



DESCRIPCIÓN BREVE

Se realiza el informe de la de la tarea de pruebas básicas con Selenium

[Ariagna Albuerne](#)

Taller de automatización del Testing Funcional

Contenido

INTRODUCCIÓN	2
DESARROLLO	3
Configuración POM Del proyecto	5
Parte B1	6
Indicaciones para ejecución con clave y contraseña.....	6
.....	8
Herramientas Utilizadas	8
CONCLUSIONES	9

INTRODUCCIÓN

En el informe se muestra los elementos que se tuvieron en cuenta para darle solución a la consigna propuesta '**Pruebas básicas con Selenium**' y una descripción de cada una de sus partes.

DESARROLLO

- Parte A - Crear pruebas básicas con Selenium WebDriver para las siguientes funcionalidades:

1. Búsqueda en Google
2. Búsqueda del artículo "Hola mundo" en Wikipedia

Se realiza una sola clase con dos test para el caso de esta parte.

1. Búsqueda en Google (Se realiza la búsqueda del artículo Hola mundo de wikipedia)

Guión

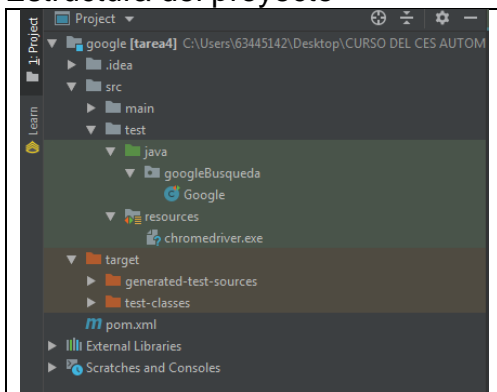
1. Acceder a la URL " https://www.google.com"
2. Escribir en el cuadro de búsqueda "Hola mundo"
3. Dar clic en el link de Wikipedia de "Hola mundo"
4. Verificar que se accedió al artículo Hola mundo de wikipedia

3. Búsqueda del artículo "Hola mundo" en Wikipedia

Guión

1. Acceder a la URL " https://es.wikipedia.org/"
2. Escribir en el cuadro de búsqueda "Hola mundo"
3. Realizar la búsqueda
4. Verificar que se accedió al artículo Hola mundo de wikipedia

Estructura del proyecto



Se realizó la parte A en una misma clase debido a que se pensó en dos vías de encontrar el mismo artículo, pensándolo como dos casos de prueba, tanto por la búsqueda en google (Parte A 1), cómo por la búsqueda dentro de Wikipedia (Parte A 2.)

```

package googleBusqueda;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.TestInstance;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

import static org.junit.jupiter.api.Assertions.assertEquals;

import java.util.concurrent.TimeUnit;

@TestInstance(TestInstance.Lifecycle.PER_CLASS)

public class Google {

    private WebDriver driver;
    By link = By.cssSelector( ".g:nth-child(1) > div:nth-child(2) .LC20lb" );

    @BeforeAll
    public void setUp () {
        System.setProperty( "webdriver.chrome.driver" , "../src/test/resources/chromedriver.exe" );
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }

    @Test
    public void testGoogle () {
        driver.get( "http://www.google.com" );
        WebElement busqueda = driver.findElement( By.name( "q" ) );
        busqueda.clear();
        busqueda.sendKeys( "Hola mundo" );
        busqueda.submit();
        driver.manage().timeouts().implicitlyWait( 10 , TimeUnit.SECONDS );
        driver.findElement( link ).click();
        assertEquals( "Hola mundo - Wikipedia, la enciclopedia libre" , driver.getTitle() );
    }

    @Test
    public void testWiki () {
        driver.get( "https://es.wikipedia.org/" );
        WebElement busqueda = driver.findElement( By.id( "searchInput" ) );
        busqueda.clear();
        busqueda.sendKeys( "Hola mundo" );
        busqueda.submit();
        driver.manage().timeouts().implicitlyWait( 10 , TimeUnit.SECONDS );
        assertEquals( "Hola mundo - Wikipedia, la enciclopedia libre" , driver.getTitle() );
    }

    @AfterAll
    public void tearDown() {
        driver.quit();
    }
}

```

Configuración POM Del proyecto

Se realiza la configuración del POM para definir las dependencias a utilizarse:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>tarea4</groupId>
  <artifactId>tarea4</artifactId>
  <version>1.0-SNAPSHOT</version>
  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
  <dependencies>

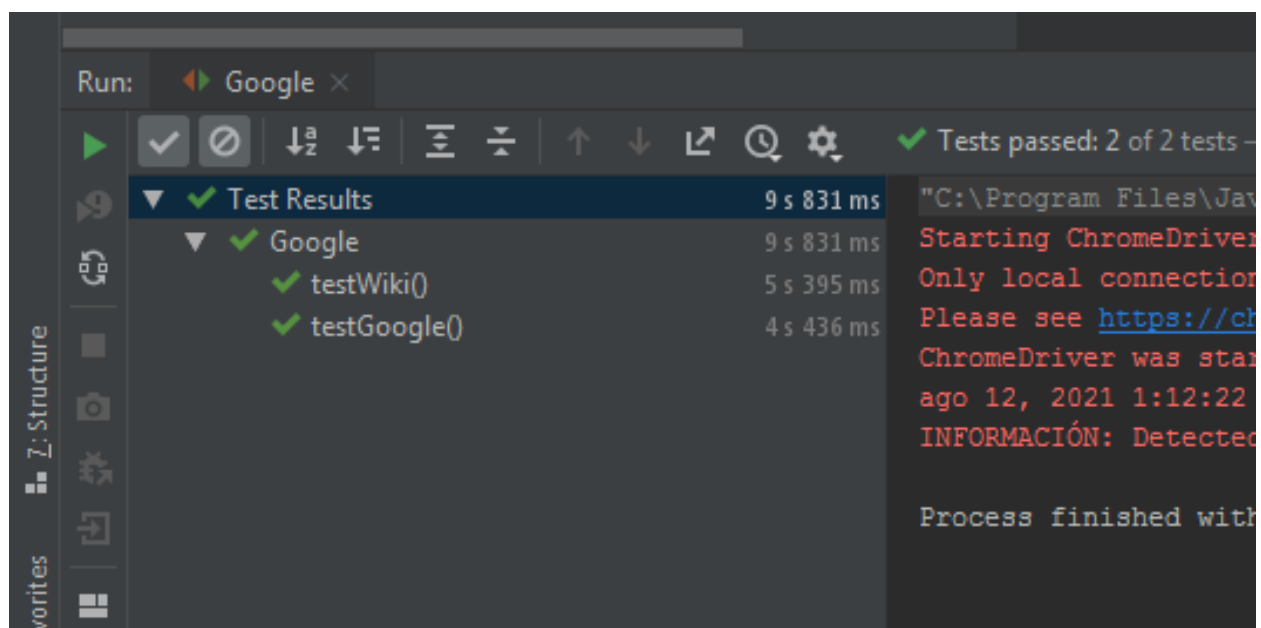
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
  </dependency>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-chrome-driver</artifactId>
    <version>3.141.59</version>
  </dependency>

  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.8.0-M1</version>
    <scope>test</scope>
  </dependency>

  </dependencies>

</project>
```

Captura de pantalla de todos los test estado: passed



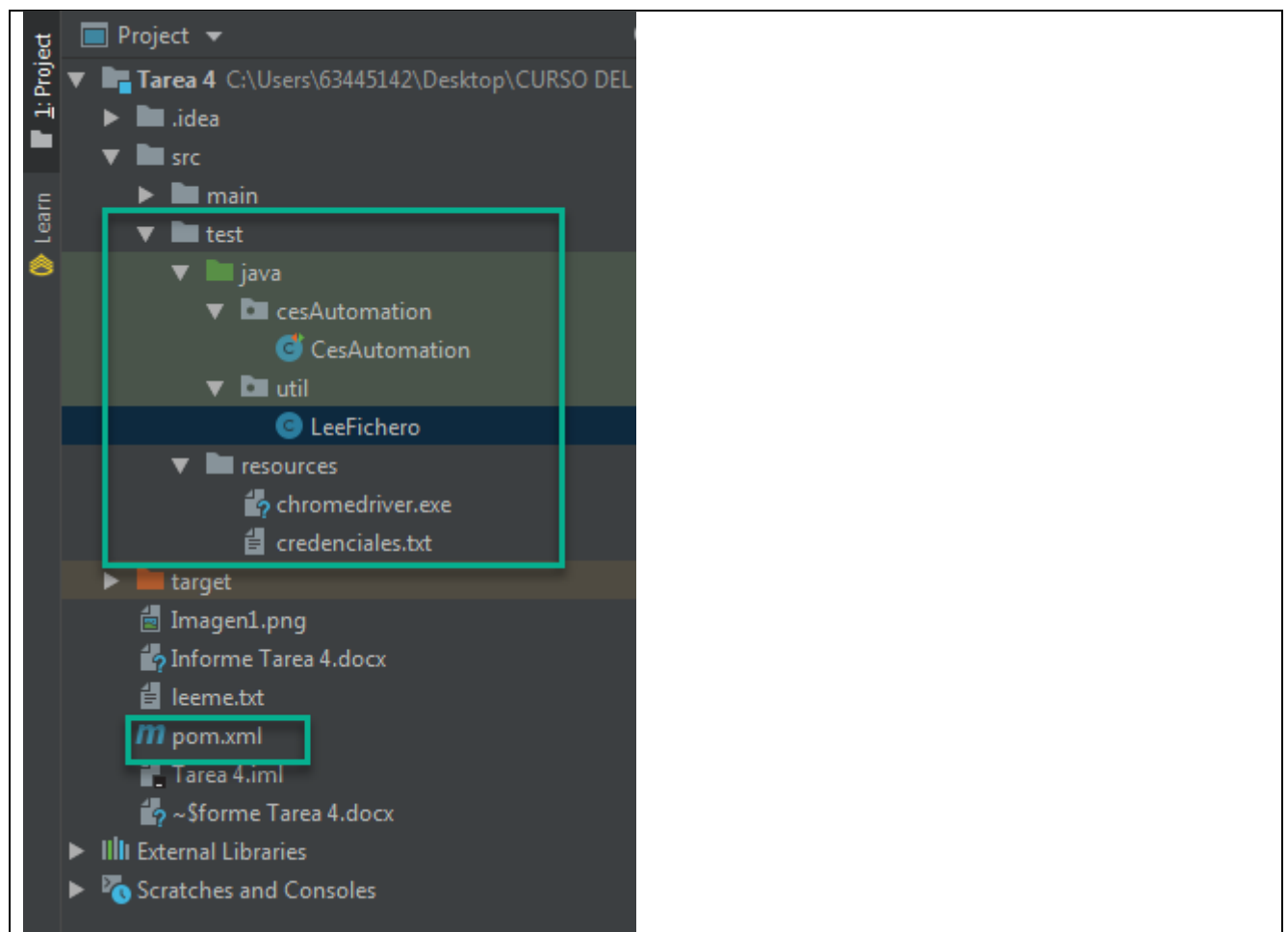
Parte B1

- Automatizar con Selenium WebDriver el siguiente guion
 1. Acceder a la URL "capacitacion.ces.com.uy"
 2. Iniciar sesión con un usuario y contraseña existentes y habilitados
 3. Acceder al curso "Taller de Automatización del Testing Funcional" (que se encuentra en la sección "Mis Cursos")
 4. En la sección "Buscar en los foros", buscar un foro que contenga la palabra "Bienvenida"

Indicaciones para ejecución con clave y contraseña.

Se entrega un fichero leeme.txt con las indicaciones.

Estructura del proyecto



Solución

```

package cesAutomation;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.TestInstance;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;
import static org.junit.jupiter.api.Assertions.assertEquals;
import java.util.ArrayList;
import util.LeeFichero;

@TestInstance(TestInstance.Lifecycle.PER_CLASS)
public class CesAutomation {

    private WebDriver driver;
    By link = By.cssSelector( ".g:nth-child(1) > div:nth-child(2) .LC201b" );
    By linkAcceder = By.linkText( "Acceder" );
    By nombre = By.id( "username" );
    By password = By.id( "password" );
    By entrar = By.id( "loginbtn" );
    By paginaCursos = By.linkText( "INTRODUCCION A LA AUTOMATIZACION DEL TESTING FUNCIONAL
202103" );
    By foro = By.cssSelector( "#module-21581 .instancename" );
    By linkBienvenida = By.linkText( "Bienvenida" );

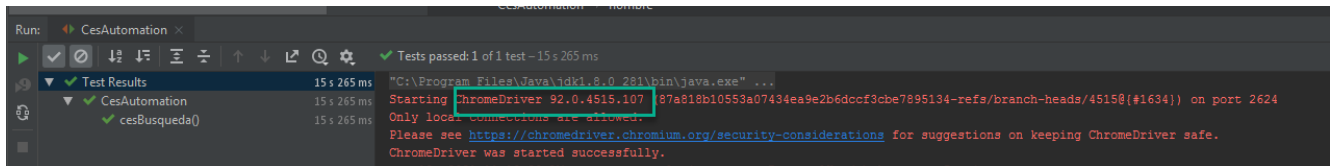
    @BeforeAll
    public void setUp () {
        System.setProperty( "webdriver.chrome.driver" , "../src/test/resources/chromedriver.exe" );
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }

    @Test
    public void cesBusqueda () {
        LeeFichero credencial= new LeeFichero();
        ArrayList<String> credenciales=credencial.leerFichero();
        String user=credenciales.get( 0 );
        String contra=credenciales.get(1);
        driver.get( "https://capacitacion.ces.com.uy" );
        driver.manage().timeouts().implicitlyWait( 5 , TimeUnit.SECONDS );
        driver.findElement( linkAcceder ).click();
        WebElement username = driver.findElement( nombre );
        username.clear();
        username.sendKeys(user);
        WebElement clave = driver.findElement( password );
        clave.clear();
        clave.sendKeys(contra );
        WebElement acceder = driver.findElement( entrar );
        acceder.submit();
        WebElement pagCursos = driver.findElement( paginaCursos );
        pagCursos.click();
        WebElement novedades = driver.findElement( foro );
        novedades.click();
        assertEquals( "Novedades" , driver.getTitle() );
        WebElement bienvenida = driver.findElement( linkBienvenida );
        assertEquals( "Bienvenida" , bienvenida.getText() );
    }

    @AfterAll
    public void tearDown() {
        driver.quit();
    }
}

```


Captura del test pasando y con la versión de Chrome utilizada.



Herramientas Utilizadas

- IntelliJ IDEA Community Edition
- Java versión 1.8.0_281
- Maven
- JUnit 5 (junit-jupiter-engine 5.8.0-M1)
- Selenium (3.141.59)

Parte B2

¿Es posible automatizar la descarga del archivo PDF "Material cubrimiento" del "Módulo 3" solamente utilizando Selenium? Justifique.

Entiendo que no es posible solamente con Selenium, hay que agregar configuraciones específicas de dependencia del navegador, agregando configuración específica del navegador (en Firefox agregando Firefox Profile y seteando configuración específica...) Chrome parte de lo que podría hacerse es activar la descarga automática desde la configuración del navegador, pero no es lo único.

CONCLUSIONES

Con la realización de este ejercicio se pudo comprobar que:

- 1- Es útil apoyarse de herramientas como selenium IDE para obtener los selectores.
- 2- Hay diferentes tipos de selectores
- 3- La importante la utilización de Junit 5 con Selenium