

## Testing II

# Atributos de las anotaciones de JUnit y TestNG

## JUnit

Algunas anotaciones de JUnit aceptan parámetros llamados **atributos** que se utilizan para proporcionar más detalles a la prueba que se está ejecutando. Selenium requiere estos parámetros para aplicar restricciones como el límite de tiempo, el número de veces que debe ejecutarse una prueba, etc. Veamos un ejemplo y exploremos algunos de estos atributos de anotación.

En el siguiente fragmento de código, se utilizan dos anotaciones **@Test** y **@RepeatedTest**. A cada uno de ellos se le ha asignado un atributo. La primera anotación **@Test** se ha establecido con el atributo **timeout** (en milisegundos) que indica a Selenium que establezca el tiempo de espera en 5 segundos. Del mismo modo, la segunda anotación **@RepeatedTest** recibe un número entero que le dice a JUnit que ejecute la prueba llamada "test\_search" seis veces.

Pasar atributos para algunas anotaciones es opcional, mientras que para otras es obligatorio. En este caso, el atributo de anotación **@Test** es opcional, en tanto que el atributo **@RepeatedTest** es obligatorio.

```
@Test(timeout=5000)
@RepeatedTest(6)
public void test_search()
{
    // code to search something using the search_bar
    driver.findElement(By.id("search_bar")).sendKeys("Browserstack automation testing");
    driver.findElement(By.name("search_btn")).click();

    String first_result = driver.findElementByXPath("/html/body/result").getText();

    assertEquals(first_result,"Testing and annotations");
}
```

# TestNG

TestNG es un marco de automatización de pruebas en el que NG significa "**Next Generation**". TestNG se inspira en JUnit, que utiliza las anotaciones (@). TestNG supera las desventajas de JUnit y está diseñado para facilitar las pruebas de extremo a extremo. Las anotaciones en TestNG son líneas de código que pueden controlar cómo se ejecutará el método que tienen debajo. Siempre van precedidos del símbolo @. A continuación se muestra un ejemplo de un TestNG muy temprano y rápido.

```
@Test(priority = 0) // example of annotations
public void goToHomepage(){
    driver.get(baseUrl);
    Assert.assertEquals(driver.getTitle(), "Welcome: Mercury Tours");
}

@Test(priority = 1) // example of annotations
public void logout(){
    driver.findElement(By.LinkText("SIGN-OFF")).click();
    Assert.assertEquals("Sign-on: Mercury Tours", driver.getTitle());
}
```

El ejemplo anterior simplemente dice que el método goToHomepage() debe ejecutarse primero antes de logout() porque tiene un número de prioridad más bajo.

## Resumen de anotaciones de TestNG

**@BeforeSuite:** el método anotado se ejecutará antes de que se ejecuten todas las pruebas de este conjunto.

**@AfterSuite:** el método anotado se ejecutará después de que se ejecuten todas las pruebas de este conjunto.

**@BeforeTest:** el método anotado se ejecutará antes de que se ejecute cualquier método de prueba perteneciente a las clases dentro de la etiqueta.

**@AfterTest:** el método anotado se ejecutará después de la ejecución de todos los métodos de prueba pertenecientes a las clases dentro de la etiqueta.

**@BeforeGroups:** la lista de grupos que este método de configuración ejecutará antes. Se garantiza que este método se ejecute justo antes de que se invoque el primer método de prueba que pertenezca a cualquiera de estos grupos.

**@AfterGroups:** la lista de grupos después de los cuales se ejecutará este método de configuración. Se garantiza que este método se ejecute justo después de que se invoque el último método de prueba que pertenezca a cualquiera de estos grupos.

**@BeforeClass:** el método anotado se ejecutará antes de que se invoque el primer método de prueba de la clase actual.

**@AfterClass:** el método anotado se ejecutará después de que se ejecuten todos los métodos de prueba de la clase actual.

**@BeforeMethod:** el método anotado se ejecutará antes de cada método de prueba.

**@AfterMethod:** el método anotado se ejecutará después de cada método de prueba.

**@Test:** el método anotado forma parte de un caso de prueba.

Las anotaciones se utilizan para describir un lote de código insertado en el programa o la lógica de negocio utilizada para controlar el flujo de los métodos en el script de prueba. Hacen que los scripts de prueba de Selenium sean más manejables, sofisticados y eficaces. Su uso es extremadamente útil para los probadores y les facilita mucho la vida.