



Universidade Federal do Ceará
Centro de Ciências/Departamento de Computação
Código da Disciplina: CK0236
Professor: Ismayle de Sousa Santos

Aula 13

Técnica de Programação II

Testes Automatizados



qpg4p5x



ismaylesantos@great.ufc.br



@IsmayleSantos

Agenda

- Por quê automatizar?
- Quando automatizar?
- Frameworks de Testes Automatizados



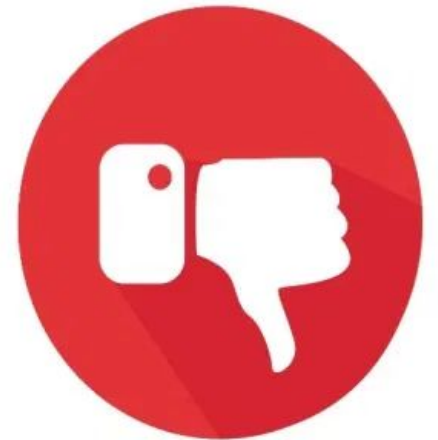
Vantagens da Automação

- Os testes automatizados são passíveis de repetição a qualquer momento e com pouco esforço
- Geralmente não requer modificação na aplicação
- Proporciona redução de tempo e custo a longo prazo
- Facilita o teste de regressão



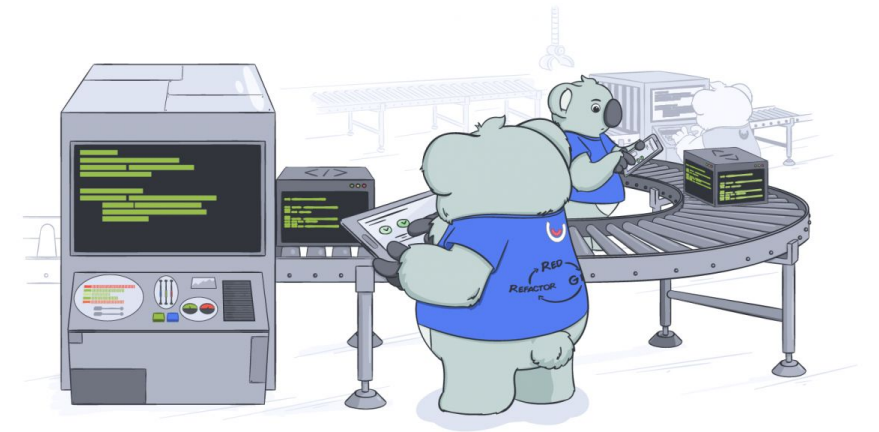
Desvantagem da Automação

- **Maior investimento inicial**
 - É necessário implementar o script de teste
- **Nem todo teste é possível de automatizar com as ferramentas atuais**
 - E.g.: verificar a mensagem de voz de uma aplicação
- **Nem tudo está especificado para servir de entrada para automatizar os testes**
 - E.g.: verificar a posição de elementos de interface na tela
- **Alterações na interface de usuário podem requerer alterações no script de teste**



Quando Automatizar

- **Levar em consideração**
 - Maturidade do Software sob teste
 - Complexidade e Ciclo de Vida da Aplicação
 - Releases do produto de software
 - Benefícios econômicos da automação
 - Disponibilidade de ferramenta de teste
 - Skill da equipe



Ferramentas de Testes de Aplicações Web

- No momento, **Selenium** é a ferramenta mais popular e usada do mercado. O Selenium WebDriver ajuda a criar scripts de automação de testes mais complexos e avançados
- **TestComplete** é uma plataforma comercial que pode automatizar testes do desktop, celular e aplicações web



Ferramentas de Testes de Aplicações Móveis

- Selendroid
 - É uma ferramenta derivada da Selenium que testa aplicações desktop e da Web
- Robotium
 - Estende a estrutura de teste do Android para simplificar a gravação de testes
- Appium
 - A Sauce Labs ampliou o Appium para suportar testes automatizados para aplicações Android e Firefox OS



Robotium





Selenium

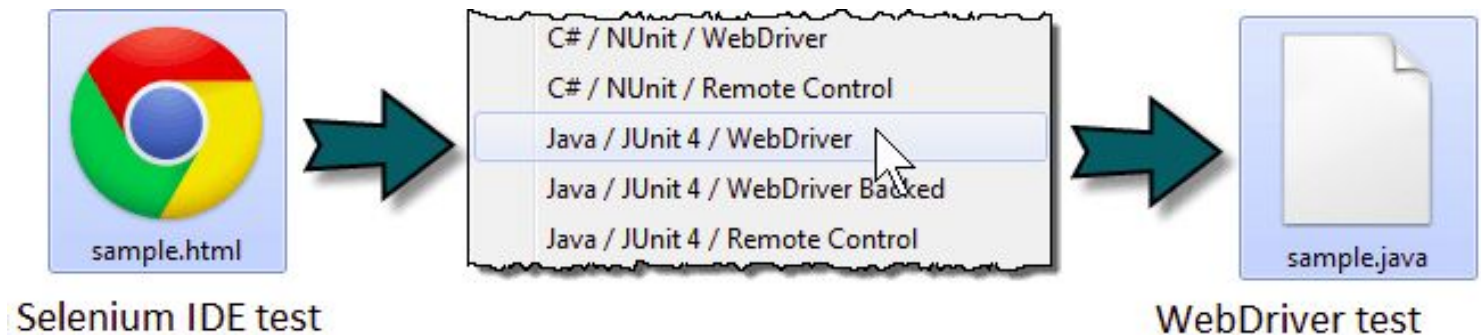
- É um ferramenta utilizada principalmente para automatizar aplicações web para fins de teste
 - Mas também é bastante utilizada para automatizar tarefas em sistemas web
 - Tipos:
 - Selenium IDE
 - plugin do tipo record-and-playback
 - Selenium WebDriver
 - permite o desenvolvimento de testes automatizadas em diferentes linguagens
 - Selenium Grid
 - permite distribuir e executar os testes em várias máquinas
-

Selenium IDE



O que é Selenium IDE?

- Ambiente de desenvolvimento integrado para scripts Selenium
- Implementado como uma extensão do Firefox e Chrome
- Permite que se grave, edite e execute testes
- Edição de scripts manualmente



Como Instalar o Selenium IDE?

- <https://www.selenium.dev/>

[Página inicial](#) > [Extensões](#) > Selenium IDE



Selenium IDE

Oferecido por: seleniumhq.org

★★★★★ 213

| [Ferramentas do desenvolvedor](#)

|  400.000+ usuários

Usar no Chrome

Visão geral

Como seus dados são usados

Comentários

Suporte

Itens relacionados

Prática Selenium IDE

Selenium IDE - Teste 1

Project: Teste 1

Tests +

Search tests...

Pesquisa no Google

https://www.google.com

	Command	Target	Value
1	open	/	
2	set window size	1552x840	
3	click	name=q	
4	type	name=q	Teste de Software
5	click	xpath=//input[@name='btnK']	
6	assert text	css=g:nth-child(2) > .tF2Cxc:nth-child(2) .LC20lb > span	Teste de Software: Introdução, Conceitos Básicos e Tipos de ...
7	store	Valor Armazenado	variableName2

Command

open

//

Target

/

Value

Description

Log

Reference

Prática Selenium IDE

- Alguns dos principais comandos utilizados

- Click
- Type
- Story
- Wait
- Verify
 - Se falhar, teste continua
- Assert
 - Se falhar, teste é finalizado





Selenium
WebDriver

O que é Selenium WebDriver?

- É uma API orientada a objetos
 - Fornece uma interface de programação mais simples e mais concisa
 - Desenvolvido para suportar melhor a automação de páginas web dinâmicas
-

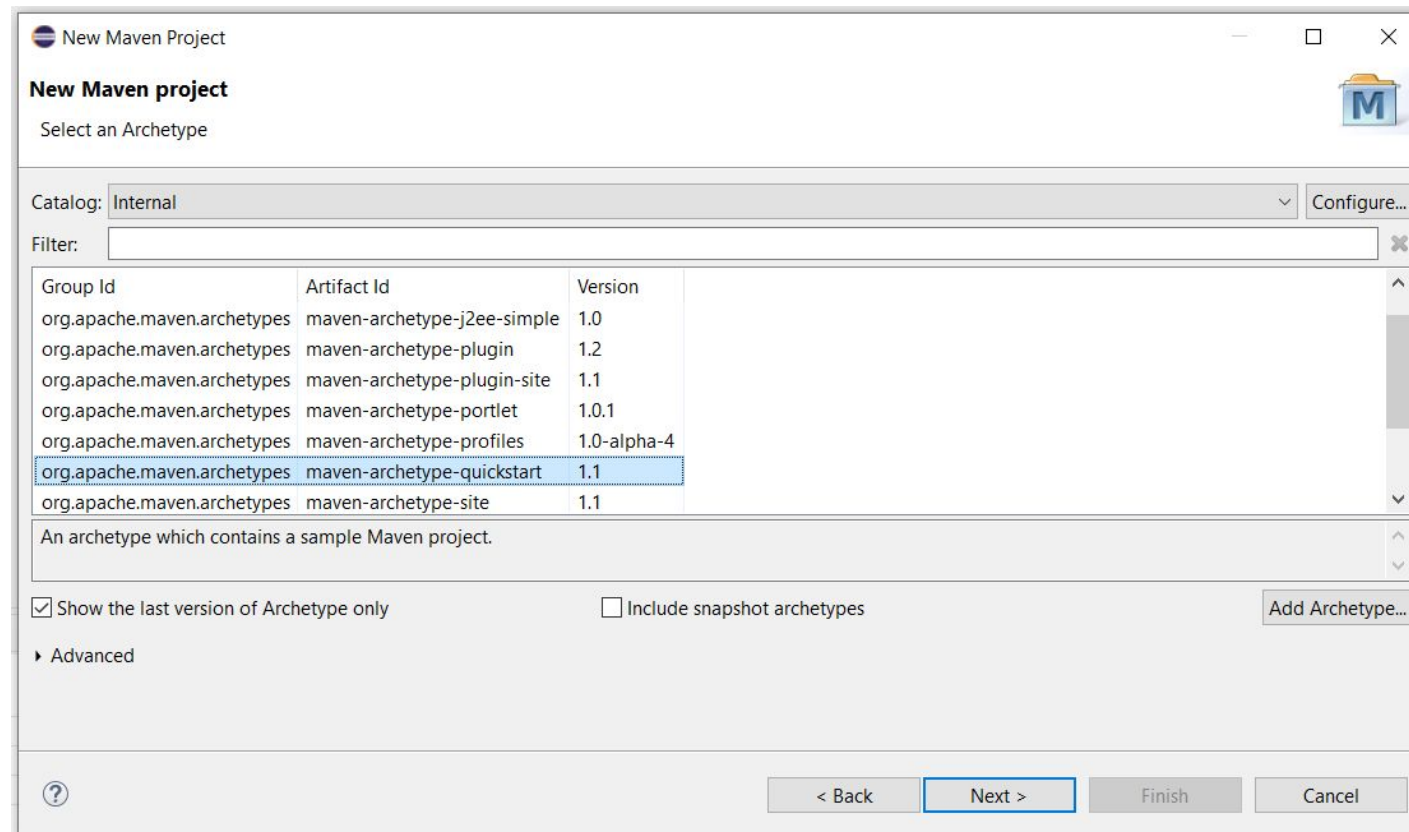
Vamos utilizar o Maven

- Maven

- Ferramenta de Automação de Compilação e Geração de Dependências
 - Vantanges
 - Facilita a configuração do projeto seguindo melhores práticas
 - Uso dos templates de projeto (archetypes)
 - Download e atualização automática de dependências
 - Dependências podem ser carregadas de repositório local ou público
 - <https://search.maven.org/classic/>
 - Configuração do projeto é feita no Project Object Model (POM)
 - pom.xml
-

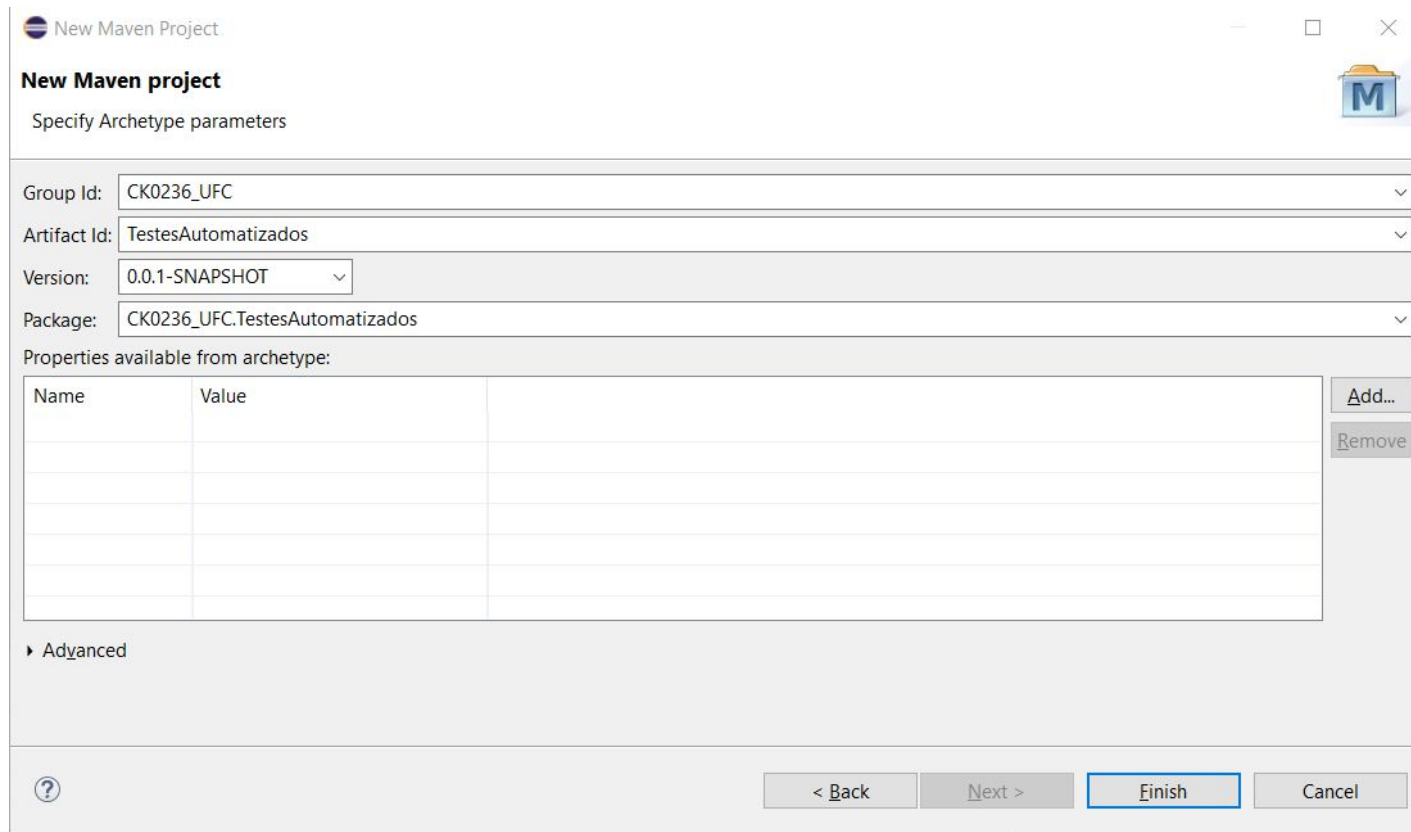
Configuração do Projeto Maven

- File → New → other → Maven Project
 - Selecione maven-archetype-quickstart



Configuração do Projeto Maven

- Selecione o Group Id (nome da organização) e o Artifact Id (e.g., nome do projeto)



New Maven Project

New Maven project
Specify Archetype parameters

Group Id: CK0236_UFC

Artifact Id: TestesAutomatizados

Version: 0.0.1-SNAPSHOT

Package: CK0236_UFC.TestesAutomatizados

Properties available from archetype:

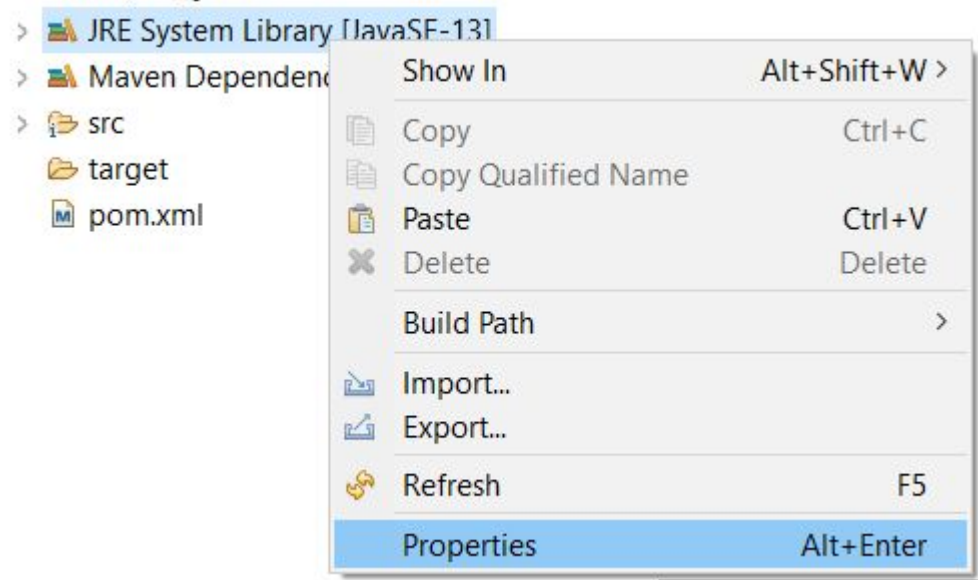
Name	Value

Advanced

< Back Next > Finish Cancel

Configuração do Projeto Maven

- Se precisar alterar a versão do ambiente de execução (JRE System Library) basta clicar com o botão direito do Mouse -> Propriedades



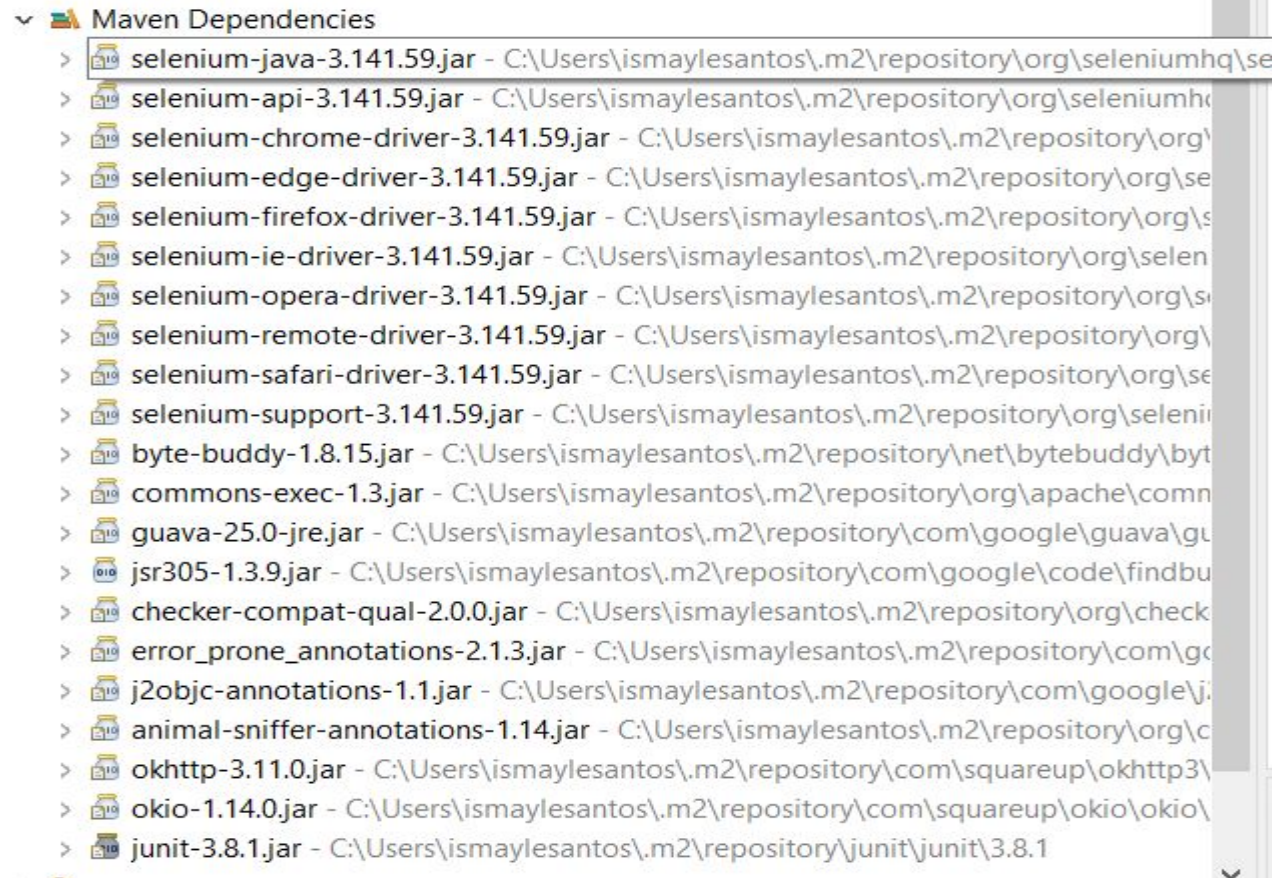
Configuração do Projeto Maven

- <https://www.selenium.dev/maven/>
 - Agora basta adicionar a dependência do Selenium no pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
  </dependency>
</dependencies>
```

Configuração do Projeto Maven

- Após editar o arquivo POM e salvar, o Maven já faz o download de todas as bibliotecas necessárias



Configurações

- Agora você já pode começar seus casos de testes
 - Se preferir, pode importar o código gerado pelo Selenium IDE para usar como base
 - Nota:
 - Você vai precisar baixar e configurar o drive do browser
 - <https://chromedriver.chromium.org/>

```
System.setProperty("webdriver.chrome.driver", "D:\\Workspace - Eclipse\\TestesAutomatizados\\chromedriver.exe");  
driver = new ChromeDriver();
```

Comandos Básicos

```
//iniciar browser
```

```
WebDriver driver = new ChromeDriver();
```

```
//abrir página
```

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

```
//fechar browser
```

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

```
//localizador do elemento por id
```

```
By locator = By.id("id");
```

```
//obter o elemento
```

```
WebElement element = driver.findElement(locator);
```

Comandos Básicos

```
//clicar no elemento
```

```
element.click();
```

```
//escrever texto
```

```
element.sendKeys("text");
```

```
//enviar tecla do teclado
```

```
element.sendKeys(Keys.ENTER);
```

```
//obter texto
```

```
String text = element.getText();
```

```
//limpar texto
```

```
element.clear();
```

Localizando Elementos

//Tag Name -> tag do elemento

```
By locator = By.tagName("tagName");
```

//Link Text -> texto que é mostrado para elementos que são link

```
By locator = By.linkText("linkText");
```

//Class Name -> classe do elemento

```
By locator = By.className("className");
```

//CSS selector -> css do elemento

```
By locator = By.cssSelector("cssSelector");
```

//Xpath -> caminho de localização do elemento

```
By locator = By.xpath("xpath");
```

Comandos em Xpath

<code>div/button</code>	//busca nós imediatamente seguintes
<code>div//button</code>	//busca em todos os descendentes do nó
<code>div/*</code>	//retorna filhos de todas as tags
<code>count(//div)</code>	//retorna número de elementos existentes
<code>div[3]</code>	//retorna o n-ésimo elemento
<code>div[position()="3"]</code>	//equivalente ao anterior
<code>button[@class="close"]</code>	//retorna nó com valor específico de atributo
<code>span[contains(text(),"texto")]</code>	//retorna nó com atributo que contém um valor

Xpath Axes

- Permitem a identificação de elementos por sua relação de parentesco como pai, filho, irmão, etc...
- São chamados assim porque eles se referem ao eixo em que os elementos estão em relação a um elemento
- Notações mais comuns:

`/following-sibling::*`

`//irmãos seguintes`

`/preceding-sibling::*`

`//irmãos anteriores`

`/child::*`

`//filhos`

`/parent::*`

`//pai`

`/descendant::*`

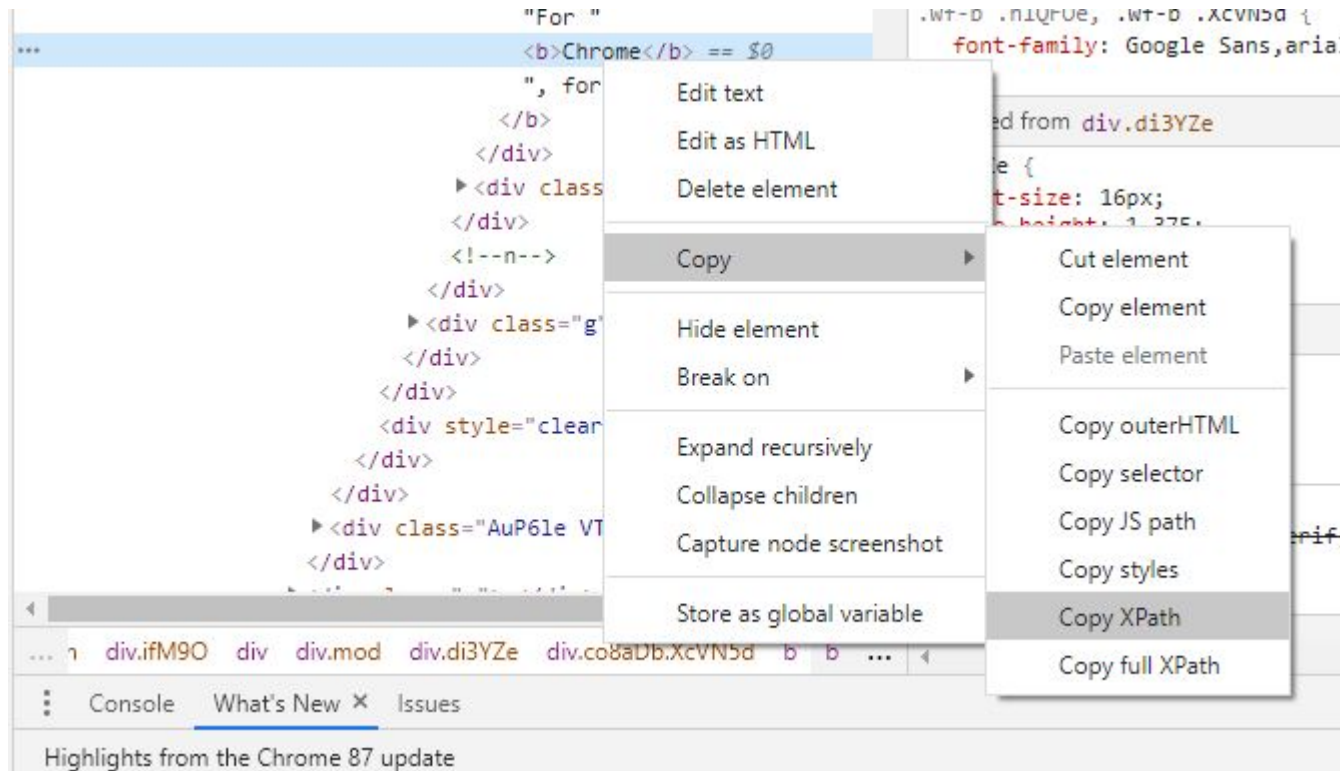
`//todos os descendentes`

`/ancestor::*`

`//todos os ancestrais`


Mais sobre Xpath

- https://www.w3schools.com/xml/xpath_syntax.asp



Mais sobre Xpath

- Usando xPath relativo

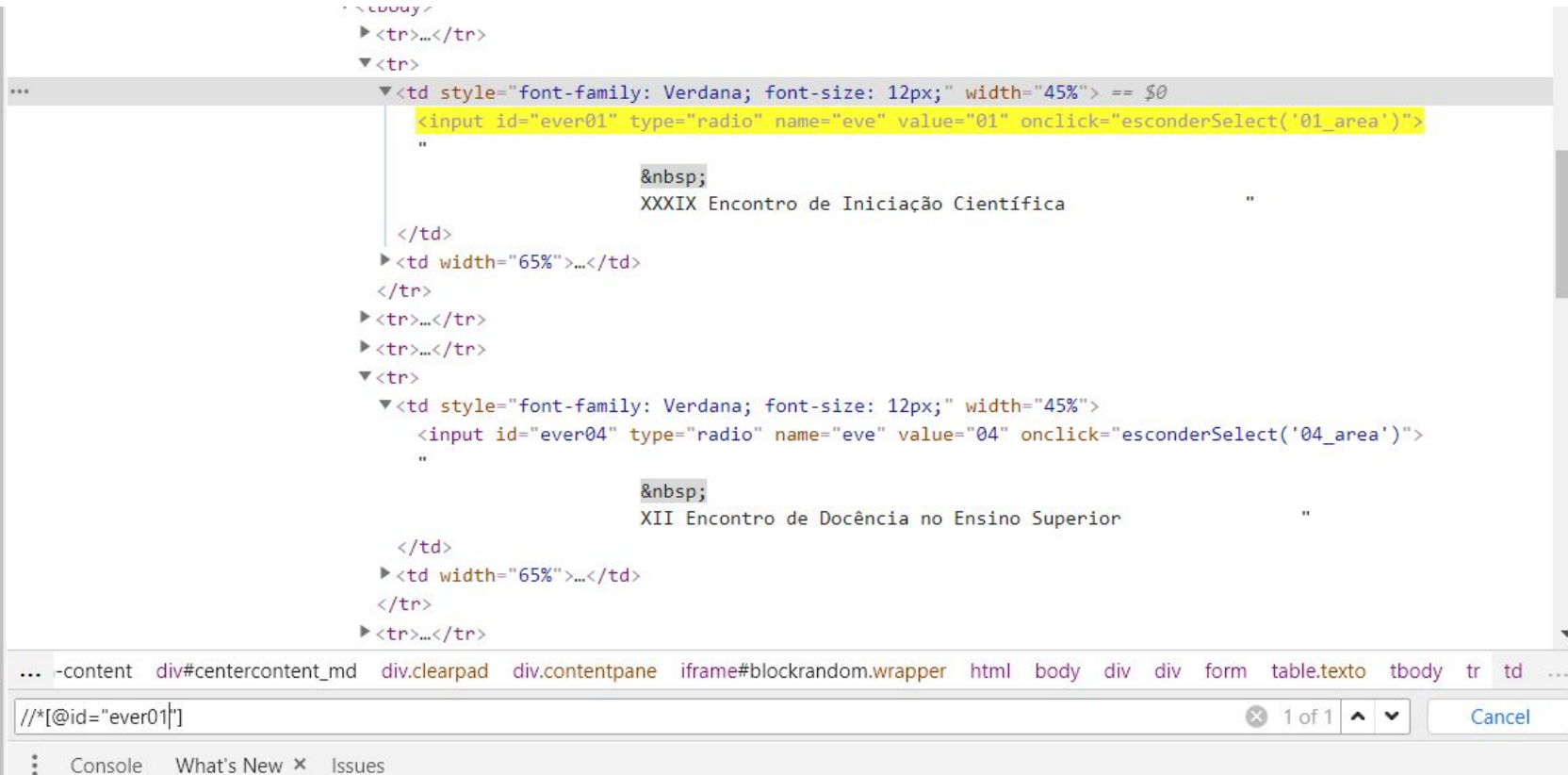


Logo of ENCONTROS UNIVERSITÁRIOS 2020, featuring a stylized 'E' and 'U' in a circle.

Encontro/Área

Resumos Cadastrados

Evento	Área / Total de Resun
<input checked="" type="radio"/> XXXIX Encontro de Iniciação Científica	Administração, Ciências Contábeis e
<input type="radio"/> XIII Encontro de Pesquisa e Pós-graduação	
<input type="radio"/> XXIX Encontro de Iniciação à Docência	
<input type="radio"/> XII Encontro de Docência no Ensino Superior	
<input type="radio"/> XXIX Encontro de Extensão	
<input type="radio"/> XIII Encontro de Experiências	



```
...<tbody>
  <tr>...</tr>
  <tr>
    <td style="font-family: Verdana; font-size: 12px;" width="45%" == $0
      <input id="ever01" type="radio" name="eve" value="01" onclick="esconderSelect('01_area')">
      "
      &nbsp;
      XXXIX Encontro de Iniciação Científica
    </td>
    <td width="65%">...</td>
  </tr>
  <tr>...</tr>
  <tr>...</tr>
  <tr>
    <td style="font-family: Verdana; font-size: 12px;" width="45%"
      <input id="ever04" type="radio" name="eve" value="04" onclick="esconderSelect('04_area')">
      "
      &nbsp;
      XII Encontro de Docência no Ensino Superior
    </td>
    <td width="65%">...</td>
  </tr>
  <tr>...</tr>
</tbody>
```

... -content div#centercontent_md div.clearpad div.contentpane iframe#blockrandom.wrapper html body div div form table.texto tbody tr td ...

//*[@id="ever01"]

1 of 1

Cancel

Console What's New Issues

Tipos de Espera

- Esperas Implícitas
- Define-se um tempo de espera máximo para encontrar qualquer elemento
- Definido após a instanciação do driver e apenas uma vez

```
driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
```

- Esperas Explícitas
- Define-se um tempo de espera e a condição esperada sobre um elemento

```
WebDriverWait waiting = new WebDriverWait(driver, 10);  
waiting.until(ExpectedConditions.visibilityOf(element));
```

Tipos de Espera

- Considerações
 - Esperas Implícitas são boas por conta da sua simplicidade
 - Nos casos em que a aplicação fica processando por tempo indefinido, onde irá cair por timeout implícito, a utilização do tempo de espera explícito é necessário.
 - Quando se utilizar tempo de espera explícito o ideal é definir o tempo de espera implícito para zero e não misturar os tempos, pois o tempo de espera fica imprevisível
-

Page Object

- Criar uma classe para cada página web
- Encapsular em cada classe os atributos e métodos relativos à página

```
public class Login{  
    public void signIn(String user, String password){  
        ...  
    }  
}
```

```
public class LoginTest{  
    @Test  
    public void testSignIn(){  
        Login loginPage = new Login();  
        loginPage.signIn("Elyfran","123");  
        ...  
    }  
}
```

Boas Práticas e Page Object

- Uso de classe abstrata para classes de páginas
 - Usada para conter métodos que são recorrentes a qualquer página web
 - O ideal é que essa classe contenha o WebDriver
- Uso de classe abstrata para classes de testes
 - Usada para conter métodos de before e after frequentes aos testes como instanciar driver
 - Útil para definir um tempo de espera máximo caso um teste falhe e fique parado em um comando esperando algo

```
@Rule
```

```
public Timeout globalTimeout = Timeout.seconds(100);
```

Navegação

- Diferentes páginas
- Controle simulado de browser

<code>driver.navigate().back();</code>	<code>//voltar</code>
<code>driver.navigate().forward();</code>	<code>//avançar</code>
<code>driver.navigate().refresh();</code>	<code>//recarregar</code>

- Execução de scripts nos browsers
 - Google Chrome
 - Internet Explorer
 - Realizar rápidas ações utilizando os comandos informados
-

Alertas, Confirmações e Prompts

- **Alertas**

Uma página incorporada nesta página diz
Hello! I am an alert!

OK

- **Confirmações**

Uma página incorporada nesta página diz
Hello! I am a confirm. It's ok?

OK

Cancelar

- **Prompts**

Uma página incorporada nesta página diz
Please enter your name

Elyfran

OK

Cancelar

```
//para clicar em 'Cancelar'  
driver.switchTo().alert().dismiss();
```

```
//para clicar em 'OK'  
driver.switchTo().alert().accept();
```

```
//para capturar a mensagem de alerta  
driver.switchTo().alert().getText();
```

```
//para escrever texto no prompt  
driver.switchTo().alert().sendKeys("Text");
```

Trabalho Prático - TP6

- TRABALHO PRÁTICO - TP6 (Aula Prática 25/01/2020)
 - Criar testes automatizados funcionais no código do projeto final
 - Escrever **pelo menos dois casos de testes** completos (com asserts) com o Selenium
 - Resposta no Google Classroom
 - Deve ser individual
 - Indicar Commit
 - Enviar video

PS: Aproveitem também para evoluir o código do projeto final

Obrigado!

Por hoje é só pessoal...

Dúvidas?



qpg4p5x



ismaylesantos@great.ufc.br



@IsmayleSantos
