Automation Frameworks

TestNG - Anotaciones

@Test : ejecuta el método anotado e informa su resultado. Aquí es donde se realizan las verificaciones.

@BeforeMethod: ejecuta el método anotado antes de cada método de prueba. Aquí debe haber todo el código de creación de instancias que debe actualizarse antes de ejecutar cada prueba.

@AfterMethod: ejecuta el método anotado después de cada método de prueba. Suele utilizarse para liberar recursos que ya no se necesitan después de ejecutar cada prueba.

@BeforeClass: ejecuta el método anotado antes de crear una instancia de la clase de prueba. Aquí deberíamos crear una instancia de todo lo que sea necesario en todas las pruebas.

@AfterClass: ejecuta el método anotado después de ejecutar toda la clase de prueba. Aquí liberamos los recursos que ya no se utilizarán después de ejecutar todas las pruebas..

IMPORTANT: importa las anotaciones de org.testng.annotations, ¡no las de junit!

TestNG - Ejemplo @ Test

```
@Test

public void nombreTest() {

        //código

        ....

        System.out.println(“Esto es un test”);

        ....

        // código

}
```

TestNG - Maven dependency

https://mvnrepository.com/artifact/org.testng/testng/7.4.0

```
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.4.0</version>
  <scope>test</scope>
</dependency>
```

TestNG - Ejercicio 1

Escriba una clase de prueba que contenga una línea de impresión con el nombre de la anotación que se utiliza.

La clase de prueba debe contener cada una de las anotaciones mencionadas anteriormente.

Resultado esperado:

```
Before class method

Before test method

Test 1 method

After test method

After class method
```

TestNG - Data Providers

Un proveedor de datos es una matriz de datos que contiene filas de parámetros de prueba. La prueba toma cada fila de parámetros y se ejecuta, por lo que las pruebas se repiten n veces, siendo n la cantidad de filas.

Esto permite la prueba basada en datos, donde los datos de prueba no están codificados.

Una prueba basada en datos ejecuta verificaciones en diferentes entradas y verifica que las salidas sean correctas para cada conjunto de entradas.

TestNG - Data Providers

```
@DataProvider(name = "Authentication")

  public static Object[][] credentials() {

        return new Object[][] { { "testuser_1", "Test@123" }, { "testuser_1", "Test@123" }};

  }


@Test(dataProvider = "Authentication")

  public void test(String sUsername, String sPassword) {
      // test code
  }
```

TestNG - Ejercicio 2

Escriba una nueva clase con la misma estructura que el ejercicio 1, pero escriba la cadena que se imprimirá en el método de prueba que debe provenir de un Data Provider.

Resultado Esperado:

```
Before class method
Before test method
FIRST TEST STRING
After test method
Before test method
SECOND TEST STRING
After test method
After class method
```

Assert

Los Assert son puntos de control durante la ejecución de una prueba donde podemos verificar que se cumple alguna condición.

Si la condición falla, la prueba fallará y se detendrá, si pasa, la prueba continuará.

Ejemplos:

```
Assert.assertTrue(a == b, “Value A and B are the same”);

Assert.assertFalse(a == b, “Value A and B are different”);
```

TestNG - Exercise 3

Escriba una clase de prueba que contenga dos @Test, uno que no pasa y otro que pasa.