|  |  |
| --- | --- |
|  | Automatización de pruebas de regresión en aplicaciones web usando Selenium |
| **Manual de instalación y configuración de componentes** | |

Versión: 0.2

Mérida, Yucatán a 11 de mayo de 2018**Tabla de Contenido**

[1. Control de Documentación](#_Toc464056187) 2

[2. Introducción AL DOCUMENTO](#_Toc464056188) 3

3. DESCARGAS E INSTALACIÓN……………..………………………………………………………..4

*3.1 JAVA JDK……………………………………………………………………………..………….…4*

*3.2 INTELLIJ IDE…..................................................................................................................4*

*3.3 MAVEN……………………………………………………………………………….………..……4*

*3.4 GOOGLE CHROME………………………………………………………………………..…….4*

*3.5 CHROMEDRIVER……………………………………………………………………….………5*

*3.6 DEPENDENCIAS MAVEN………………………………………………………………...…5*

4. CONFIGURACIÓN DE UN PROYECTO DE AUTOMATIZACIÓN……………...………...7

5. REFERENCIAS…………………………………………………………………………………………..10

# Control de Documentación

**Control de Configuración**

|  |  |
| --- | --- |
| Título del documento: | Manual de instalación y configuración de componentes |
| Nombre del proyecto: | Automatización de pruebas de regresión en aplicaciones web usando Selenium |
| Autor(es): | Rosaura Irizell Garcilazo Cuevas |
| Fecha: | 17/04/2018 |

**Histórico de versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| 0.1 | 17/04/2018 | B | Rosaura Garcilazo | Automation\_SetUp |
| 0.2 | 11/05/2018 | B | Rosaura Garcilazo | Automation\_SetUp |

Estado: (B)orrador, (R)evisión, (A)probado

**Histórico de cambios**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Cambios** |
| 0.1 | 17/04/2018 | Ninguna, primera versión borrador |
| 0.2 | 11/05/2018 | Correcciones de redacción |
|  |  |  |
|  |  |  |
|  |  |  |

# Introducción AL DOCUMENTO

Este documento tiene como objetivo detallar la instalación y el uso de los componentes necesarios para la automatización de pruebas de regresión en aplicaciones web y su futura integración con Jenkins.

1. **DESCARGA E INSTALACIÓN**
   1. **Java JDK**

Entorno de desarrollo para aplicaciones, applets y módulos usando Java como lenguaje de programación. Versión : 8 +

Link de descarga:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

* 1. **IntelliJ IDE**

Este IDE proporcionar el marco de trabajo sobre el cual se construirán las pruebas unitarias del proyecto. Versión: 3.2 +

Link de descarga:

<https://www.jetbrains.com/idea/>

* 1. **Maven**

La utilidad de este manejador de dependencias en el proyecto es su fácil integración con Jenkins CI y el control que ofrece sobre las pruebas unitarias mediante comandos. Versión 3 +

* 1. **Google Chrome**

Como se verá más adelante, Selenium funciona como software de pruebas automatizadas para distintos navegadores; por comodidad y practicidad, Chrome se usará como navegador para todas las pruebas

* 1. **chromedriver**

Para poder ejecutar la automatización de procesos web en Chrome es necesario contar con el controlador de navegador correspondiente, este debe situarse en el *PATH* del ordenador para poder usarse. Versión: 2.37 +

Link de descarga:

<https://sites.google.com/a/chromium.org/chromedriver/>

* 1. **Dependencias Maven**

A partir de aquí, las librerías y software necesario serán añadidos como dependencias de Maven al archivo pom.xml del proyecto como se describe en la sección 4 de este documento.

* + 1. **Selenium**

Esta es la librería que permitirá manipular los elementos web de la aplicación web que se automatizará. Versión: 3.11

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>3.11.0</version>

</dependency>

* + 1. **JUnit 5**

Marco de trabajo de desarrollo de pruebas unitarias.

El manejar el proyecto como un conjunto de pruebas unitarias sirve para mantener un control de la estabilidad de la aplicación y la modularización del proyecto de automatización. Versión: 5.0.1

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

<version>5.0.1</version>

<scope>test</scope>

</dependency>

Este marco de trabajo es complementado con Hamcrest. Versión: 1.3

<dependency>

<groupId>org.hamcrest</groupId>

<artifactId>hamcrest-core</artifactId>

<version>1.3</version>

<scope>test</scope>

</dependency>

* + 1. **Maven plugins**

Los siguientes plugins permiten ejecutar las pruebas con comandos Maven y generar los archivos xml con los resultados de cada prueba

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.19.1</version>

<configuration>

<testFailureIgnore>true</testFailureIgnore>

</configuration>

<dependencies>

<dependency>

<groupId>org.junit.platform</groupId>

<artifactId>junit-platform-surefire-provider</artifactId>

<version>1.1.0</version>

</dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-engine</artifactId>

<version>5.0.1</version>

</dependency>

</dependencies>

</plugin>

</plugins>

</build>

* + 1. **Adicional:**

El siguiente script debe colocarse al final del archivo pom.xml para indicar el formato de texto que procesarán los plugins antes mencionados

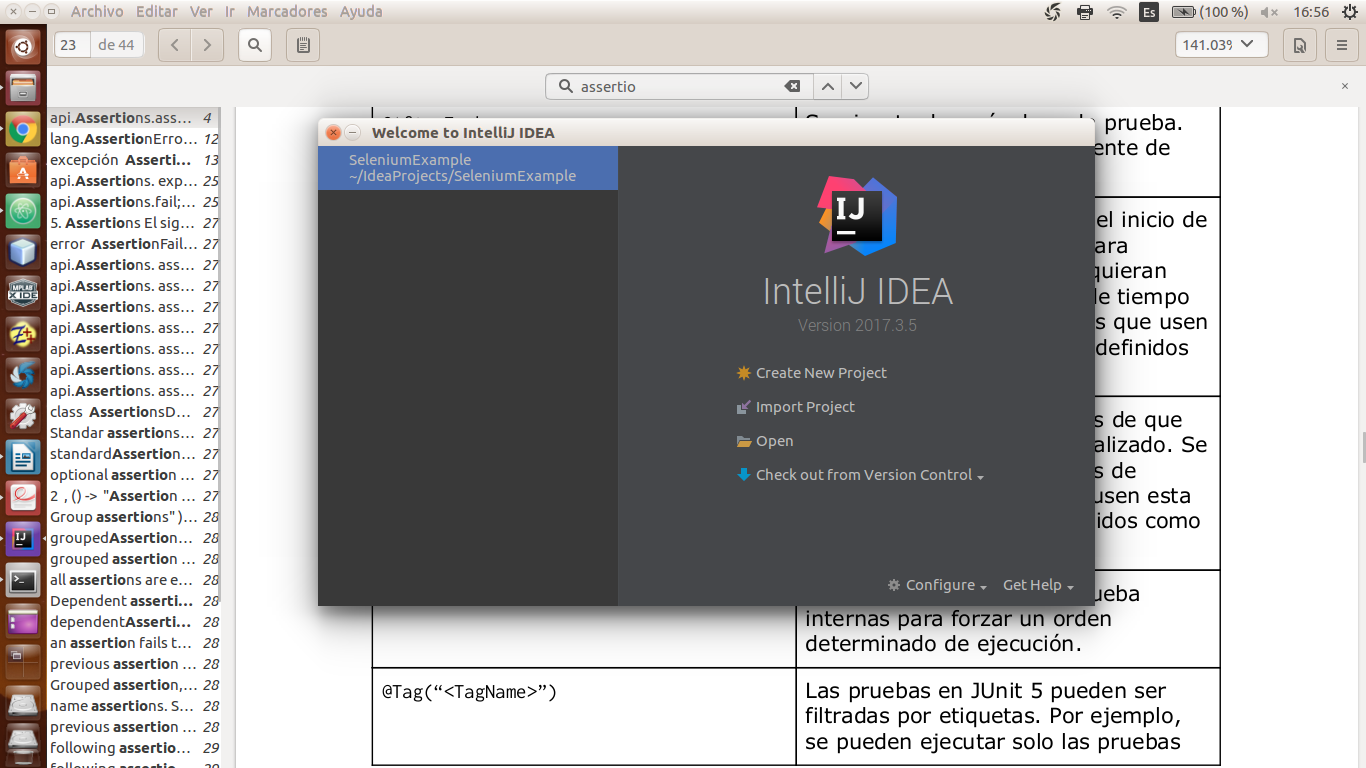
<properties>

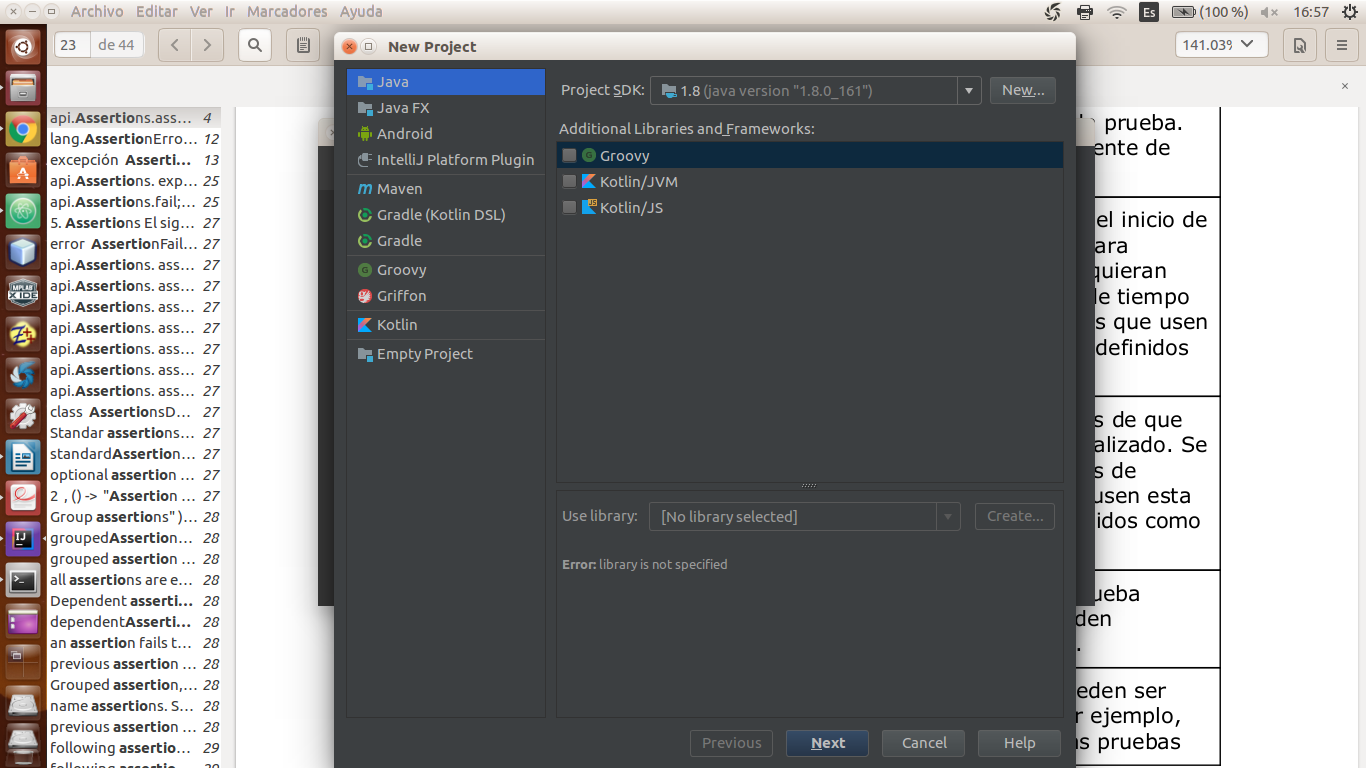
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

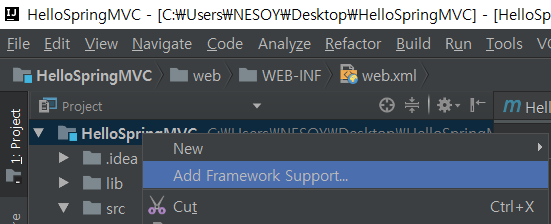
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding

</properties>

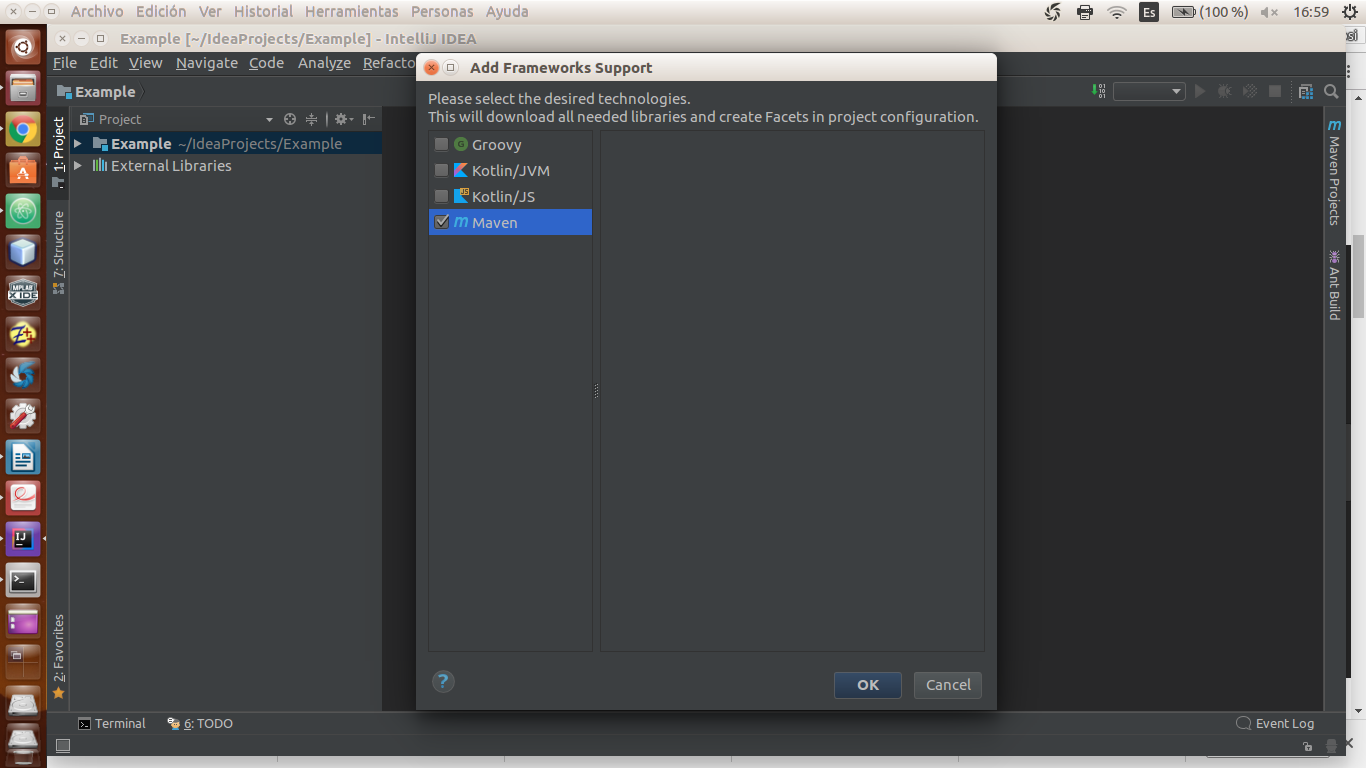
1. **CONFIGURACIÓN DE UN PROYECTO DE AUTOMATIZACIÓN**
   1. **Abrir intelliJ**



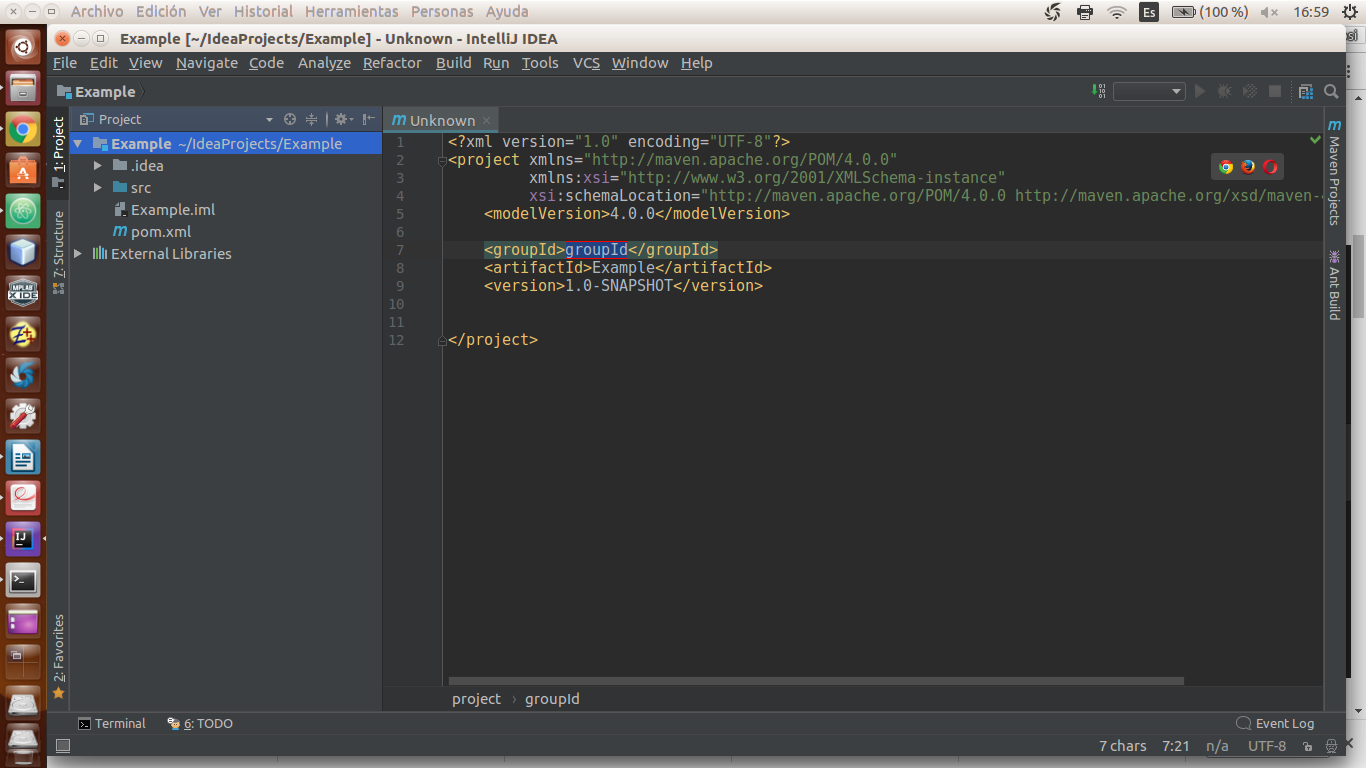
* 1. **Crear un proyecto Java** 
  2. **Dar click derecho sobre el nombre del proyecto y seleccionar “Add FrameWork Support”**



* 1. **Seleccionar Maven como manejador de dependencias.**



* 1. **Esto automáticamente creará el archivo pom.xml al que se le agregarán las dependencias necesarias (Sección 3)**



* 1. **Inicializar Maven**

Para poder usar los comandos de Maven y generar los reportes de resultados de pruebas es necesario inicializar Maven desde la consola de IntelliJ con el comando “mvn site”.

Esto creará los directorios target/ que recopilarán la información de los reportes de pruebas.

1. **Referencias:**

* *Guía de uso Selenium Br. Rafael Rodríguez Guzmán Revisión: Br. Roberto Esteban Gil Flores 15/Enero/2018*
* [*https://mvnrepository.com/*](https://mvnrepository.com/)