Dando continuidad al tutorial anterior referente a la instalación de IntelliJ IDEA, en este tutorial veremos crear un test para abrir un sitio web, localizar un link y acceder al mismo.

Para comenzar vamos a importar el Framework TestNG, que nos permitirá trabajar con tests dentro de las clases de nuestro proyecto mediante anotaciones, parámetros, etc. Para esto accedemos al repositorio de Maven para obtener el código de dependencia.

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>6.14.3</version>
    <scope>test</scope>
</dependency>
```

Pegamos el código en el archivo POM.xml:

Para asegurarnos de que funcione correctamente, actualizamos los proyectos de Maven dando clic en el icono *«Maven»* en la parte superior derecha, y luego en el icono de *«Actualizar todos los proyectos de Maven».*

Ahora, en la clase BaseTests vamos a crear 2 parámetros, BeforeClass y AfterClass, estos nos

servirán para especificar lo que requerimos hacer antes y después de trabajar con otra clase en el mismo o en otro paquete. Puedes ver el uso completo de TestNG en la documentación oficial.

En @BerforeClass creamos un setUp para la configuración inicial de nuestro proyecto, que es el uso de la propiedad ChromeDriver y luego abrir una dirección de Internet. Después de esto, colocamos la anotación @AfterClass, en la que podemos elegir qué hacer al terminar el test, por ejemplo, cerrar el navegador. El código completo queda de la siguiente forma:

```
package base;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;
import pages.HomePage;

public class BaseTests {

    public WebDriver driver;
    protected HomePage homePage;

    @BeforeClass
    public void setUp(){
        System.setProperty("webdriver.chrome.driver",
"resources/chromedriver.exe");
```

```
            driver = new ChromeDriver();
            driver.get("http://blogs.masterhacks.net");
                }

        @AfterClass
        public void tearDown(){
        //driver.quit();
        }
    }
```

Ahora, crearemos un nuevo paquete con el nombre MHTests

Dentro de ese paquete, creamos una nueva *Java Class* llamada ClickTests. En el código que se creará, modificaremos la línea de clase pública para agregar la clase Base, mediante el parámetro *extends:*

```
public class ClickTests extends BaseTests {
```

Ahora, crearemos una marca @Test usando la biblioteca TestNG, al ingresar esto, se importa dicha biblioteca. Luego, crearemos una variable llamada PruebaBlog donde colocaremos el test que queremos realizar.

```
 @Test
    public void PruebaBlog(){
    WebElement cliclink = driver.findElement(By.id("menu-
item-10032"));
    cliclink.click();

    }
```

Aquí, se utiliza WebElement para crear una variable llamada *cliclink*, y con el método *driver.find.Element* buscaremos un elemento, en este caso, mediante el *id* del elemento. Lo que queremos hacer en el sitio web, es situarnos en el enlace Noticias del menú. Entonces, para conocer el id de este elemento del sitio web, vamos a dar clic derecho sobre el enlace y en inspeccionar.

Después, utilizamos el parámetro *click()* en la variable *cliclink* para ingresar al enlace con el ID encontrado. El código completo queda de la siguiente forma:

Cuando se corre el test, al ejecutar el contenido de *@Test* el programa regresa a BaseTests y ejecuta el contenido de *@AfterClass*. Para probar el test, damos clic derecho sobre la clase ClickTests y *Run 'ClickTests'*

Si requieres algún programa en específico, no dudes en contactarnos aquí para una cotización.

Si tienes algún comentario déjalo aquí abajo!