

1. Crear el Objeto Driver basado en el explorador elegido

Creemos una clase de Java en el proyecto anteriormente generado en IntelliJ y sobre la que vamos a trabajar.

Recordemos que lo primero que debemos hacer es crear la Property para el WebDriver a usar, asignándole la **Key** del navegador a emplear y el **Valor** que será la ruta de su ubicación.

WebDriver de Chrome

```
System.setProperty("webdriver.chrome.driver","/Users/Fbogas/Desktop/Sikuli/chromedriver");
```

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

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

```
System.out.println(driver.getTitle());
```

Con el Script anterior generamos un WebDriver llamado **driver** para controlar el navegador Chrome y le indicamos que alcance la web de Google. Posteriormente se escribe en la consola de comandos de IntelliJ el título asignado a la web (¡OJO! que no su URL).

WebDriver de Firefox

```
System.setProperty("webdriver.gecko.driver","/Users/Fbogas/Desktop/Sikuli/geckodriver");
```

Creemos el objeto driver. Para ello accedemos a la web de selenium para comprobar la clase que debemos utilizar para crear el objeto correspondiente al navegador seleccionado. En este caso Firefox. <https://www.seleniumhq.org/projects/webdriver/>

```
WebDriver driver = new FirefoxDriver(); driver.get("http://google.com");
```

```
System.out.println(driver.getTitle());
```

Error típico en el código de invocación del explorador:

Si cometemos algún error al invocar el explorador recibiremos un mensaje de error del estilo:

The path to the driver executable must be set by the webdriver.xxxxx.driver system...

Normalmente el error será porque la ruta de ubicación del webdriver no es correcta, o el webdriver no es el del navegador invocado.

Otras veces puede ser que Selenium no de soporte a las últimas versiones del navegador mediante el que se quieren automatizar las pruebas (Actualmente ocurre con el navegador Edge bajo OS X).

La mayoría de las veces que ocurre esto se da en las últimas versiones del navegador Firefox.

El navegador con mayor compatibilidad con Selenium en la actualidad es Chrome.

2. Ejemplos para poner en practica lo visto hasta ahora.

Un primer ejercicio y al mismo tiempo, una buena práctica, es determinar si hemos aterrizado en la web correcta o no. Hasta ahora hemos estado usando:

```
System.out.println(driver.getTitle());
```

Pero este paso no es suficiente para determinarlo. ¿Cómo leer entonces la URL actual en Selenium?

Antes de dar la solución quiero recomendar al lector que se valga de la ayuda de IntelliJ para ver los métodos de la clase driver, escribiendo:

```
System.out.println(driver.
```

Y viendo las opciones mostradas intenta deducir cuál sería la que buscamos.

Si has hecho este ejercicio habrás llegado fácilmente a la solución:

```
System.out.println(driver.getCurrentUrl());
```

Mirando en la consola de IntelliJ tras la ejecución observaremos que nos muestra la URL exacta del sitio web:

```
> Task :Test.main()
Starting ChromeDriver 75.0.3770.140 (2d9f97485c7b07dc18a74666574f19176731995c-refs/branch-heads/3770@{#1155}) on port 41688
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
jul. 25, 2019 11:01:52 A. M. org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
https://www.google.com/7qws\_rd=ssl
```

Con el comando actual verificamos que hemos aterrizado justo en la web que queríamos.

Hay veces que el desarrollador o propietario de la web desactiva que hagas click derecho en la web para impedirte que puedas conocer el código fuente de la web. Con Selenium esto no es problema ya que podemos usar:

```
System.out.println(driver.getPageSource());
```

[illegible]

3. Métodos Básicos con webDriver II

¿Qué ocurre si le decimos a Selenium que alcance 2 webs diferentes a la vez?

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

```
driver.get("http://yahoo.com");
```

Si has probado a ejecutarlo en IntelliJ habrás comprobado que carga las URL en orden secuencial tal como las encuentra pero en la misma ventana. Con lo que en el caso anterior primeramente cargará *Google* y posteriormente *Yahoo!*.

¿Cómo podemos hacer para volver a la página anteriormente cargada en el navegador?

```
driver.navigate().back();
```

Hasta ahora hemos realizado varios ejemplos y si nos has seguido habrás visto que una vez ejecutadas las pruebas el navegador se quedaba abierto. Para cerrar la sesión en el navegador una vez finalizadas las pruebas:

```
driver.close(); // Cierra el navegador actual abierto por el Script
```

```
driver.quit(); // Cierra todas las sesiones abiertas por el Script
```

4. Técnicas de Localización (Locator) y Herramientas utilizadas para identificar un Objeto

Para proceder con las técnicas *Locator* vamos a acceder a una web cualquiera, por ejemplo, facebook. Una vez en ella vamos a observar sus elementos y cómo los desarrolladores identifican los elementos contenidos en ella.

Vamos a seleccionar el elementos de **login de email or phone > click derecho > inspeccionar**.



Al hacer click se nos desplegará la versión web para desarrollador. En ella aparece el código fuente de la web y resaltado el mismo código fuente que hace referencia al elemento seleccionado.



Es importante aquí reconocer varias partes del código:

```
<input type="text" class="inputtext _58mg_5dba_2ph-" data-type="text" name="firstname"
```

```
value="" aria-required="true" placeholder="" aria-label="Nombre" id="u_0_l"
```

```
style="background-image: XXXXX background-repeat: no-repeat; background-attachment:
```

```
scroll; background-size: 16px 18px; background-position: 98% 50%; cursor: auto;">
```

En verde tenemos la etiqueta (tag) y en azul oscuro los atributos.

Conozcamos ahora los Locators soportados por Selenium:

- ID
- ClassName
- Name
- LinkText
- Xpath
- CSS

Si echamos un vistazo a la etiqueta anterior vemos que podemos identificar el ID, el ClassName y el Name claramente del objeto.

Vamos a hacer un ejercicio en el que vamos a introducir nuestro email en dicha caja de texto.

```
System.setProperty("webdriver.chrome.driver","/Users/Fbogas/Desktop/Sikuli/chromedriver");
```

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

```
driver.get("http://facebook.com"); //URL in the browser
```

```
driver.findElement(By.id("email")).sendKeys("micorreo@autentia.com");
```

```
driver.findElement(By.id("pass")).sendKeys("mipassword");
```

```
//driver.findElement(By.id("u_0_a")).click();
```

```
// ¡OJO! Con usar ID alfanuméricos ya que cambian con cada refresh de la web
```

```
driver.findElement(By.id("loginbutton")).click();
```

```
// Pero ¿Qué Ocurre cuando no disponemos ninguno de los locators?
```

Veamos ahora otro ejemplo pero utilizando el LinkText. En la misma web de Facebook vemos que existe un link que dice: *¿Has olvidado los datos de la cuenta?*

Con el siguiente comando podemos hacer click en los enlaces:

```
driver.findElement(By.linkText("¿Has olvidado los datos de la cuenta?")).click();
```

NOTA: cuando buscamos un elemento por clase debemos tener en cuenta que no se pueden buscar cuando el nombre de la clase tiene espacios. Por ejemplo:

```
class ="input r4 wide mb16 mt8 username"
```

Si lo usamos obtendremos el mensaje de error:

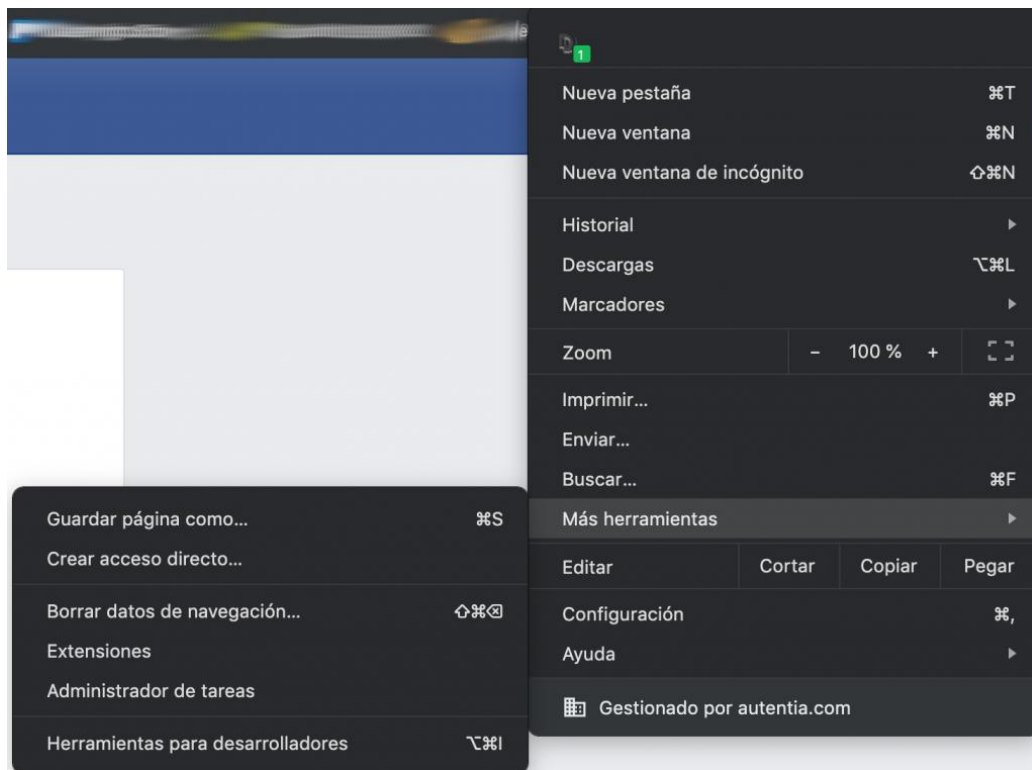
Compound class names not permitted

5. Identificando XPath en los navegadores Chrome y Firefox. Utilizando las herramientas del navegador.

¿Qué es XPATH?

XPath (*XML Path Language*) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. La idea es parecida a las [expresiones regulares](#) para seleccionar partes de un texto sin atributos (*plain text*). Este nos permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. XPath fue creado para su uso en el estándar [XSLT](#), en el que se usa para seleccionar y examinar la estructura del documento de entrada de la transformación. Por último, indicar que fue definido por el consorcio [W3C](#).

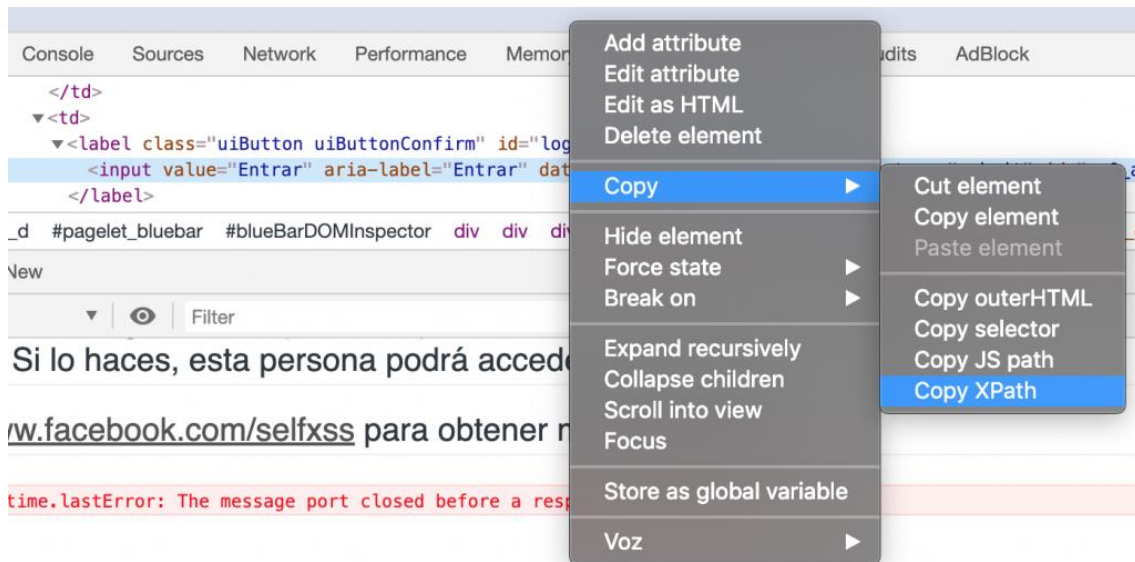
Para poder utilizarlo en un navegador, primero debemos tener activadas las **Herramientas para desarrolladores**.



Posteriormente para conocer el XPATH del elemento en cuestión lo seleccionamos.



Veremos que el código fuente queda resaltado. En este código fuente hacemos click con el botón derecho del ratón y seleccionamos **Copy > Copy XPath**.



De esta forma ya tendremos en el portapapeles el XPath a usar:

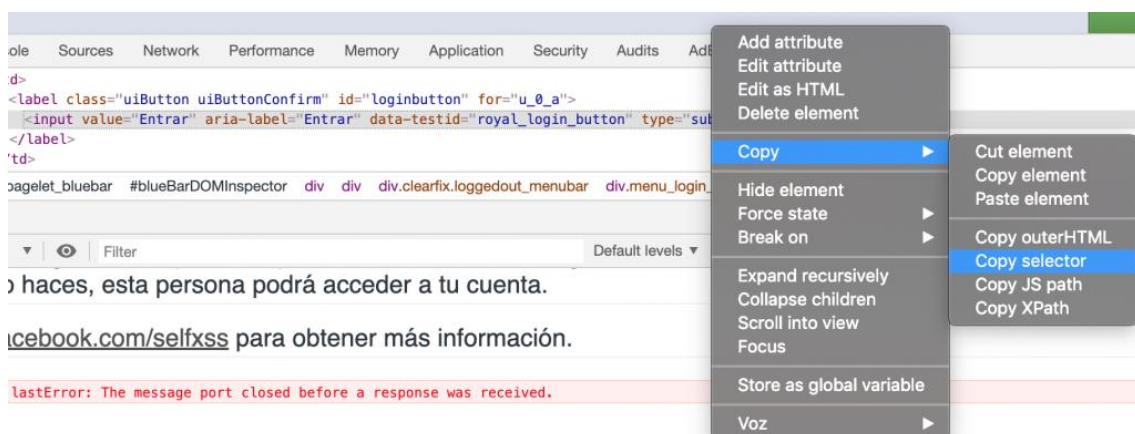
```
#u_0_a
```

```
driver.findElement(By.xpath("//*[@id=\"u_0_a\"]")).click();
```

¡OJO! se ha puesto la \ como símbolo de escape. En otro caso se debería escribir con comillas simples: `By.xpath(<code>\"/>`

6. Identificando CSS selector en los navegadores Chrome y Firefox. Utilizando las herramientas del navegador.

CSS Locator es similar a XPath. En este caso procederemos de igual forma que hacemos para obtener el XPath. Pero en esta ocasión seleccionamos Copy Selector en Chrome y Copy CSS Selector en Firefox.



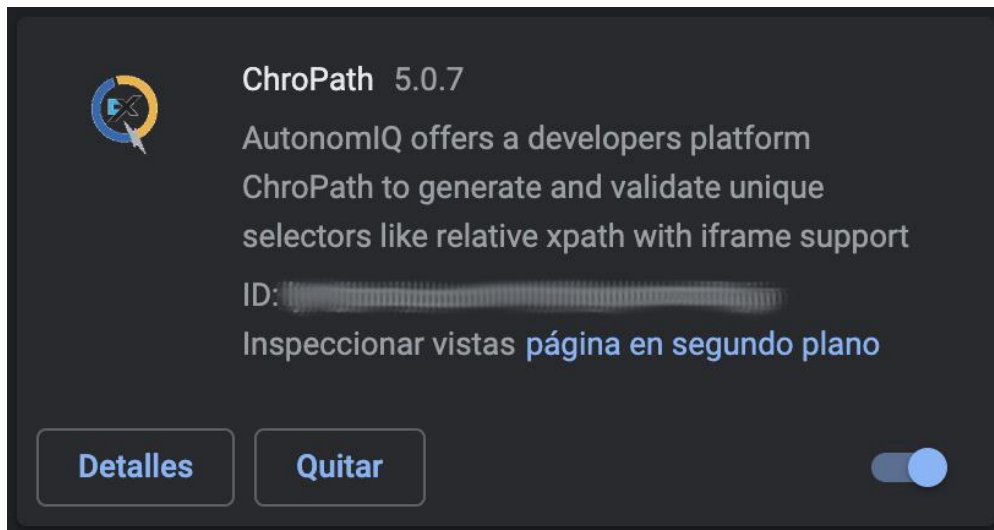
Obteniendo:

```
#u_0_a
```

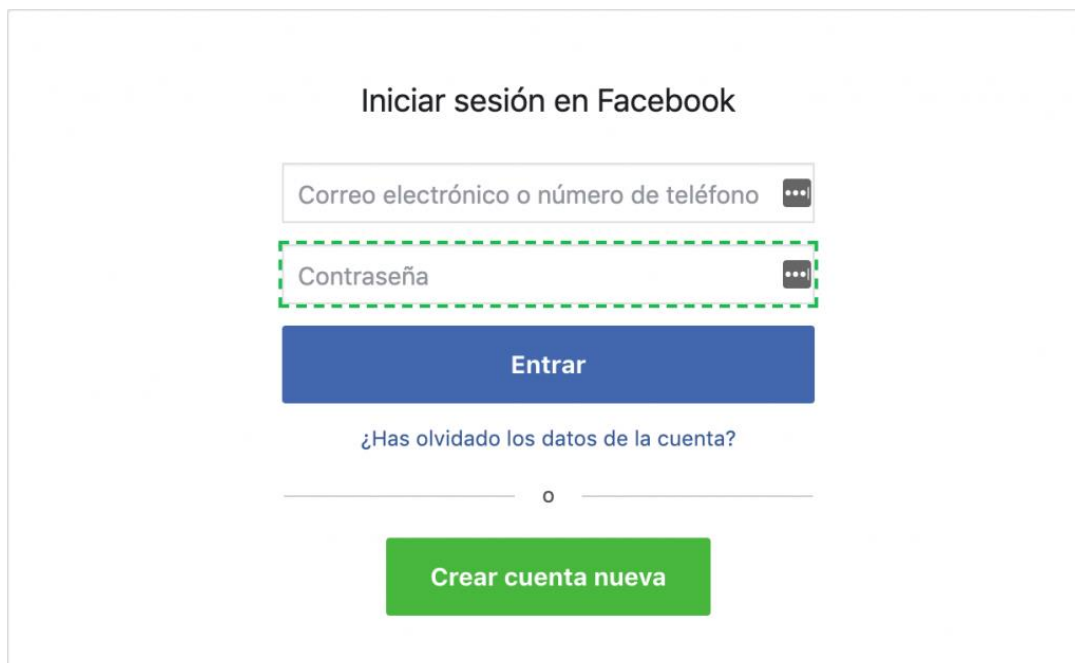
Aunque también podemos obtenerlo buscando en la ToolBar de las herramientas de desarrollador.

7. ChroPath, la extensión del navegador Chrome que te facilitará la vida.

Tanto como para Chrome como para Firefox existen extensiones o plugin que te pueden facilitar la vida más de lo que te imaginas. En el momento de escribir el artículo la última versión de ChroPath es soportada por Firefox, Chrome y Opera:



Lo que nos trae esta extensión es tener al alcance de un click los Locators de todos los objetos de la web. Si en el ejemplo anterior de Facebook hacemos click sobre el casillero del password:



Veremos en el ChroPath todos los Locators:



Fácil, ¿verdad?

Por tanto, cerraremos este segundo post con un ejercicio para poner en práctica todo lo aprendido hasta ahora que no es poco.

8. Ejercicio Práctico

Para este ejercicio se solicita capturar un mensaje de error y mostrarlo por la consola de comandos. Para ello en la página web de Facebook vamos a loguearnos con un usuario ficticio y vamos a capturar el mensaje de error mostrado.

Una posible solución al ejercicio es:

```
public static void main(String[] args) {
```

```
System.setProperty("webdriver.chrome.driver","/Users/Fbogas/Desktop/Sikuli/chromedriver");
```

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

```
driver.get("http://facebook.com");//URL in the browser
```

```
driver.findElement(By.id("email")).sendKeys("micorreo@autentia.com");
```

```
driver.findElement(By.id("pass")).sendKeys("mipassword");
```

```
driver.findElement(By.xpath("//*[@id=\"u_0_a\"]")).click();
```

```
System.out.println(driver.findElement(By.xpath("//div[@class='_4rbf_53ij']")).getText());
```

```
}
```