	4.0.0-alpha.7	March 05, 2020	Download Alpha Download Changelog API Docs

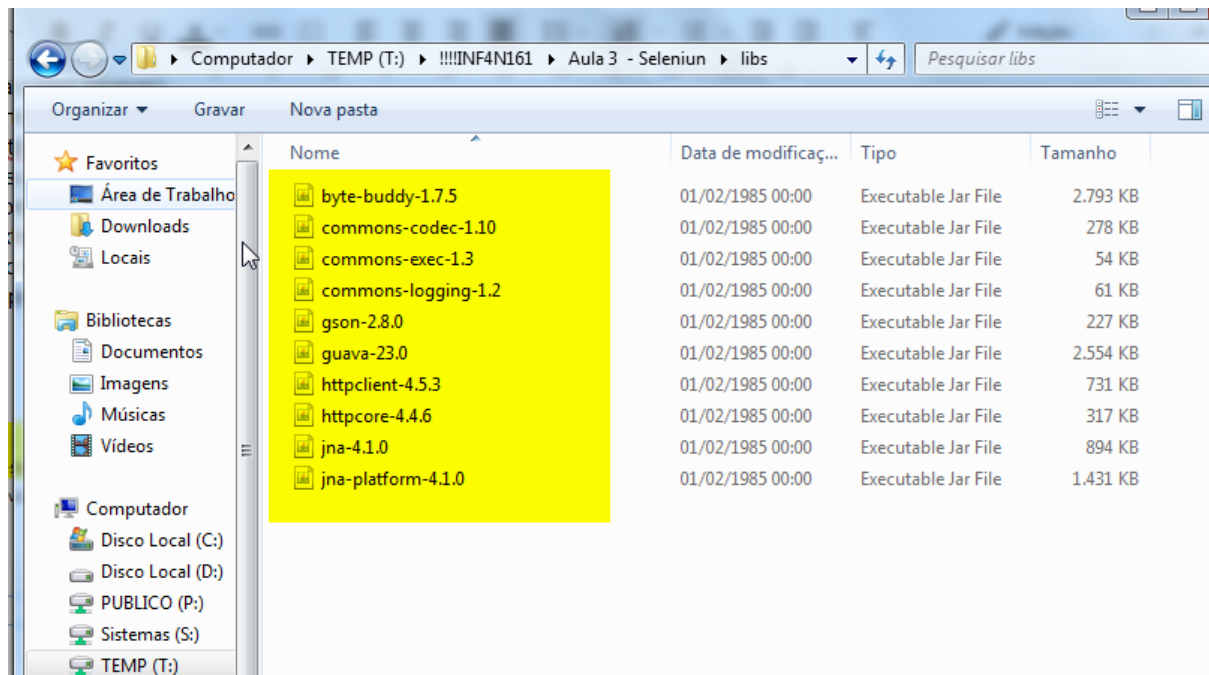
3. Extrair o Seleniun na pasta documentos do seu computador
4. Fazer o download do Gecko driver em <https://github.com/mozilla/geckodriver/releases>



5. Colocar este arquivo dentro do local a onde está o eclipse (em documentos/eclipse caso você tenha feito certo o passo 3)
6. Abra a pasta eclipse e dê um duplo clique sobre o arquivo eclipse (o executável que tem uma lua cheia como símbolo)
7. Criar no eclipse um novo projeto Java (file -> new -> other -> java project)
8. Clicar em BuildPath -> Configure BuildPath
9. Clicar em Libraries e depois em Add Library
10. Selecionar JUnit, preferencialmente a versão mais atualizada
11. Selecionar Add External Jars
12. Localizar o local onde está foi extraído o Selenium e selecionar os dois arquivos .jar que estão fora da pasta libs.

Nome	Data de modificação	tipo	Tamanho
eclipse	09/10/2017 19:43	Pasta de arquivos	
libs	01/02/1985 00:00	Pasta de arquivos	
CHANGELOG	01/02/1985 00:00	Arquivo	109 KB
client-combined-3.6.0	01/02/1985 00:00	Executable Jar File	1.393 KB
client-combined-3.6.0-sources	01/02/1985 00:00	Executable Jar File	466 KB
eclipse-jee-oxygen-1a-win32-x86_64	01/11/2017 09:56	Pasta compactada	338.942 KB
Guia Seleniun	06/11/2017 19:18	Adobe Acrobat D...	132 KB
LICENSE	01/02/1985 00:00	Arquivo	12 KB
NOTICE	01/02/1985 00:00	Arquivo	1 KB
selenium-java-3.6.0	01/11/2017 09:51	Pasta compactada	9.241 KB

13. Localizar o local onde está foi extraído o Selenium, abrir a pasta libs e selecionar todos os .jar que estão dentro da pasta libs



14. Criar neste projeto um novo pacote com o nome seguindo padrão de projeto pais.com.empresa.projeto.arquivos no caso br.com.senacrs.automacao.testes
15. Clicar com o botão direito no pacote criado e ir em new JUnitTestCase (se não aparecer na lista clicar em other e localizar este JUnitTestCase).

Fundamentos principais de JUnit e Selenium

JUnit é utilizado principalmente para testes unitários, o Selenium utiliza a sintaxe do JUnit para efetuar testes funcionais. A estrutura básica de uma classe de teste conta com as notações @before @test e @after, sendo que: @before deve ser colocado sobre o método que é executado antes dos testes, por exemplo: método que abre o navegador; @test é colocado nos métodos que contém testes e @after nos métodos executados após o teste, por exemplo fechar navegador

- O geckodriver que foi colocado nas máquinas é o que permite que o Selenium interaja com o navegador Firefox, caso utilize-se outro navegador é necessário outro driver.
- A seguir um exemplo que deve ser implementado com os alunos de abertura de browser e inserção de valores em **campos de texto**.

ATENÇÃO: Os comandos Selenium que serão utilizados neste exemplo, fazem referência ao código php que está no final desta página. Para que execute corretamente você precisa ter o ambiente de desenvolvimento php instalado. Recomenda-se instalar a ferramenta Xampp e colocar este código dentro de um arquivo chamado index.php na pasta c://xampp//htdocs

Primeiro método (para abertura de navegador, acesso a página web e instância de browser)

```
@Before
public void setUp() {
//setando o caminho do gecko
    System.setProperty("webdriver.gecko.driver", "C:\\Eclipse\\geckodriver.exe");
//instanciando o firefox
    driver = new FirefoxDriver();
//informando qual arquivo que vai abrir
    driver.get("http://localhost/automacao/index.php");
//maximizando as telas
    driver.manage().window().maximize();
}
```

Segundo método (para localizar campos de texto)

```
@Test
public void insereValores() {

//findElement localiza um elemento na página
    WebElement txNome = driver.findElement(By.name("txNome"));
//sendKeys coloca o valor que está informado ali no campo da tela
    txNome.sendKeys("Aline Zanin");

    WebElement txcpf = driver.findElement(By.name("txcpf"));
    txcpf.sendKeys("05647890323");

    WebElement txbithDate = driver.findElement(By.name("txbithDate"));
    txbithDate.sendKeys("09/10/1991");

    WebElement txprovince = driver.findElement(By.name("txprovince"));
    txprovince.sendKeys("Rio Grande do Sul");

    WebElement txcity = driver.findElement(By.name("txcity"));
    txcity.sendKeys("Marau");

    WebElement txzipCode = driver.findElement(By.name("txzipCode"));
    txzipCode.sendKeys("99150000");

    WebElement txstreet = driver.findElement(By.name("txstreet"));
    txstreet.sendKeys("xxxx yyy");

    WebElement txnumber = driver.findElement(By.name("txnumber"));
    txnumber.sendKeys("124");

    WebElement txadInfo = driver.findElement(By.name("txadInfo"));
```

```
txadInfo.sendKeys("apto 8");

WebElement txpassword = driver.findElement(By.name("txpassword"));
txpassword.sendKeys("12345");
WebElement txpasswordC = driver.findElement(By.name("txpasswordC"));
txpasswordC.sendKeys("12345");

WebElement txemail = driver.findElement(By.name("txemail"));
txemail.sendKeys("a@a.com");

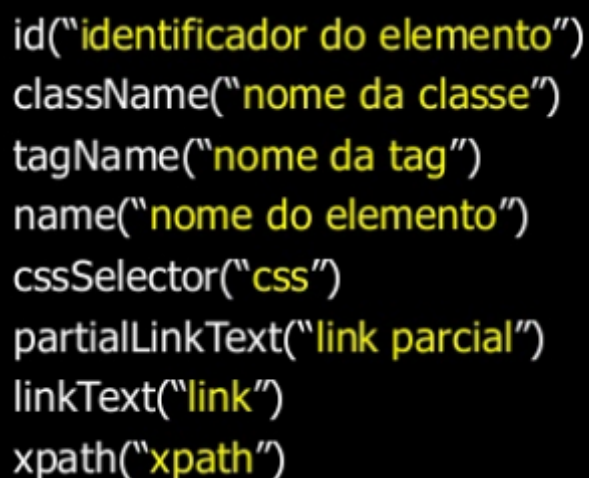
WebElement btnEnviar = driver.findElement(By.name("btnEnviar"));
btnEnviar.click();
}
```

Terceiro método

```
@After
public void fecharPagina() {
//fecha o navegador
    driver.quit();
}
```

```
WebElement radioBtnF = driver.findElement(By.id("pfisica"));
//click(); clica no lugar determinado
radioBtnF.click();
```

- **Volve para o slide e mostre os demais tipos de localizadores de campos de páginas (link, link parcial, css ...)**



```
id("identificador do elemento")
className("nome da classe")
tagName("nome da tag")
name("nome do elemento")
cssSelector("css")
partialLinkText("link parcial")
linkText("link")
xpath("xpath")
```

Capturar dados em Checkbox, RadioButton

Para estes testes utilizaremos a seguinte url: <http://toolsqa.wpengine.com/automation-practice-form>
Existem duas formas de capturar valores de radio e combo, a primeira delas é através do índice

```
@Test
public void test() {
    driver.get("http://toolsqa.wpengine.com/automation-practice-form");

    /*criar um list de WebElement e utilizar findElements no plural para localizar todos
    os elementos que tem o mesmo name e colocar em uma lista.*/
    List<WebElement> rdBtn_Sex = driver.findElements(By.name("sex"));

    /*utilizando o comando get(indice) e passando para ele o número de um índice,
    consegue capturar os valores pelo índice, respeitando a ordem em que eles
    aparecem no rádio e iniciando em 0. O comando click vai marcar a opção
    determinada pelo índice.*/
    rdBtn_Sex.get(1).click();
}
```

A segunda forma é através do value

```
@Test
public void test() {

    /*Carregando a URL*/
    driver.get("http://toolsqa.wpengine.com/automation-practice-form");

    /*Cria um list de webElement localizando pelo nome*/
    List<WebElement> oCheckBox = driver.findElements(By.name("exp"));

    /*Através de size verifica o tamanho do list, quantas opções o radio ou o check possui*/
    int iSize = oCheckBox.size();

    /*Faremos um for para percorrer a lista*/
    for(int i=0; i < iSize ; i++){

        /*Para cada um dos elementos do list capturaremos o value*/
        String sValue = oCheckBox.get(i).getAttribute("value");
```

```

        /*Com um if vamos verificar se é o value que estamos buscando
        e se for vamos clicar nele*/
        if (sValue.equalsIgnoreCase("3")){
            oCheckBox.get(i).click();
            break;
        }
    }
}

```

Para verificar se uma opção está selecionada utilizaremos
bValue = rdBtn_Sex.get(0).isSelected();

Capturar dados em List

- Existem três formas de capturar dados de list, sendo elas por texto, por index ou por value e por index

- Por texto:

```

@Test
public void test() {
    /*Carregando a URL*/
    driver.get("http://toolsqa.wpengine.com/automation-practice-form");
    /*Cria-se um objeto do tipo Select*/
    Select dropdown = new Select(driver.findElement(By.name("continents")));
    /*Seleciona-se por texto*/
    dropdown.selectByVisibleText("Europe");
}

```

- Por index

```

@Test
public void test() {
    /*Carregando a URL*/
    driver.get("http://toolsqa.wpengine.com/automation-practice-form");
    /*Cria-se um objeto do tipo Select*/
    Select dropdown = new Select(driver.findElement(By.name("continents")));
    /*Seleciona-se por index*/
    dropdown.selectByIndex(1);
}

```

- Por value

```

@Test
public void test() {
    /*Carregando a URL*/
    driver.get("http://toolsqa.wpengine.com/automation-practice-form");

```

Tutorial de utilização da Ferramenta de automação de Testes Selenium WebDriver
 Criado e Mantido pela Prof. Dra. Aline Zanin e atualizado pelo Prof. MSc. Adalto Selau Sparremerger
 Edição, cópia e reprodução é permitida somente com permissão dos autores.

```
/*Cria-se um objeto do tipo Select*/
    Select dropdown = new Select(driver.findElement(By.name("continents")));
/*Seleciona-se por value*/
    dropdown.selectByValue("Africa");
}
```

EXERCÍCIO

1. Crie um arquivo HTML que contenha um rádio button com três opções, um checkbox com três opções e um list com cinco opções.
2. Crie um caso de teste que seleciona o segundo elemento do radio button
3. Crie um caso de teste que seleciona um elemento do radio button por nome
4. Crie um caso de teste que seleciona o terceiro elemento do checkbox
5. Crie um caso de teste que seleciona um elemento do checkbox por nome
6. Crie um caso de teste que seleciona o quarto elemento do list
7. Crie um caso de teste que seleciona um elemento do list por nome

```
//limpa o campo
WebElement firstname = driver.findElement(By.name("firstname")); firstname.sendKeys("aaaaaaa");
firstname.clear()

// Retorna uma lista do tipo WebElement com todas as tags "<option>".
dropdown.getOptions();

// Remove todas as seleções.
dropdown.deselectAll();

// Remove seleção de uma opção específica através de um dado índice.
dropdown.deselectByIndex(int index);

// Remove seleção de todas as opções que tiverem seus atributos "value" especificado.
dropdown.deselectByValue(String value)

// Remove seleção de todas as opções que forem iguais ao argumento especificado.
dropdown.deselectByVisibleText(String text);

// Retorna objeto do tipo WebElement contendo a primeira tag "<option>" que esta selecionada.
dropdown.getFirstSelectedOption();
```



```
// Retorna uma lista do tipo WebElement com todas as tags "<option>" que estão selecionadas.
dropdown.getAllSelectedOptions();
```

```
// Retorna uma lista do tipo WebElement com todas as tags "<option>".
dropdown.getOptions();
```

```
// Retorna verdadeiro ou falso, dependendo do valor do atributo "multiple".
dropdown.isMultiple();
```

```
// vai para uma pagina específica
driver.navigate().to("http://www.google.com");
```

```
// volta para pagina anterior
driver.navigate().back();
```

```
// avança para proxima pagina
driver.navigate().forward();
```

```
// atualizar pagina
driver.navigate().refresh();
```

//Upload de arquivos

```
driver.findElement(By.id("fileToUpload")).sendKeys("C:\\Users\\lazanin\\Desktop\\diarioClasse
Impressao.pdf");
```

EXERCICIO

- No mesmo arquivo criado na aula anterior adicione um select do tipo multiple
- Exercite os comandos estudados na aula de hoje, mostrando os resultados em console

Como manipular alerts

Exemplo de html:

```
<html>
<header><script> alert("Eu sou um alerta!"); </script> </header>
<body>
```

```
<h1>oi</h1>
</body>
</html>
```

```
@Test
public void main() {
    driver.get("file:///C:/Users/daniele/Desktop/teste.html");
    driver.manage().window().maximize();
    Alert alerta = driver.switchTo().alert();
    alerta.accept();
}
```

Trabalhando com Confirm (Alert de dois botoes)

```
<!DOCTYPE html>
<html>
<head>
<title>Tutorial de Alert em JavaScript - Linha de Código</title>
</head>
<body>
<p>Clique no botão para exibir a caixa de confirmação.</p>
<button id="btn1" onclick="funcao1()">Clique aqui</button>
<p id="demo"></p>
<script>
```

```
function funcao1()
{
    var x;
    var r=confirm("Escolha um valor!");
    if (r==true) {
        x="você pressionou OK!";
    } else {
        x="Você pressionou Cancelar!";
    }
    document.getElementById("demo").innerHTML=x;
}
</script>
</body>
</html>
```

Código selenium

```
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.junit.Test;
```

Tutorial de utilização da Ferramenta de automação de Testes Selenium WebDriver

Criado e Mantido pela Prof. Dra. Aline Zanin e atualizado pelo Prof. MSc. Adalto Selau Sparremerberger

Edição, cópia e reprodução é permitida somente com permissão dos autores.

```
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class teste {

@Test

public void main() {
    System.setProperty("webdriver.gecko.driver", "C:\\Users\\daniele\\eclipse\\geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    // Setando um tempo de espera
    // Launch the URL
    driver.get("file:///C:/Users/daniele/Desktop/teste.html");
    driver.manage().window().maximize();
    WebElement btn1 = driver.findElement(By.id("btn1"));
    btn1.click();
    Alert alerta = driver.switchTo().alert();
    alerta.accept();
    alerta.dismiss();
}
```

CÓDIGO PHP exemplo

```
<!DOCTYPE html>

<html>
<head>
<meta charset="UTF-8">
<title></title>
<style>
body{
font-family: Verdana, Arial;
}
</style>
</head>

<body>
<h1 Customer Registration Page! </h1>
<form>
```

Tutorial de utilização da Ferramenta de automação de Testes Selenium WebDriver

Criado e Mantido pela Prof. Dra. Aline Zanin e atualizado pelo Prof. MSc. Adalto Selau Sparremberger

Edição, cópia e reprodução é permitida somente com permissão dos autores.

```

<label>    Name:    </label>    <input    type="numeric"    name="txNome"    required/>    <br/>
<label>Birth    Date:</label>    <input    type="date"    name="txbithDate"    required=""/><br/>
<label>Province:</label> <input type="text" name="txprovince" required=""/><br/>    <label>City:</label>
<input
    type="text"
    name="txcity"
    required=""/><br/>
<label>Zip Code:</label> <input type="text" name="txzipCode" required=""/><br/>    <label>Street:</label>
<input
    type="text"
    name="txstreet"
    required=""/><br/>
<label>Number:    </label>    <input    type="text"    name="txnumber"    required=""/><br/>
<label>Additional    Information:    </label>    <input    type="text"    name="txadInfo"    required=""/><br/>
<label>Password:</label> <input type="text" name="txpassword" required=""/><br/>

<label>Confirm Password: </label> <input type="text" name="txpasswordC" required=""/><br/>
<label> E-mail: </label> <input type="text" name="txemail" required="" /><br/>Select the customer
type:<br/>
<input type="radio" name="tipoPessoa" value="fisica" id="pfisica"> Business Customer (legal person) <br/>
<input type="radio" name="tipoPessoa" value="juridica" id="pjuridica"> Personal Customer (natural person )<br/>
Natural Person Identification (Brazilian CPF): <input type="text" name="txcpf" /><br/>
Legal Person Identification (Brazilian CNPJ): <input type="text" name="txcnpj" /><br/>
<input type="submit" >
</form>
</body>
</html>

```

Exemplo de código para colocar o foco em um elemento sem clicar nele, útil para menus

```

package testeEcommerce;
import java.util.concurrent.TimeUnit;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

```

```
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;

public class teste {

    WebDriver driver;

    @Before

    public void setUp() {
        System.setProperty("webdriver.gecko.driver",
"C:\\Users\\lzanin\\Documents\\eclipse\\geckodriver.exe");
    }

    @Test
    public void insereValores() {
        driver = new FirefoxDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.get("http://store.demoga.com/");
        WebElement element = driver.findElement(By.linkText("Product Category"));
        Actions action = new Actions(driver);
        action.moveToElement(element).build().perform();
        driver.findElement(By.linkText("iPads")).click();
    }

    @After
    public void fecharPagina() {
        driver.quit();
    }
}
```