**Comandos de Selenium**

**Métodos de Búsqueda**

*WebDriver*: representa el navegador

*WebElement*: representa un nodo DOM particular (un control, por ejemplo, un enlace o campo de entrada, etc.)

|  |  |
| --- | --- |
| class name | Localiza elementos cuyo nombre de clase contiene el valor de búsqueda (no se permiten nombres de clase compuestos) |
| css selector | Localiza elementos que coinciden con un selector CSS |
| Id | Localiza elementos cuyo atributo ID coincide con el valor de búsqueda |
| Name | Localiza elementos cuyo atributo NAME coincide con el valor de búsqueda |
| Link text | Localiza elementos de anclaje cuyo texto visible coincide con el valor de búsqueda. |
| Partial link text | Localiza elementos de anclaje cuyo texto visible contiene el valor de búsqueda. Si varios elementos coinciden, solo se seleccionará el primero. |
| Tag name | Localiza elementos cuyo nombre de etiqueta coincide con el valor de búsqueda |
| xpath | Localiza elementos que coinciden con una expresión XPath |

La siguiente manera de buscar los elementos es por partes dos partes, la primera línea es una manera general en el nodo del DOM y la segunda línea es una búsqueda en un contexto restringido

*WebElement cheese = driver.findElement(By.id("cheese"));*

*WebElement cheddar = cheese.findElement(By.id("cheddar"));*

*findElements(By):* La versión pluralizada del comando nos devuelve una colección de elementos web, si llegara a ser solo un elemento se enviaría este como listas y si no hay elementos se devuelve vacía.

**Navegación Dentro Del Navegador**

Abrir un sitio Web:

* driver.get("https://selenium.dev"); / Conveniente
* driver.navigate().to("https://selenium.dev"); / de manera explicita

driver.getCurrentUrl(); / Puede leer la dirección en la barra de direcciones

driver.navigate().back(); / Para regresar a la página anterior

driver.navigate().forward(); / Para avanzar a la siguiente pagina

driver.navigate().refresh(); / Refrescar la página en el navegador

driver.getWindowHandle(); / Puede obtener el identificador de ventana, de la ventana actual

Cambiar ventanas o pestañas:

* String originalWindow = driver.getWindowHandle(); / Se almacena el ID de la ventana original
* Ciclo hasta que encuentre una nueva ventana:

*for (String windowHandle : driver.getWindowHandles()) {*

*if(!originalWindow.contentEquals(windowHandle)) {*

*driver.switchTo().window(windowHandle);*

*break;*

*}*

*}*

driver.switchTo().newWindow(WindowType.TAB); / Abre una nueva pestaña y se pasa a ella

driver.switchTo().newWindow(WindowType.WINDOW); / Abre una nueva ventana y se pasa a ella

driver.close(); / Cierra una pestaña o ventana

driver.switchTo().window(originalWindow); / Al cerrar cambia a la ventana original \* *Olvidar cambiar a otro identificador de ventana después de cerrar una ventana dejará WebDriver ejecutándose en la página ahora cerrada, y desencadenará un “****No Such Window Exception”*** *interrumpiendo su ejecución.*

driver.quit(); / La función de salir, Cierra todas las ventanas y pestañas asociadas con esa sesión de WebDriver, Cierra el proceso del navegador, Cierra el proceso del controlador de fondo.

**Cambio en Frames**

Hay tres formas de cambiar de frame:

A través de un elemento Web:

*WebElement iframe = driver.findElement(By.cssSelector("#modal>iframe")); / Se almacena el elemento*

*driver.switchTo().frame(iframe); / Se salta dentro del frame*

A través de un ID o nombre:

*driver.switchTo().frame("buttonframe"); / a través del ID*

*driver.switchTo().frame("myframe"); / a través del name*

A través de un indice:

*driver.switchTo().frame(1); / Se le indica el índice al cual saltar*

Para poder regresar al frame anterior:

*driver.switchTo().defaultContent();*

**Gestión de Ventanas**

Para obtener el tamaño de una ventana, manera individual:

*int width = driver.manage().window().getSize().getWidth(); / Anchura de la ventana*

*int height = driver.manage().window().getSize().getHeight(); / Altura de la ventana*

Para obtener el tamaño de una ventana, almacenándola y luego llamándolo:

*Dimension size = driver.manage().window().getSize(); Se almacenan en una variable la anchura y altura*

*int width1 = size.getWidth();*

*int height1 = size.getHeight();*

Para establecer el tamaño de la ventana:

*driver.manage().window().setSize(new Dimension(1024, 768));*

Para obtener la posición de una ventana, manera individual (Obtiene las coordenadas de la coordenada superior izquierda de la ventana del navegador):

*int x = driver.manage().window().getPosition().getX();*

*int y = driver.manage().window().getPosition().getY();*

Para obtener la posición de una ventana, almacenándola y luego llamándolo (Obtiene las coordenadas de la coordenada superior izquierda de la ventana del navegador):

*Point position = driver.manage().window().getPosition();*

*int x1 = position.getX();*

*int y1 = position.getY();*

Establecer la posición de la ventana:

*driver.manage().window().setPosition(new Point(0, 0));*

Para maximizar la ventana:

*driver.manage().window().maximize();*

Para que la ventana se ve en pantalla completa:

*driver.manage().window().fullscreen();*

**Esperas (Waits)**

Wait Explicito:

*WebElement foo = new WebDriverWait(driver, 3).until(driver -> driver.findElement(By.name("q")));*

*//new WebDriverWait(driver,3).until(some\_condition(WebElement))*

*new WebDriverWait(driver, 3).until(ExpectedConditions.elementToBeClickable(By.xpath("//a/h3")));*

Algunas condiciones esperadas:

* alert is present
* element exists
* element is visible
* title contains
* title is
* element staleness
* visible text

Wait Implicito:

*driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);*

Wait Fluido:

// Esperando 30 segundos para que un elemento esté presente en la página, verificando su presencia una vez cada 5 segundos.

*Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)*

*.withTimeout(Duration.ofSeconds(30))*

*.pollingEvery(Duration.ofSeconds(5))*

*.ignoring(NoSuchElementException.class);*

**Alertas JavaScript, avisos y confirmaciones**

Alertas:

*Alert alert = wait.until(ExpectedConditions.alertIsPresent()); / Espere a que se muestre la alerta y almacénela en una variable*

*String text = alert.getText(); / Almacenar el texto de alerta en una variable*

*alert.accept(); / Presione el botón OK*

*Confirmar:*

*wait.until(ExpectedConditions.alertIsPresent()); / Espere a que se muestre la alerta*

*Alert alert = driver.switchTo().alert(); / Almacenar la alerta en una variable*

*String text = alert.getText(); / Almacene la alerta en una variable para reutilizar*

*alert.dismiss(); / Presione el botón Cancelar*

Prompt:

*Alert alert = wait.until(ExpectedConditions.alertIsPresent()); / Espere a que se muestre la alerta y almacénela en una variable*

*alert.sendKeys("Selenium"); / Escribe tu mensaje*

*alert.accept(); / Presione el botón OK*