Codificando o teste com Selenium WebDriver - Apêndice

O método **findElement** provavelmente é o método que você mais vai utilizar durante sua trajetória na automação de testes com Selenium WebDriver. Isso porque ele retorna um objeto do tipo **WebElement** que representa um objeto qualquer contido na página, por exemplo, um botão, um campo, uma caixa de textos, etc.

Alguns dos métodos mais utilizados em um WebElement são:

click()

Para clicar em um determinado elemento.

sendKeys(String texto)

Para atribuir um texto a um elemento, esse método pode ser mesclado com o uso da classe **Keys**, que possibilita, por exemplo, o digitar de teclas especiais como Shift, Enter, Backspace, etc.

https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/Keys.html

getText()

Retorna o texto contido entre o abrir e fechar da tag. No código <a id="mensagem">Este é meu texto</a>, o getText() retornaria "Este é meu texto".

clear()

Limpa um campo que já possui um valor default.

getAttribute(String atributo)

Retorna o texto contido em um atributo. No código <a id="mensagem">Este é meu texto</a>, o *getAttribute*("id") retornaria "mensagem".

**Você pode ter acesso a todos os métodos da classe WebElement através do link abaixo:**

<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/WebElement.html>

Usando @Before e @After do JUnit - Apêndice

Além das anotações @Test, @Before e @After ainda há outras três anotações, vejamos no próximo parágrafo.

**@BeforeClass**

Método público, sem retorno e estático usado para executar comandos antes da execução do primeiro teste contido no projeto.

**@AfterClass**

Método público, sem retorno e estático usado para executar comandos depois da execução do último teste contido no projeto.

**@Ignore(String motivo)**

Permite que testes sejam ignorados.