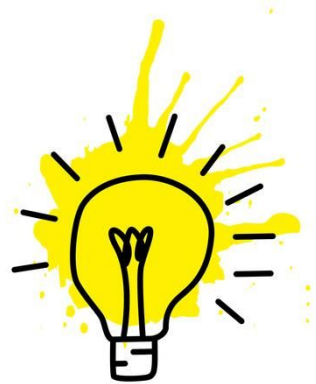


TestNG



TestNG

TestNG es un herramienta de automatización que se inspira en JUnit y sus anotaciones (@)

TestNG

Testng en el POM

```
<!-- https://mvnrepository.com/artifact/org.testng/testng -->  
<dependency>  
  <groupId>org.testng</groupId>  
  <artifactId>testng</artifactId>  
  <version>7.1.0</version>  
  <scope>test</scope>  
</dependency>
```

1. @BeforeSuite
 - a. @BeforeTest
 - i. @BeforeClass
 1. @BeforeMethod

@Test

2. @AfterMethod
3. @BeforeMethod

@Test

4. @AfterMethod
 - ii. @AfterClass
- b. @AfterTest
2. @AfterSuite

Ejercicios

@BeforeMethod

```
public void setBaseUrl(){
    GetProperties properties = new GetProperties();
    String chromeDriverUrl = properties.getString( propertyName: "CHROMEDRIVER_PATH");
    System.setProperty("webdriver.chrome.driver", chromeDriverUrl);
    driver = new ChromeDriver();
    driver.get(SPOTIFY);
}
```

@Test

```
public void verifyGoogleTitle(){
    String expectedTitle = "Música para todos - Spotify";
    String actualTitle = driver.getTitle();
    Assert.assertEquals(actualTitle, expectedTitle);
}
```

@AfterMethod

```
public void closeDriver(){
    driver.close();
}
```

@Test (Priority = 0)

Priority es un parámetro.

Requiere que se le asigne un valor numérico indicando el orden de ejecución de los tests.

```
@Test (priority = 0)
public void mostrarTitulo(){
    //Validar el título de la p
```

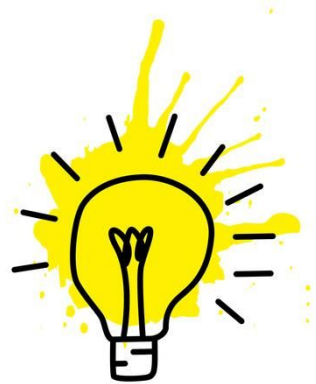


```
@Test (priority = 0)
public void mostrarTitulo(){
    //Validar el título de la
    System.out.println(driver.
```

```
@Test (priority = 2)
public void cantidadElementosP
    //Mostrar la cantidad de e
    List<WebElement> parrafos :
}
```

```
@Test (priority = 1)
public void mostrarLinks(){
    //Mostrar la cantidad de e
    List<WebElement> links = d
    for (WebElement e : links)
        System.out.println("---
    }
}
```


Testng.xml

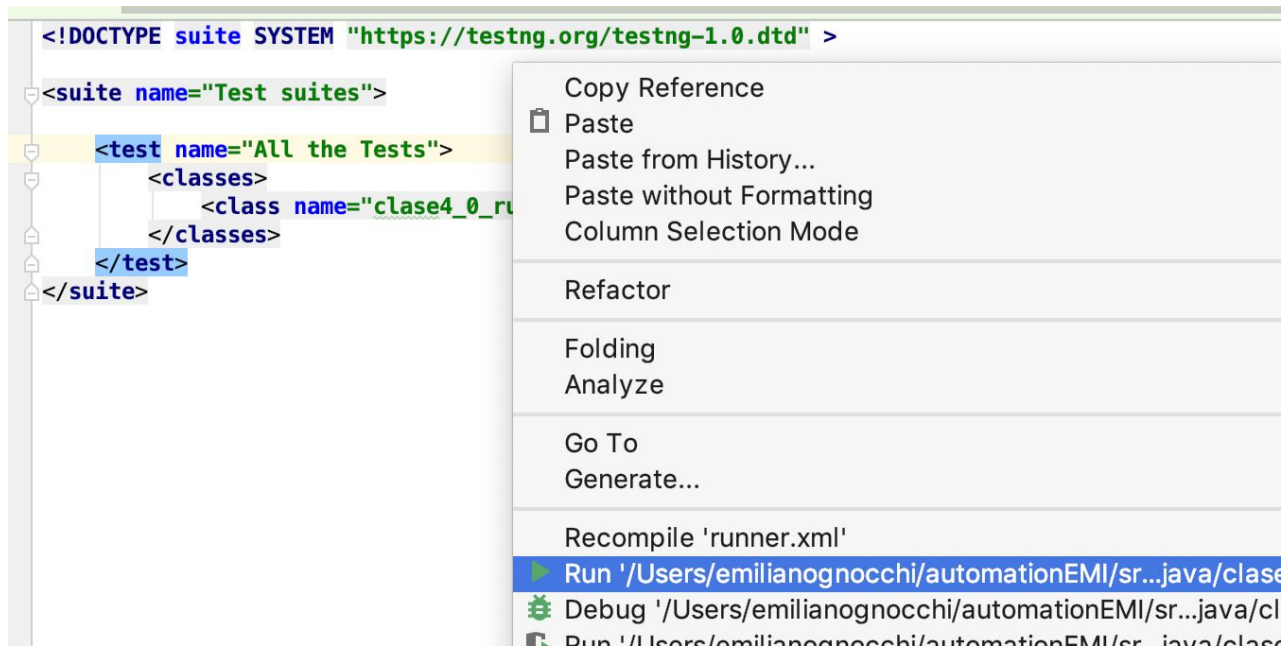


Testng.xml

Permite ejecutar uno o más tests y/o clases al mismo tiempo

```
<suite name="Clase 12">  
  <test name="Ejercicios Testng">  
    <classes>  
      <class name="practico12.testngTests"></class>  
    </classes>  
  </test>  
</suite>
```

Como se ejecuta el testng.xml?



@Parameters



@Parameters

Esta funcionalidad permite enviar parámetros a los tests .

Definir parámetros

Se define un parámetro llamado tagName y su valor es "a"

```
<parameter name="tagName" value="a"></parameter>
```

```
<parameter name="username" value="testing@qa.com"></parameter>
```

```
<parameter name="password" value="holamundo123"></parameter>
```

Esta variable puede ser vista desde la clase de test

Recibir parámetros

```
@Test
@Parameters({"specificTag"})
public void spotifyRegistrationForm(@Optional("h1") String tagName){
    driver.get(URL);
}
```

@optional

En caso de correr el tests directamente sin ir al runner, el valor que se encuentre en el parámetro @optional será el manejado por el método!

```
@Test
@Parameters({"specificTag"})
public void spotifyRegistrationForm(@Optional("h1") String tagName){
    driver.get(URL);
}
```


Ejemplo

```
@Test
@Parameters({"specificTag"})
public void spotifyRegistrationForm(@Optional("h1") String tagName){
    driver.get(URL);

    List<WebElement> elements = driver.findElements(By.tagName(tagName));

    if (tagName.equalsIgnoreCase( anotherString: "h1")){
        System.out.println("Se mostraran los h1 ");
    } else if (tagName.equalsIgnoreCase( anotherString: "h2")){
        System.out.println("Se mostraran los h2 ");
    } else if (tagName.equalsIgnoreCase( anotherString: "h3")){
        System.out.println("Se mostraran los h3 ");
    }
}
```

@Test(groups)

Se pueden agrupar los tests. De esta forma, podemos elegir desde el runner.xml cuales deseamos correr o cuales no.

```
@Test(groups = { "successTests", "failTests" })
```

```
@Test(groups = { "failTests" })  
public void failTest1()  
{  
}
```

```
@Test(groups = { "successTests" })  
public void successTest1()  
{  
}
```

```
<suite name="Test suites" verbose="1" >
  <parameter name="specificTag" value="h2"/>
  <test name="All the Tests">
    <groups>
      <run>
        <include name = "successTests" />
        <include name = "failTests" />
      </run>
    </groups>
  </test>
</suite>
```