Introducción a Selenium

Qué es el Testing Automatizado?

El Testing Automatizado es el proceso de automatizar los test manuales para probar una aplicación.

Involucra la creación de scripts que pueden ser ejecutados de forma repetitiva y no requiere ninguna intervención manual

Beneficios de Automatizar el testing

1. Ejecución de tests de forma repetitiva
2. Permite la ejecución en paralelo
3. Ejecución automática de los tests
4. Evita “errores humanos” a la hora de testear
5. Permite detectar errores

Por qué Selenium?

- Es gratis y open-source
- Tiene una gran cantidad de foros y comunidades activas
- Compatibilidad con muchos browsers
- Compatibilidad con Sistemas Operativos (Windows, Mac, Linux, etc)
- Soporta múltiples lenguajes de programación (Java, Ruby, C#, Python, etc)

Limitaciones de Selenium

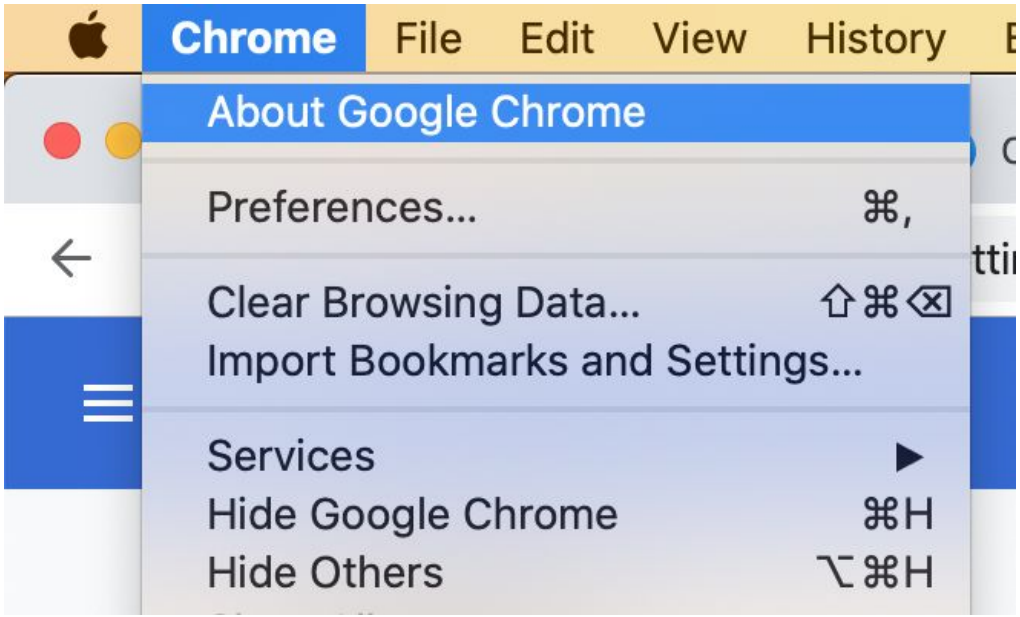
- Solo soporta testing de sólo aplicaciones web
- Las aplicaciones móviles no pueden ser automatizadas
- Los captcha no pueden ser automatizados
- No hay un “soporte” activo por parte de los creadores
- El usuario debe saber programar de antemano

Primeros pasos con Selenium

Descargar los drivers

https://chromedriver.chromium.org/downloads

Chrome version



Settings

About Chrome

Google Chrome

Google Chrome may not be able to keep itself updated

Learn more

Version 76.0.3809.132 (Official Build) (64-bit)

Get help with Chrome

Report an issue

Your browser is managed by joinmosaic.com

ChromeDriver - WebDriver for Chrome

Downloads

Current Releases

- If you are using Chrome version 84, please download ChromeDriver 84.0.4147.30
- If you are using Chrome version 83, please download ChromeDriver 83.0.4103.39
- If you are using Chrome version 81, please download ChromeDriver 81.0.4044.138
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary page.

For more information on selecting the right version of ChromeDriver, please see the Version Selection page.

ChromeDriver 84.0.4147.30

Supports Chrome version 84

- Resolved issue 3420: after switching to the print window, the chromedriver stops responding
- Resolved issue 3421: Driver returns Cyrillic text without styles
- Resolved issue 3422: GetElementText breaks with prototype 1.6.0.3
- Resolved issue 3434: Cannot get 'assert' messages from the 'browser' logs

ChromeDriver 81.0.4044.138

Supports Chrome version 81

- Updated Chromedriver to work correctly with prototype.js.

For more details, please see the release notes.

ChromeDriver 81.0.4044.69

Supports Chrome version 81

- Fixed: Chromedriver crashes on getPageSource on some sites.
- Fixed: ChromeDriver crashes on certain element or click commands.

For more details, please see the release notes.

ChromeDriver 81.0.4044.20

Supports Chrome version 81

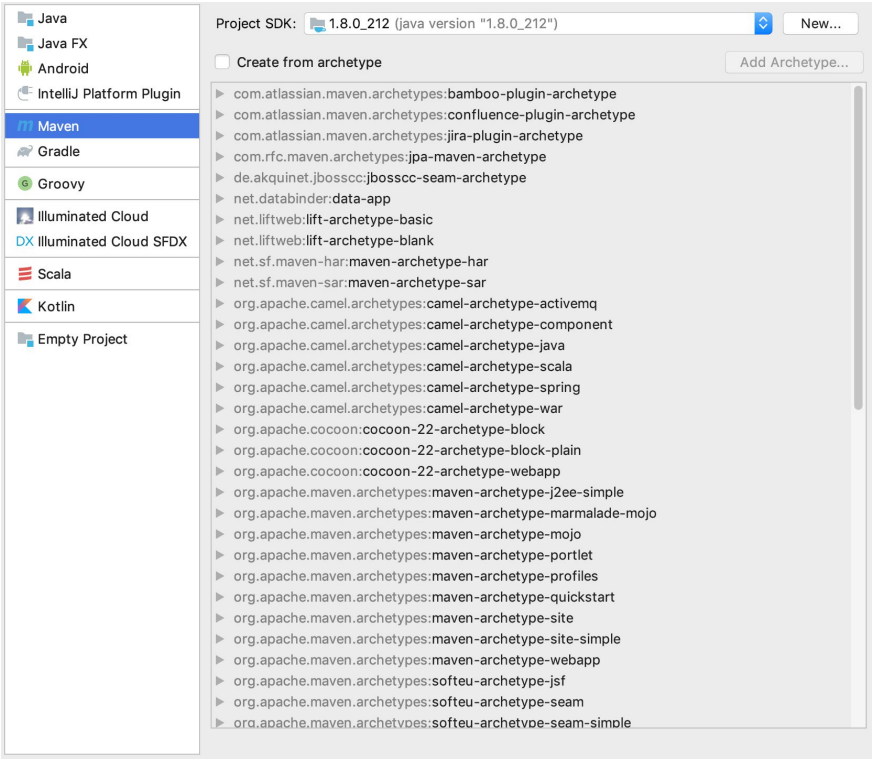
- Fixed error when attempting to get location of html element.
- Get and Add Cookie commands are now frame specific.

Elegimos el driver para nuestro sistema operativo

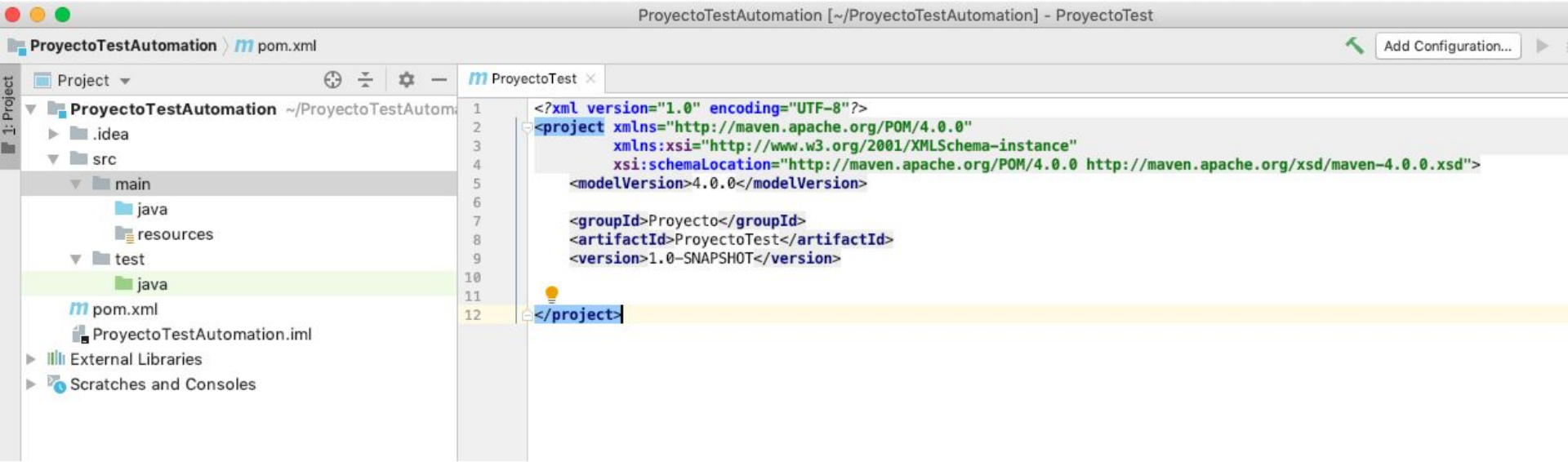
Index of /81.0.4044.138/

Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2020-05-05 20:33:58	4.74MB	6581e09a8ce12da2239ac21b2a1cd20b
chromedriver_mac64.zip	2020-05-05 20:34:00	6.70MB	4b2ace862187dc9e53d29c9f12710731
chromedriver_win32.zip	2020-05-05 20:34:01	4.20MB	d19aef5daf9dbaeac152d27066285a7b
notes.txt	2020-05-05 20:34:05	0.00MB	a40835254450bd55cb1c9a1895939357

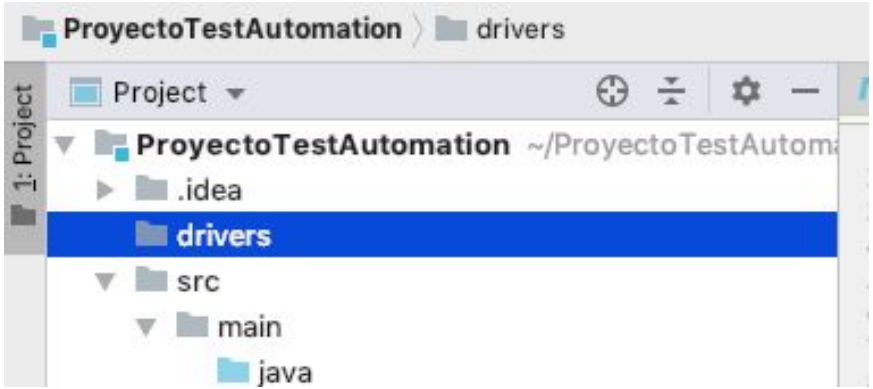
Crear proyecto Mavens en Intellij



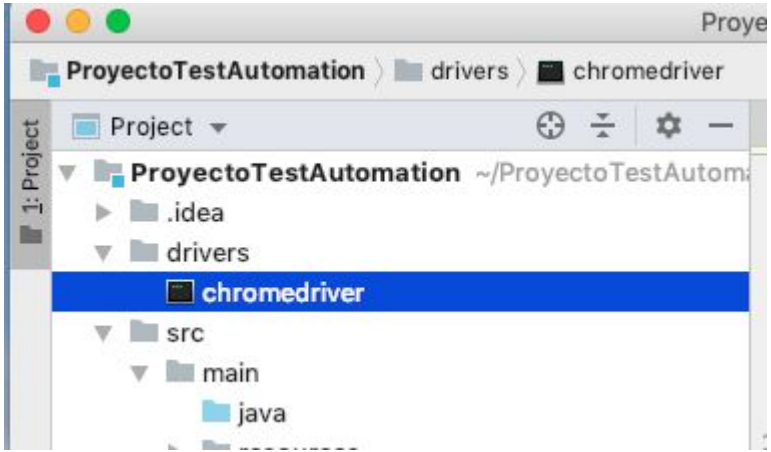
Estructura del proyecto



Crear una carpeta llamada drivers



Colocamos el driver que descargamos en la carpeta



POM.xml

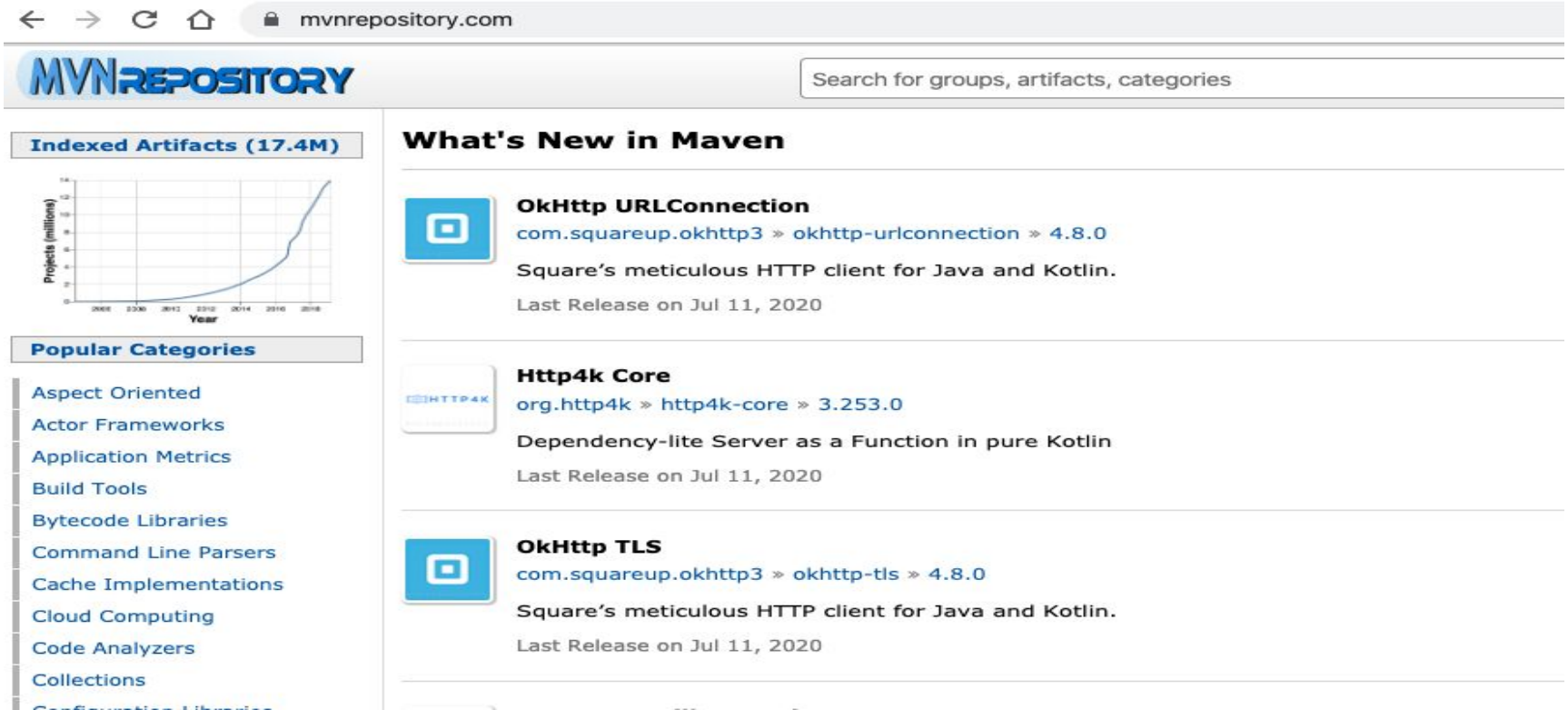
El archivo POM contiene información fundamental sobre el proyecto y su configuración en los proyectos Mavens.

Contiene dependencias, variables, etc.

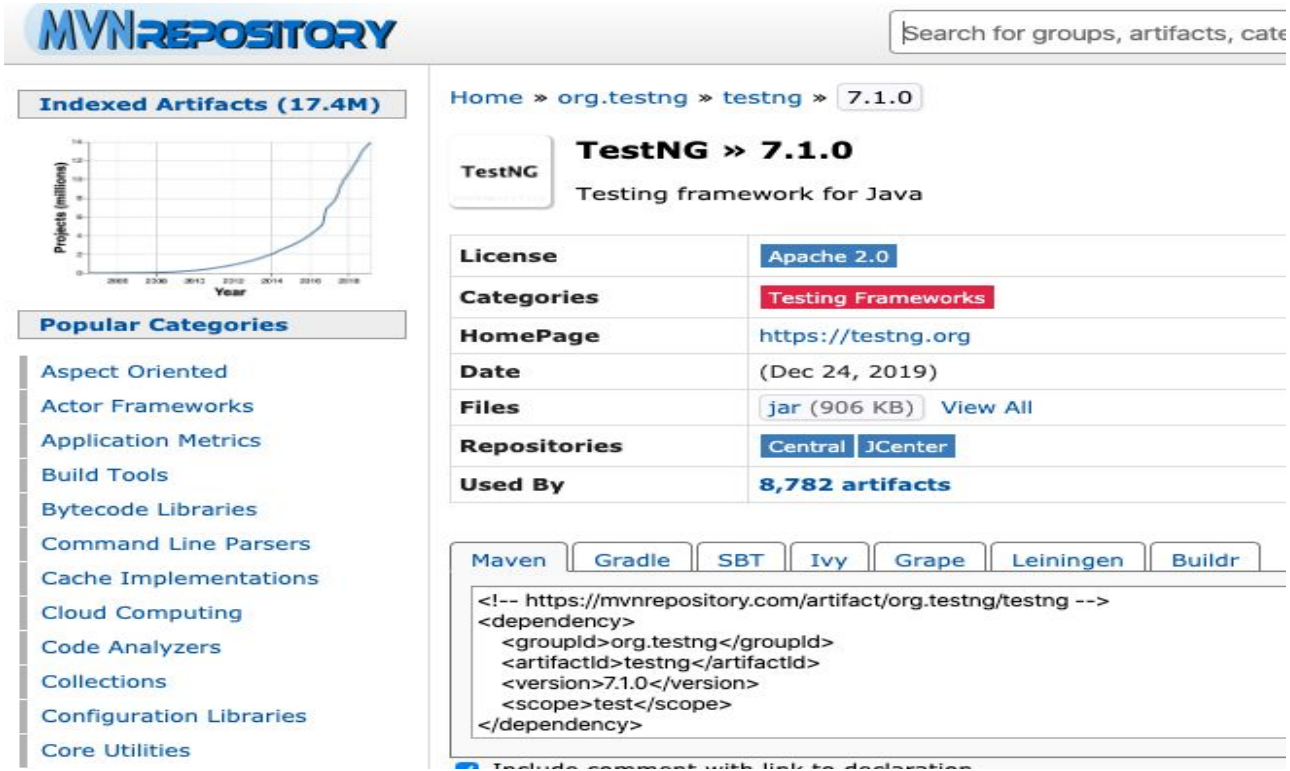
POM

```
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mav
    <modelVersion>4.0.0</modelVersion>

    <groupId>Proyecto</groupId>
    <artifactId>ProyectoTest</artifactId>
    <version>1.0-SNAPSHOT</version>


    <properties>
        <testng.version>7.1.0</testng.version>
        <selenium.version>3.141.59</selenium.version>
        <aerogearotp.version>1.0.0</aerogearotp.version>
        <rest-assured.version>4.3.0</rest-assured.version>
        <gson.version>2.8.6</gson.version>
    </properties>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>${selenium.version}</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.testng/testng -->
        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>${testng.version}</version>
            <scope>test</scope>
        </dependency>

        <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
        <dependency>
            <groupId>io.rest-assured</groupId>
            <artifactId>rest-assured</artifactId>
            <version>${rest-assured.version}</version>
            <scope>test</scope>
        </dependency>
```

Ingresamos a Maven Repository



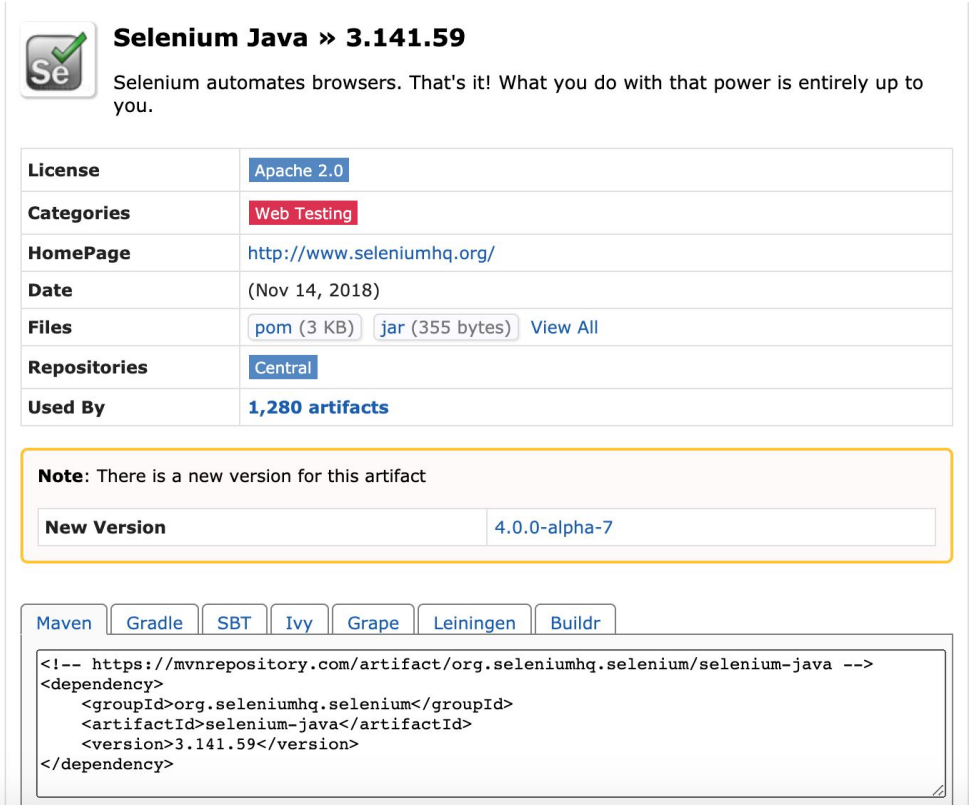
Copiamos la dependencia a Testng



En el pom.xml….

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://mav
    <modelVersion>4.0.0</modelVersion>

    <groupId>proyectoViernes</groupId>
    <artifactId>proyectoViernes</artifactId>
    <version>1.0-SNAPSHOT</version>

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

    </dependencies>

</project>
```

Copiamos la dependencia a Selenium Java



Selenium Java » 3.141.59

Selenium automates browsers. That's it! What you do with that power is entirely up to you.

License	Apache 2.0
Categories	Web Testing
HomePage	http://www.seleniumhq.org/
Date	(Nov 14, 2018)
Files	pom (3 KB) jar (355 bytes) View All
Repositories	Central
Used By	1,280 artifacts

Note: There is a new version for this artifact

New Version	4.0.0-alpha-7

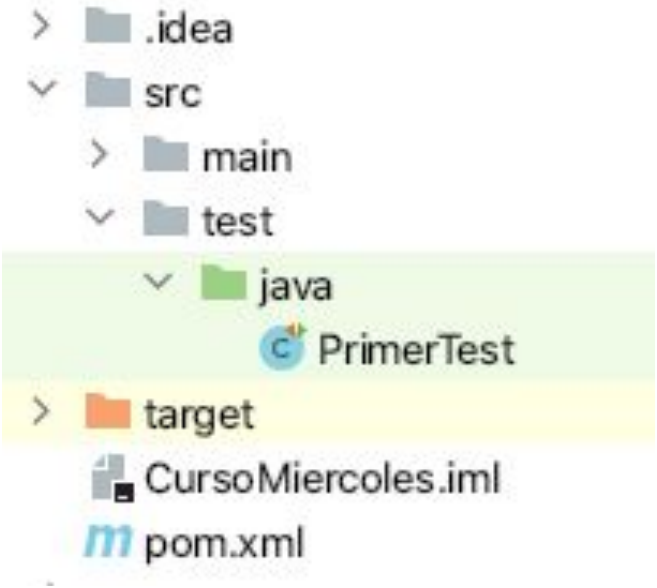
Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- 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>
```

En el pom.xml….

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

Creamos una clase para realizar los tests



Primer test en Windows

```
WebDriver driver;

@Test
public void testInWindows(String URL){
    System.setProperty("webdriver.chrome.driver", "drivers/chromedriver.exe");
    driver = new ChromeDriver();
    driver.get("https://www.facebook.com");
}
```

Primer test en Mac

```
WebDriver driver;

@Test
public void testInMac(String URL){
    System.setProperty("webdriver.chrome.driver", "drivers/chromedriver");
    driver = new ChromeDriver();
    driver.get("https://www.facebook.com");
}
```

Cómo se crean instancias de los Browsers?

WebDriver driver = new FirefoxDriver();

WebDriver driver = new ChromeDriver();

WebDriver driver = new InternetExplorerDriver();

Acceder a un sitio web

```
WebDriver driver = new ChromeDriver();


String baseURL = "https://www.bbc.com/mundo";

driver.get(baseURL);
```

Qué tipos de Drivers están disponibles?

- FirefoxDriver
- InternetExplorerDriver
- ChromeDriver
- SafariDriver
- OperaDriver

Métodos del Driver

driver.navigate().refresh()

driver.navigate().back();

driver.navigate().forward();

driver.manage().window().maximize();

Métodos de Webdrivers

```
int altura = driver.manage().window().getSize().getHeight();

int ancho = driver.manage().window().getSize().getWidth();
```

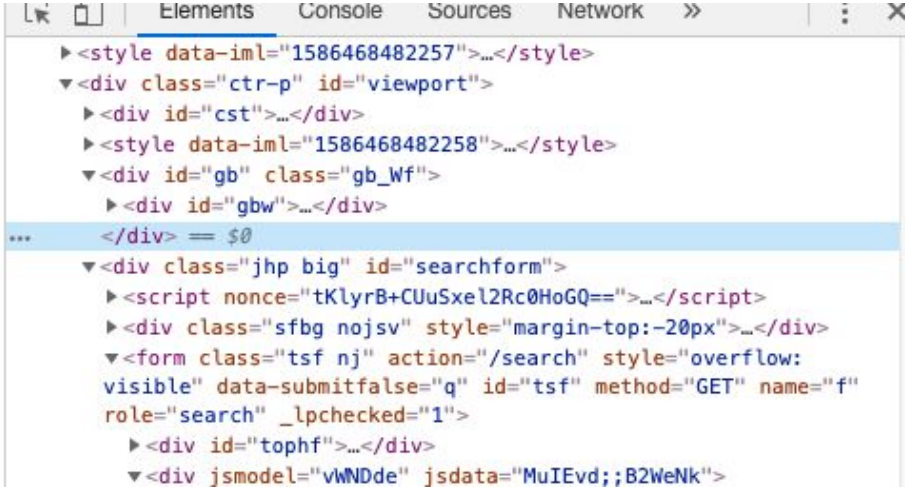
Como se cierra el explorador?

Cuando corremos nuestros tests, el browser queda abierto...

driver.close(); → cierra el browser actual

driver.quit(); → cierra el TODOS los browser

Elementos de un HTML



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

```
<!DOCTYPE html>
<html>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

<p><b>Tip:</b> Use h1 to h6 elements only for headings. Do not use
them just to make text bold or big. Use other tags for that.</p>

</body>
</html>
```

Links en HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>Esto es un título</h1>

<a href="https://www.facebook.com">Facebook</a>

</body>
</html>
```

Esto es un título

Facebook

Encontrar elementos en un sitio web

driver.**findElement**(LOCALIZADOR)

Busca el primer elemento que contiene la propiedad y lo retorna

Retorno: **WebElement**

driver.**findElements**(LOCALIZADOR)

Busca TODOS los elementos que contienen la propiedad y los retorna en forma de lista

Retorno: List<**WebElement**>

Métodos de WebElement

```
WebElement buttonLogin = driver.findElement(By.tagName("button"));

buttonLogin.click();


driver.findElement(By.id("password")).sendKeys("holamundo");
```

Métodos de WebElement

System.out.println(driver.**findElement**(By.tagName("h3")).**getText()**);

boolean isNameBox= driver.**findElement**(By.id(“name”)).**isDisplayed()**;

Métodos de WebElement

```
boolean isTipoProductoSelected =
driver.findElement(By.id("tipoProducto")).isSelected();

boolean isPromoEnabled =
driver.findElement(By.id("PremiumPromo")).isEnabled();
```

Tipos de Localizadores

- By.id()
- By.name()
- By.tagName()
- By.className()
- By.linkText()
- By.partialLinkText()
- By.xpath()
- By.cssSelector()