Técnicas de Documentación y Validación 2022

## Trabajo evaluador Parte 3

# Automatización Web

Consigna: Describir sintéticamente los pasos a seguir para automatizar el testing de una página web con el driver de Selenium desde el momento en que se descarga

A continuación, se detallan los pasos a seguir para tener un ambiente en donde se pueda configuración acciones para aplicar un test de automatización web.

* Primero se descarga la versión del IDE que vamos a usar, en este caso eclipse desde el siguiente [link](https://www.eclipse.org/downloads/) .
* Luego la versión de java que vamos a usar enlace [link](https://www.oracle.com/ar/java/technologies/downloads/).
* Se procede a abrir el Eclipse instalado y vamos a crear un proyecto maven. Dentro del archivo pom.xml vamos a agregar las dependencia de selenium y testng descargadas de [maven\_repository](https://mvnrepository.com/).

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

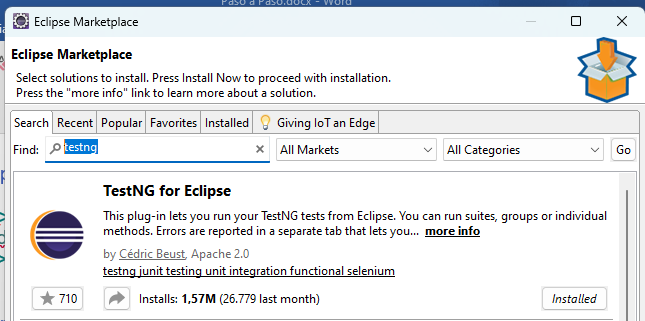
<artifactId>testng</artifactId>

<version>6.14.2</version>

<scope>test</scope>

</dependency>

* Luego vamos en la opción “Help”, “Eclipse Marketplace” e instalaremos el plug-in de testng.

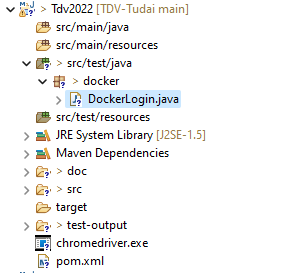


* Finalmente, como las pruebas serán realizadas sobre el navegador Chrome se necesita descargar el driver. Se buscará la versión actual de nuestro navegador y descargar [link](https://chromedriver.chromium.org/downloads) .

Siguiendo los pasos anteriores obtendremos un ambiente para realizar testing automatizados.

Ahora vamos a crear una clase para realizar un primer test.

Se tomará de ejemplo la página de [Docker](https://www.docker.com/). En dicha clase se abrirá la página de Docker validando el título de dicha página y luego cerrando el browser.



Quedando la clase:

**public** **class** DockerLogin {

**private** WebDriver driver;

@BeforeClass

**public** **void** launchBrowser() {

System.*setProperty*("webdriver.gecko.driver", "chromedriver.exe");

**this**.driver = **new** ChromeDriver();

}

@Test(priority = 1)

**public** **void** loadPage() {

driver.navigate().to("https://www.docker.com");

System.***out***.println(driver.getTitle());

*assertEquals*(driver.getTitle(), "Docker: Accelerated, Containerized Application Development");

}

@Test(priority = 2)

**public** **void** login() {

// **TODO**:

}

@Test(priority = 3)

**public** **void** finishTest() {

**if** (driver != **null**) {

driver.close();

driver = **null**;

}

System.***out***.println("Termino satisfactoriamente");

Assert.*assertNull*(driver);

}

}

Una vez creada la clase con los métodos procedemos a correr la aplicación para verificar la prueba

