

# **Find Security Bugs**

## **Manual de Instalación y Configuración**

**Realizado por**

Grupo 3 – Corocotta

### **Integrantes**

Hamza Hamda

Iván Sánchez Calderón

Juan David Corrales Gil

Ricardo Armando Blanco López

Jaime Eduardo Baires Escalante

### **Asignatura**

Calidad y Auditoría

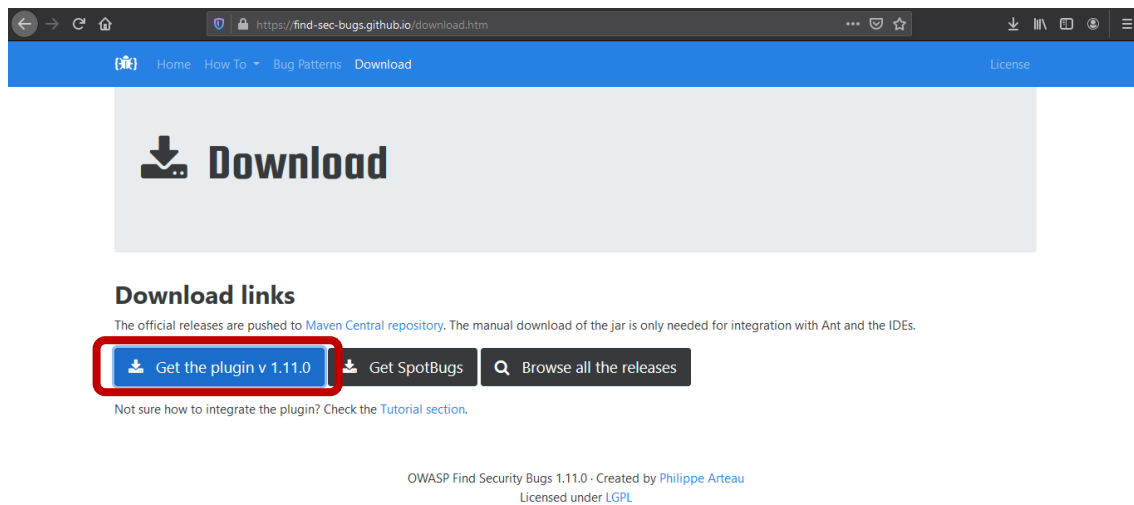
## Índice

1. Instalación En Eclipse .....	3
<b>1.1 Descarga de Plugin</b> .....	3
<b>1.2 Instalación en Eclipse</b> .....	4
2. Configuración en Eclipse .....	6
3. Find Security Bugs como plugin con Maven .....	8
<b>3.1 Crear un proyecto Maven</b> .....	8
4. Configuración de Find Security Bugs con Maven .....	10
5. SonarQube con Find Security Bugs .....	12
5.1 Software necesario .....	12
5.2 Configuración .....	12

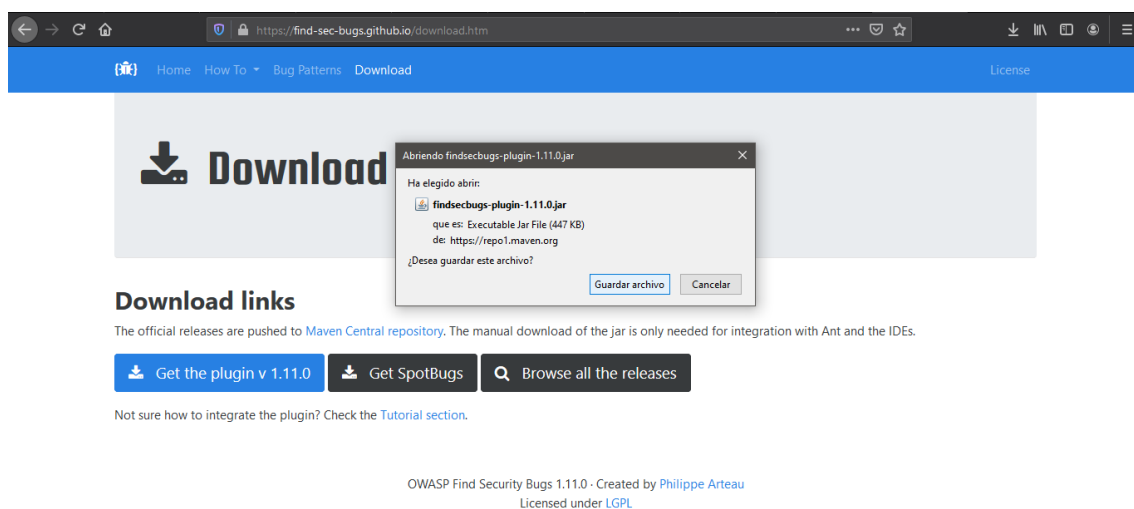
# 1. Instalación En Eclipse

## 1.1 Descarga de Plugin

Url: <https://find-sec-bugs.github.io/download.htm>

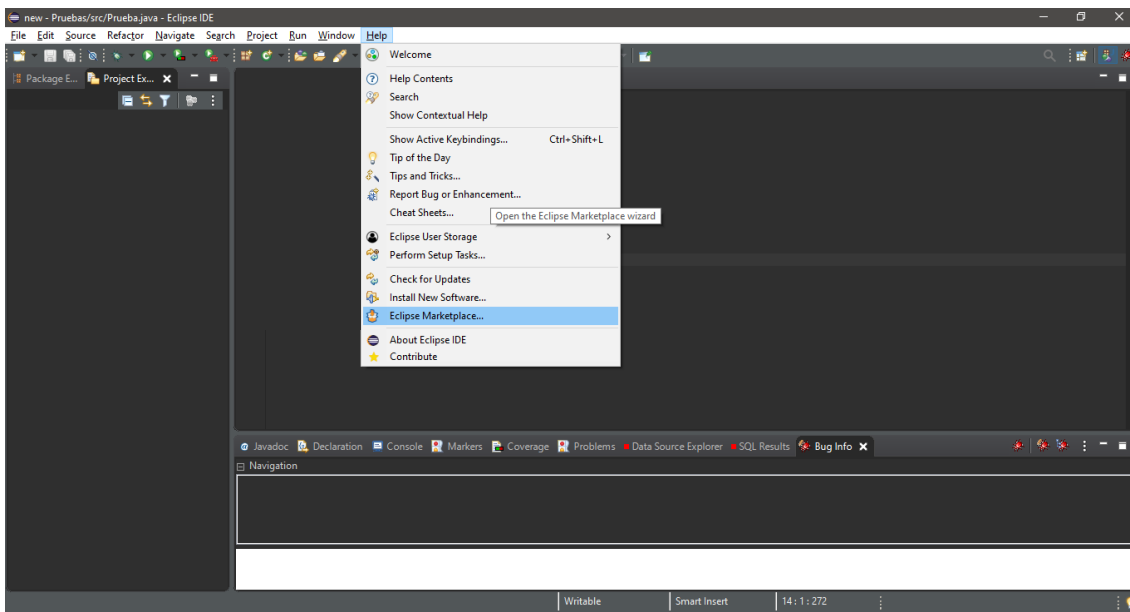


<https://search.maven.org/remotecontent?filepath=com/h3xstream/findsecbugs/findsecbugs-plugin/1.11.0/findsecbugs-plugin-1.11.0.jar>



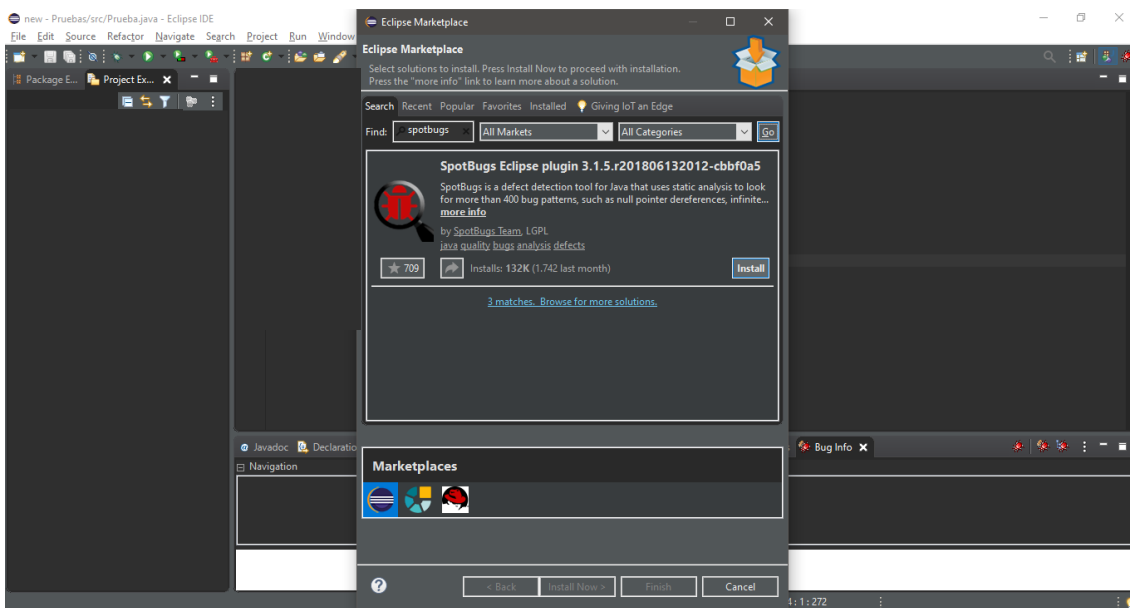
## 1.2 Instalación en Eclipse

Help -> Eclipse Marketplace

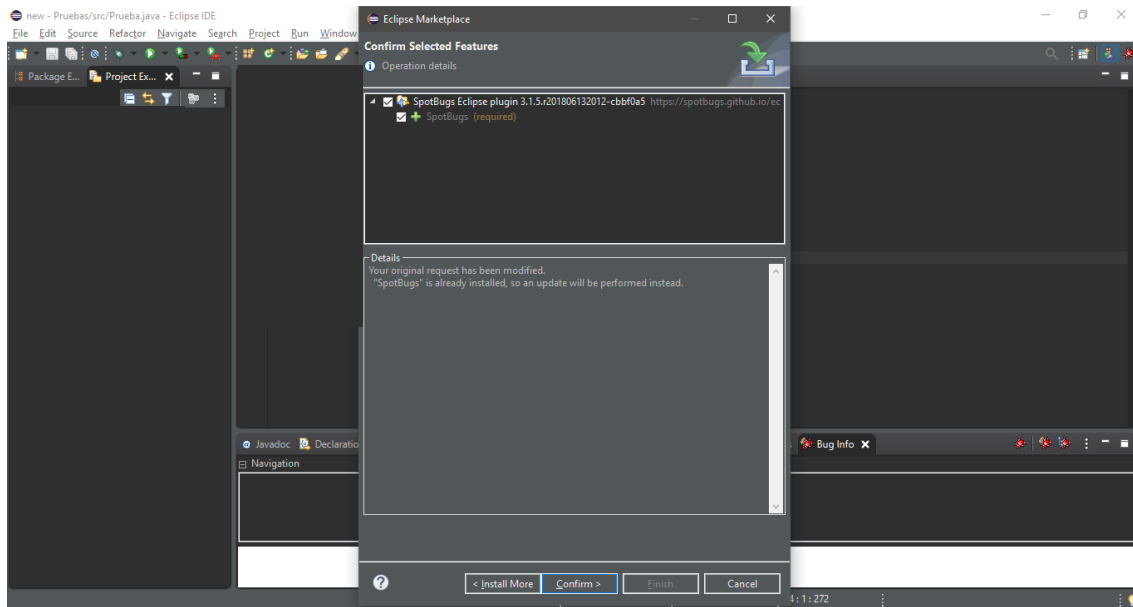


En el input “Find” escribir “spotbugs” y presionar el botón “Go”.

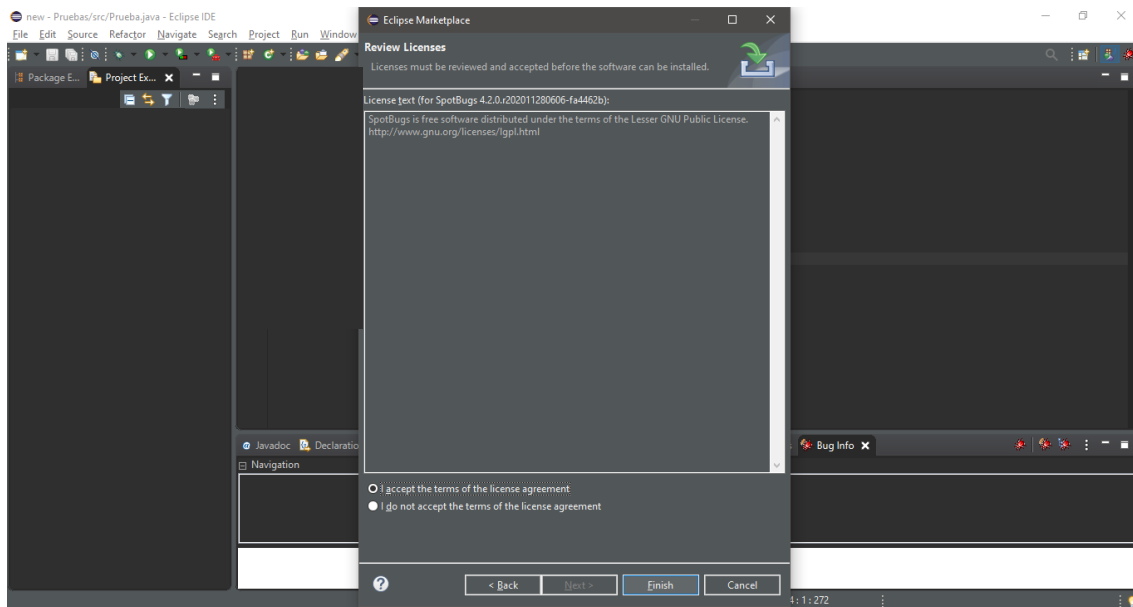
En este resultado, presionar “Install”.



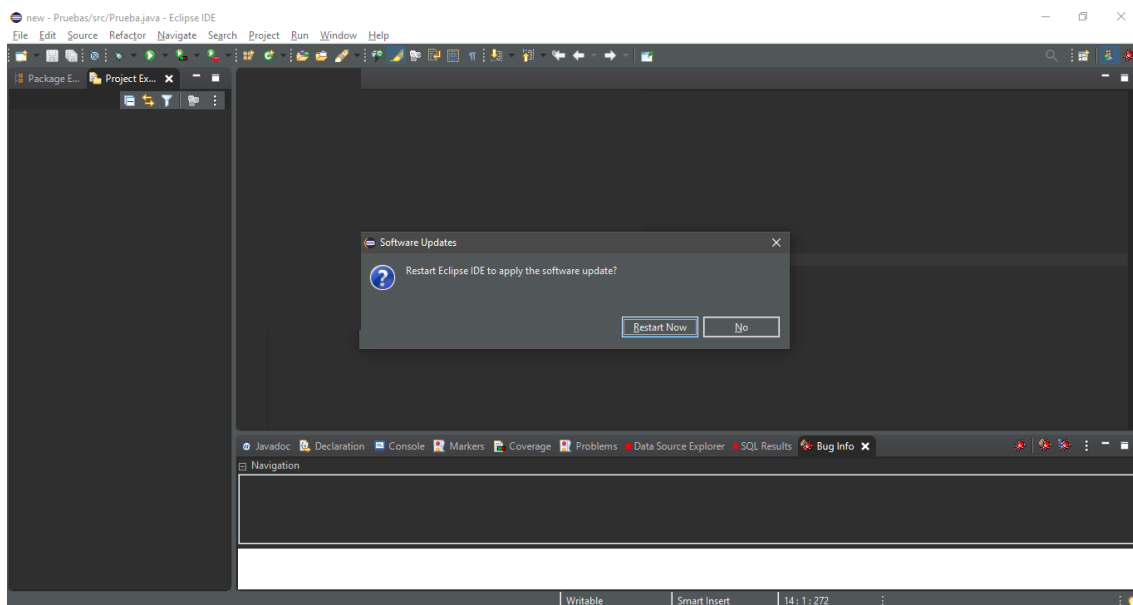
Seleccionar la herramienta, si no lo está. Presionar “Confirm”.



Aceptar los términos y hacer clic en “Finish”.

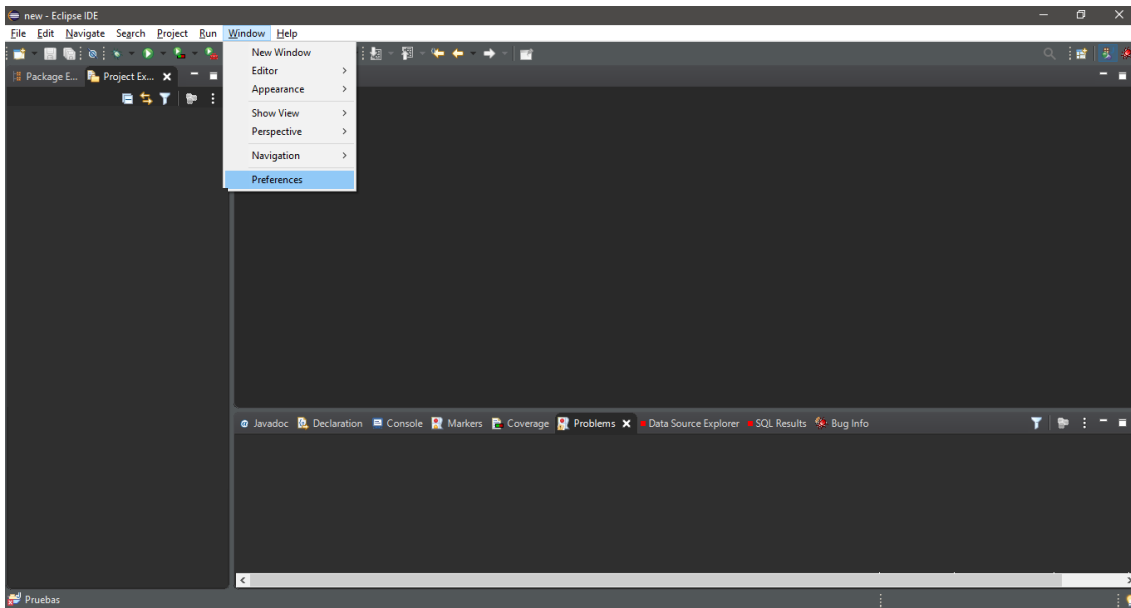


Reiniciar Eclipse.

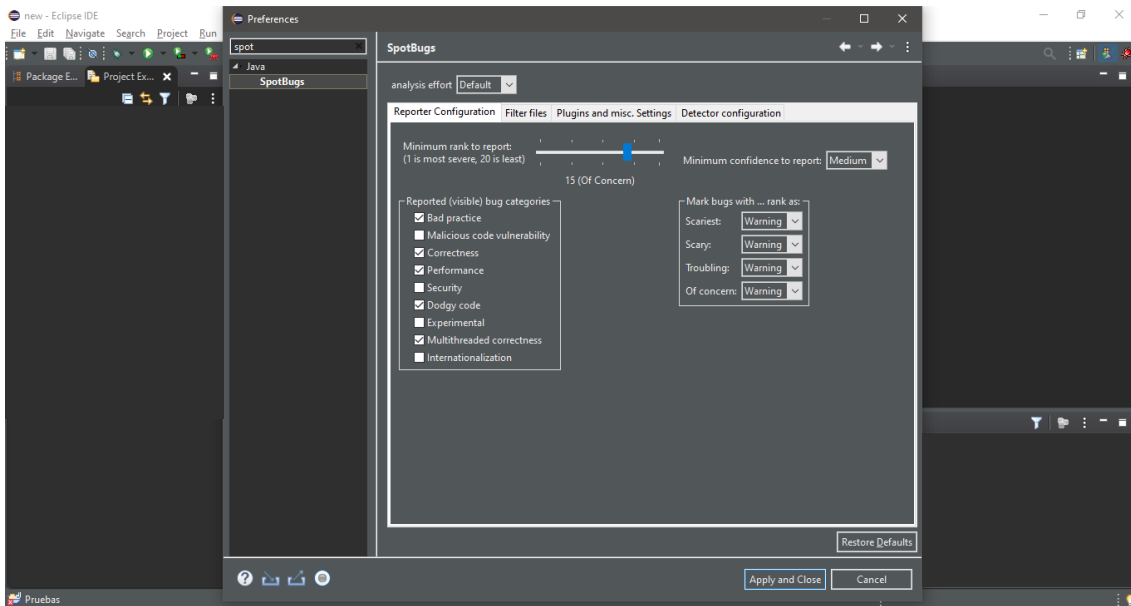


## 2. Configuración en Eclipse

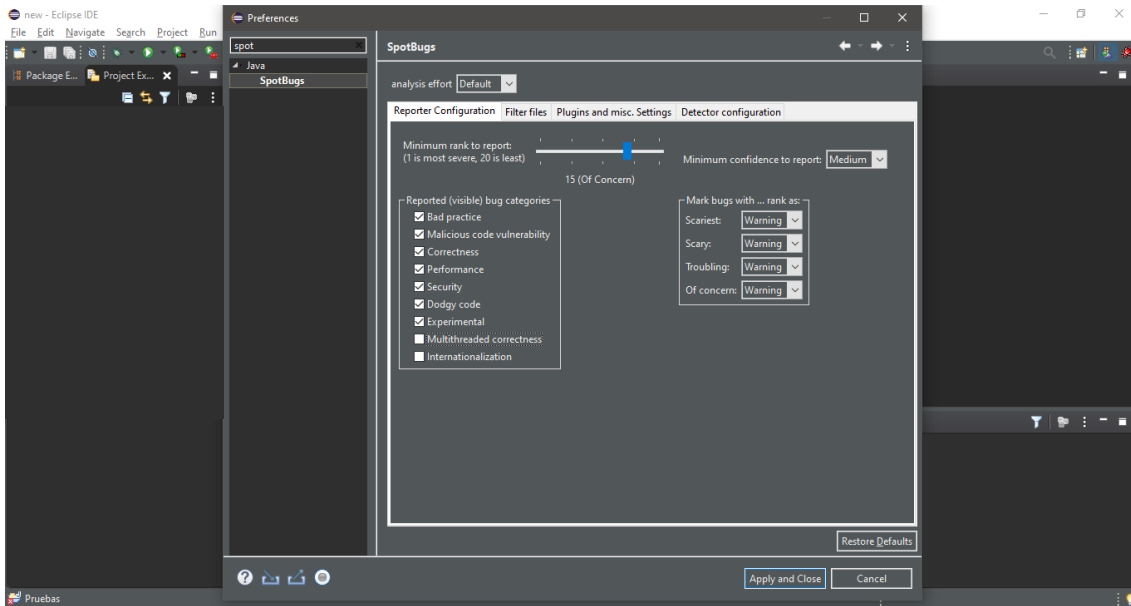
Window -> Preferences



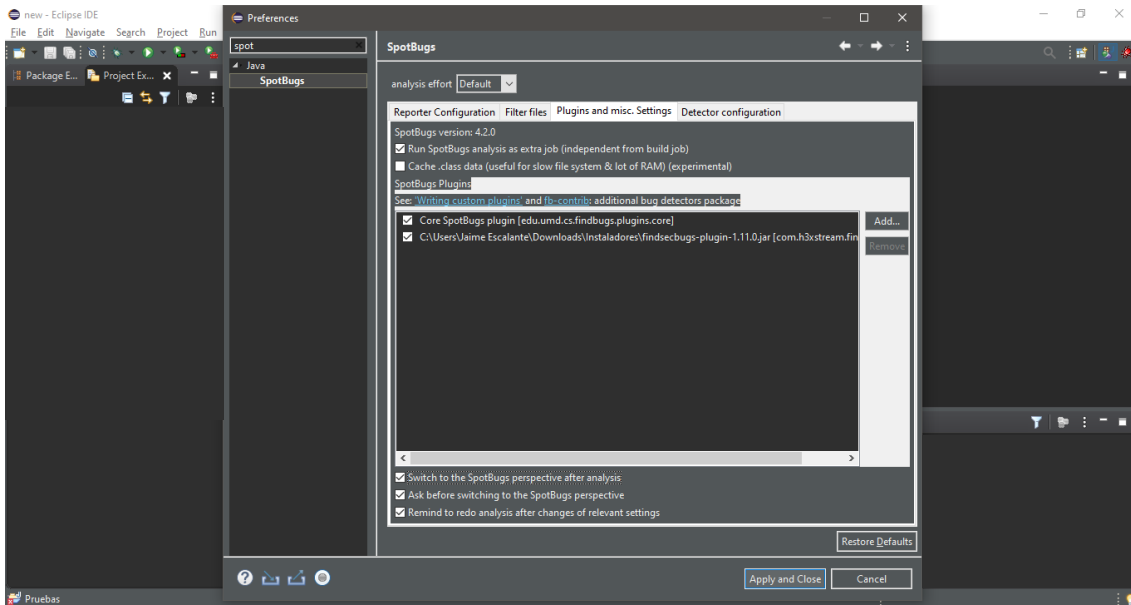
En la barra de búsqueda escribir “spotbugs” y entrar al apartado correspondiente.



En la pestaña que sale por defecto, en el apartado “Reported (visible) bug categories” se pueden seleccionar las clases de bugs a reportar. Para que muestre todo lo necesario, dejar seleccionado todas las casillas excepto las últimas dos.



En la pestaña “Plugins and misc. Settings” añadir el jar descargado previamente.

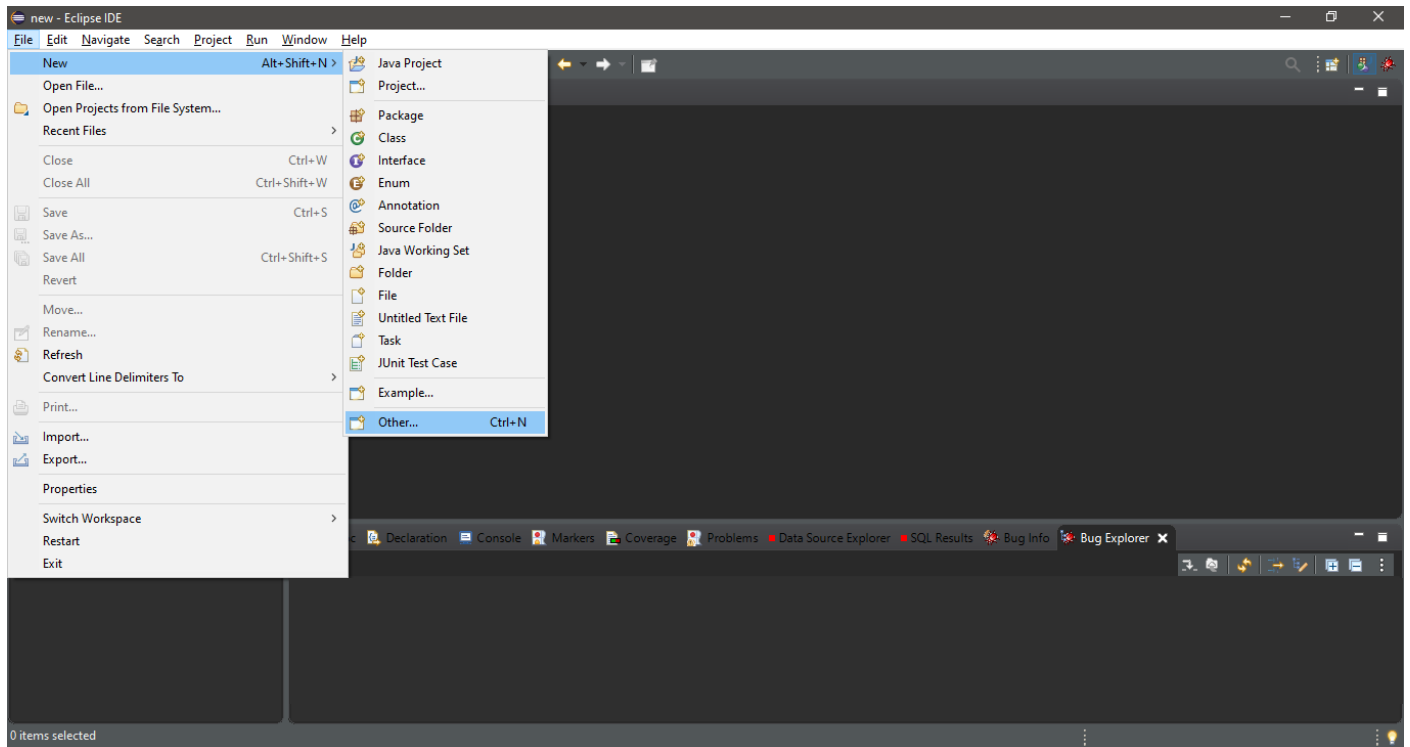


Reiniciar manualmente Eclipse.

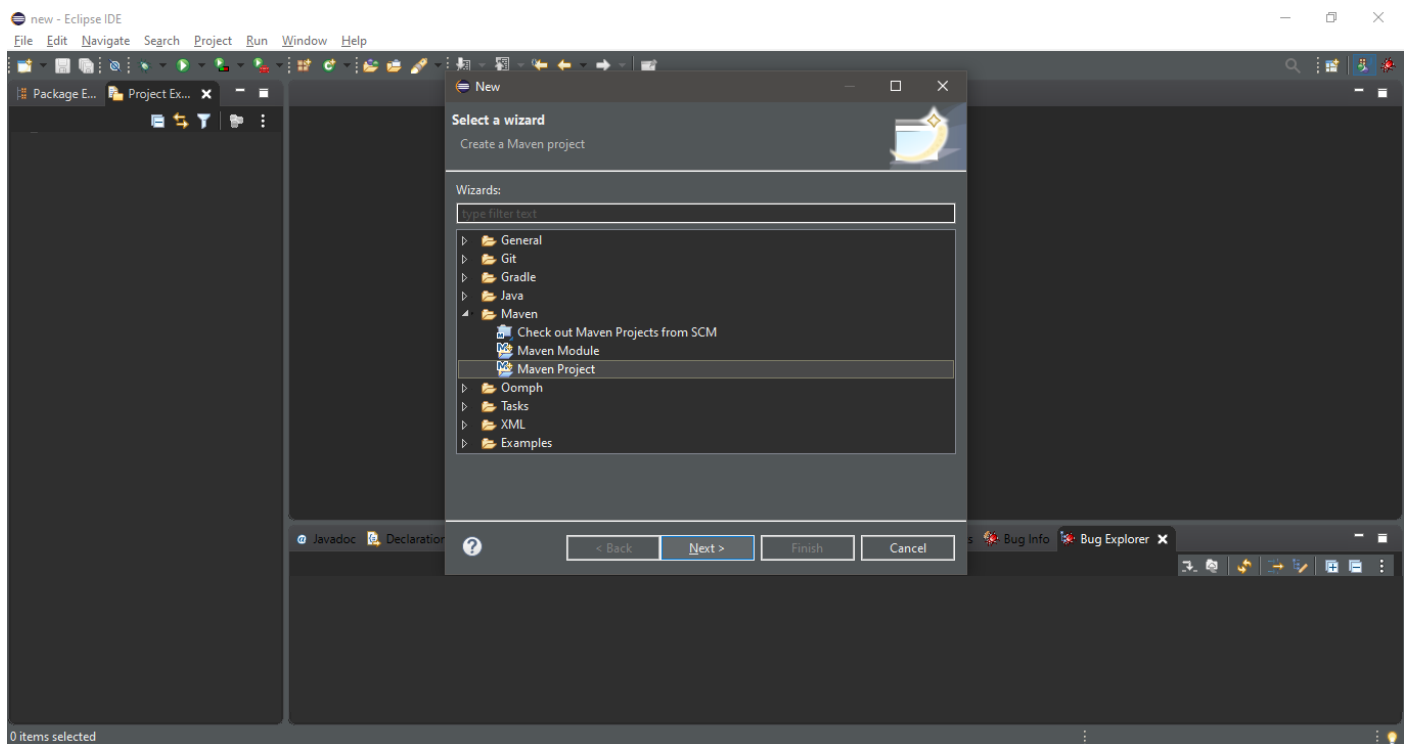
### 3. Find Security Bugs como plugin con Maven

#### 3.1 Crear un proyecto Maven

File -> New -> Other

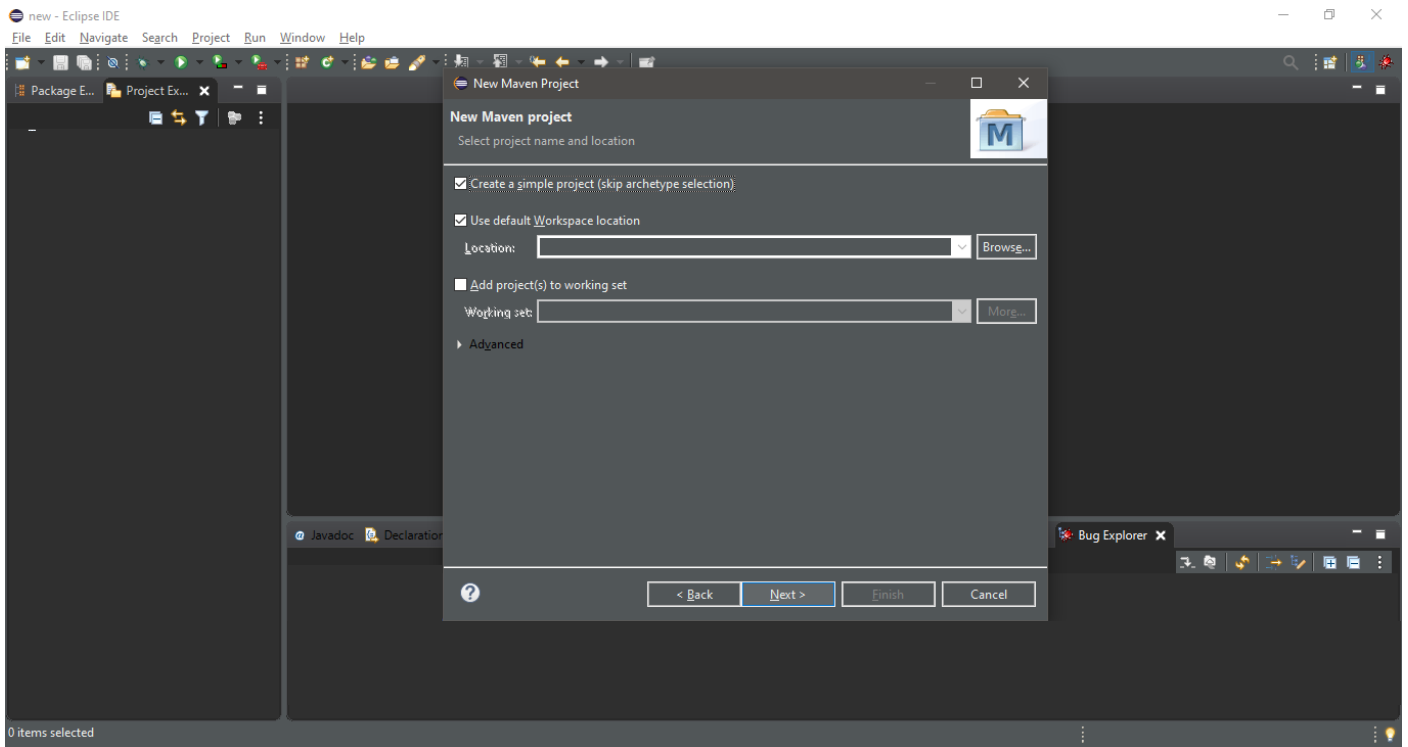


#### Seleccionar Maven Project

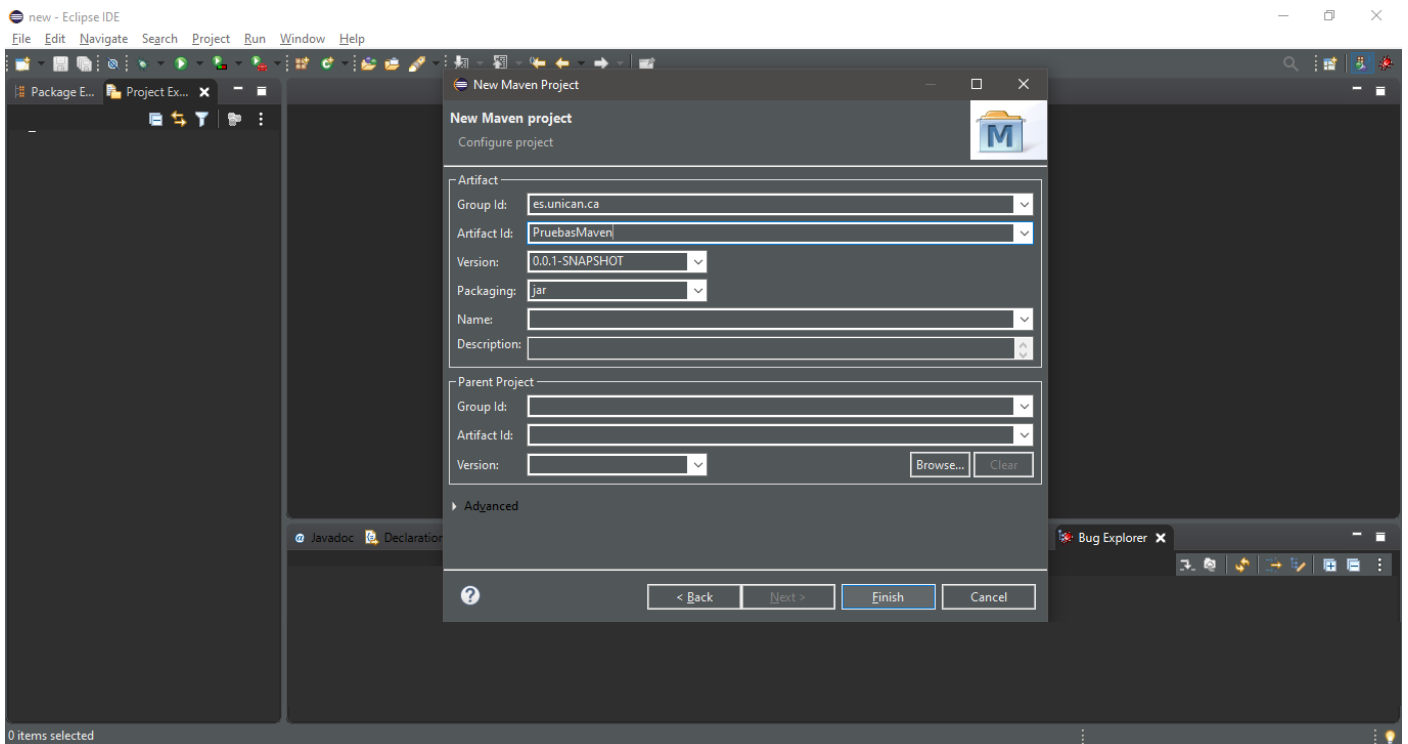




Para este caso, será necesario solamente un proyecto simple, por lo que se selecciona “Create a simple project”.

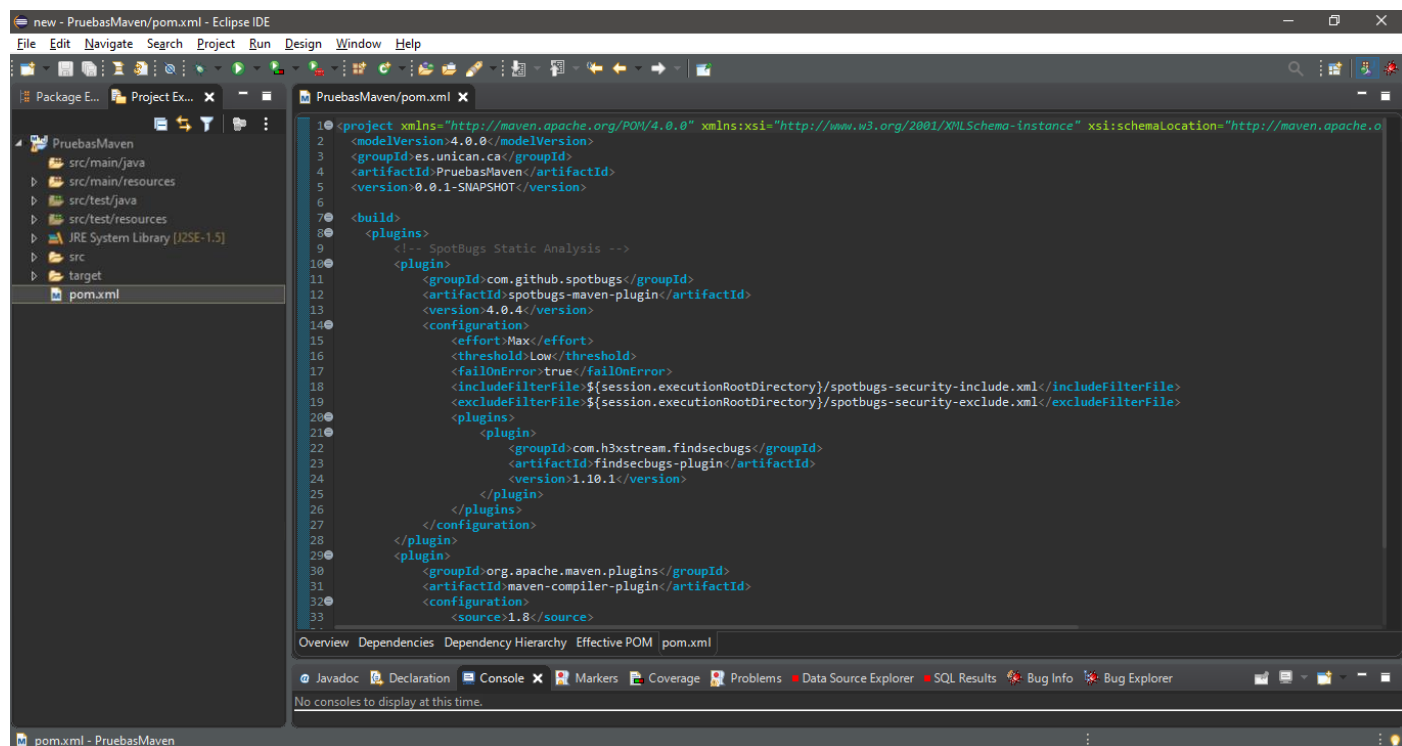


Se le da un Group Id, Artifact Id, y se selecciona el Packaging “jar” (si no está seleccionado previamente).



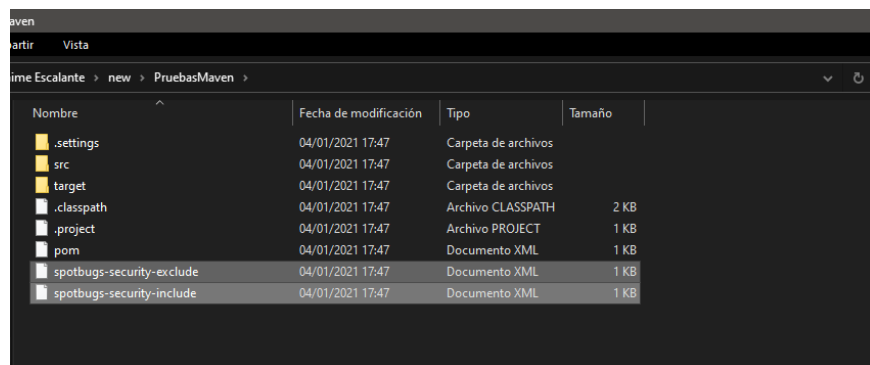
## 4. Configuración de Find Security Bugs con Maven

Modificar el archivo pom.xml que se encuentra en la raíz del proyecto, añadiendo el plugin.

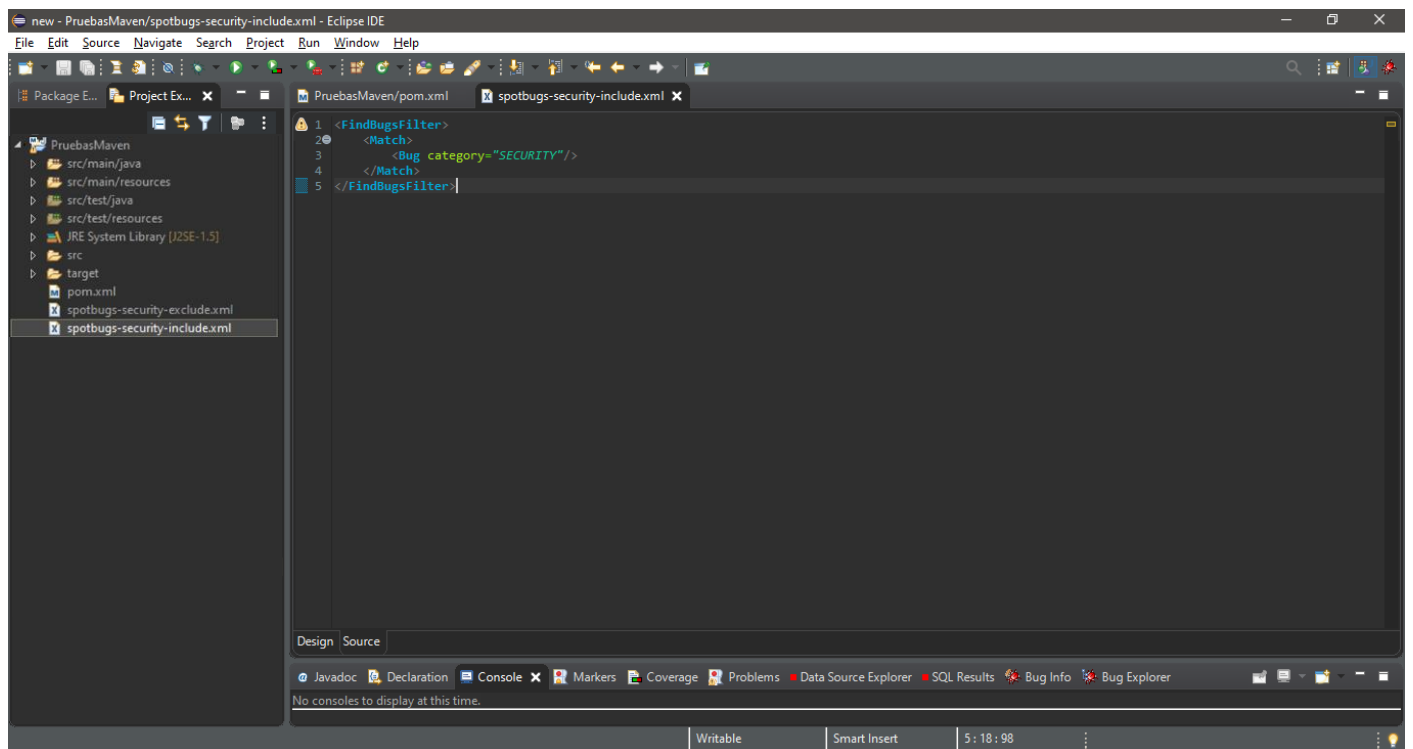


En la carpeta raíz del proyecto, añadir dos archivos .xml:

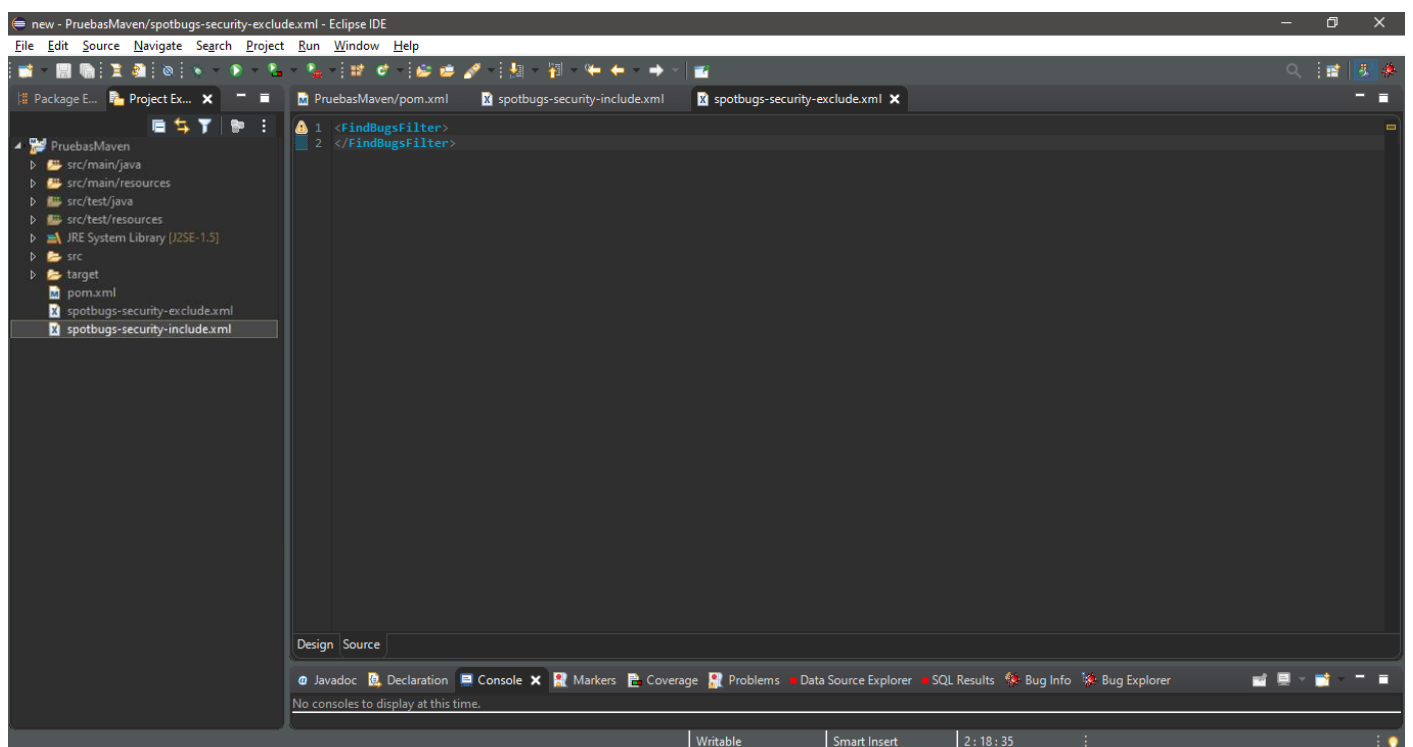
- spotbugs-security-include.xml
- spotbugs-security-exclude.xml



En el archivo spotbugs-security-include.xml se incluyen todas aquellas clases de bugs que queremos que se analicen. En este caso, solamente se incluyeron las de categoría “Security”.



Si hubiese alguna clase de bug que no se quiera analizar, se debe dejar constancia de ello en el archivo spotbugs-security-exclude.xml. En este caso, no se quiere excluir ninguna, por lo que queda de esta manera.



## 5. SonarQube con Find Security Bugs

### 5.1 Software necesario

#### 1. Java 11 o superior

Es necesario disponer de java 11 o superior para poder arrancar el servidor local de SonarQube.

URL: <https://www.oracle.com/es/java/technologies/javase-jdk11-downloads.html>

#### 2. SonarQube Community edition

Para este taller utilizaremos la última versión disponible de SonarQube Community edition.

URL: <https://www.sonarqube.org/downloads/>

#### 3. Maven

Es necesario tener también instalado Maven para lanzar el análisis de Sonar a nuestro proyecto.

URL: <https://maven.apache.org/download.cgi>

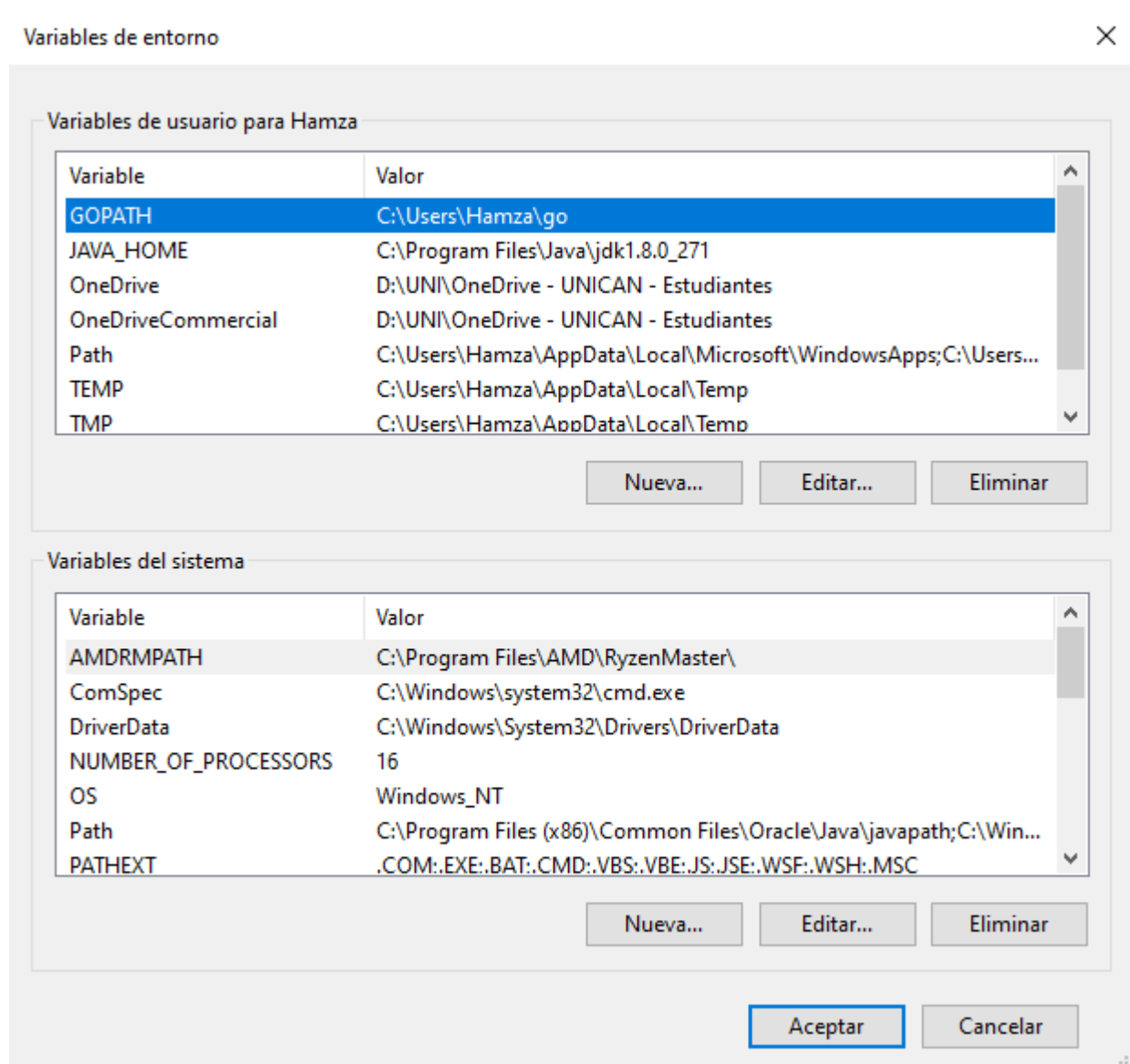
### 5.2 Configuración

#### 1. Java

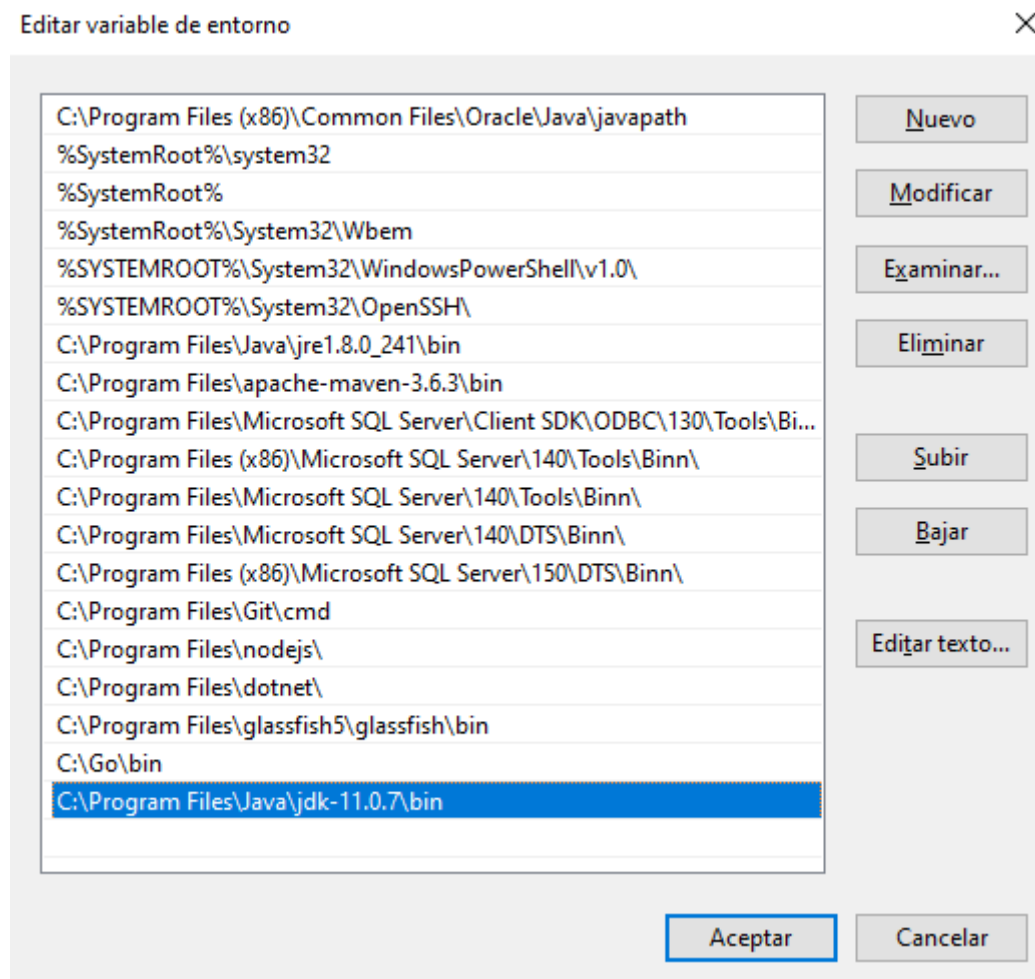
En este paso os explicaremos como agregar correctamente Java 11 al path de Windows.

Para abrir las variables de entorno, tenemos tres opciones:

- Propiedades de Mi PC -> Configuración avanzada del sistema -> Variables de entorno
- Desde la cmd, rundll32 sysdm.cpl,EditEnvironmentVariables
- WinKey+Pause -> Configuración avanzada del sistema -> Variables de entorno



Ahora en variables del sistema pincharemos sobre Path, si no está añadiremos la ruta al bin del jdk 11, tal que así.









Y pulsamos aceptar. Para comprobar que se ha añadido el java 11 correctamente, en una nueva terminal escribimos `java -version`, el mensaje con debería de salir es el siguiente:

```
C:\Users\Hamza>java -version
java version "11.0.7" 2020-04-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.7+8-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.7+8-LTS, mixed mode)
```

## 2. Sonar

Una vez descargado Sonar, añadiremos el plugin Find Security Bugs a nuestro servidor local de Sonar para poder realizar el correspondiente análisis de seguridad a nuestro proyecto.

Una vez descargado el archivo zip de SonarQube Community edition, lo descomprimos en un directorio (el que se considere oportuno). Entonces deberían de aparecer los siguientes archivos:

Nombre	Fecha de modificación	Tipo	Tamaño
 bin	09/12/2020 10:22	Carpeta de archivos	
 conf	09/12/2020 10:22	Carpeta de archivos	
 data	04/01/2021 18:01	Carpeta de archivos	
 elasticsearch	09/12/2020 10:22	Carpeta de archivos	
 extensions	04/01/2021 18:02	Carpeta de archivos	
 lib	09/12/2020 10:29	Carpeta de archivos	
 logs	04/01/2021 18:02	Carpeta de archivos	
 temp	04/01/2021 21:54	Carpeta de archivos	
 web	09/12/2020 10:29	Carpeta de archivos	
 COPYING	09/12/2020 10:22	Archivo	8 KB

Ahora colocándonos en la carpeta bin\windows-x86-64 abriremos una terminal. En la terminal escribiremos startsonar.bat y pulsamos la tecla Enter, entonces arrancará nuestro servidor de Sonar local.

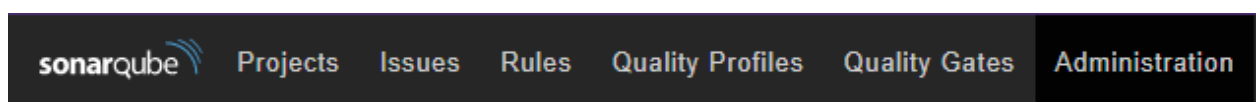
```

SonarQube
Microsoft Windows [Versión 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

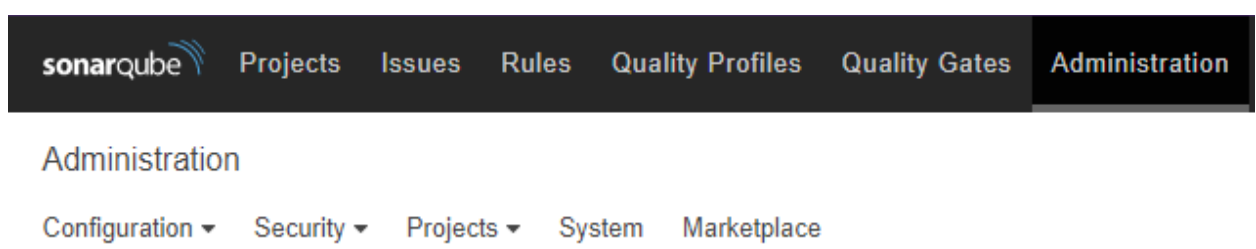
D:\Descargas\sonarqube-8.6.0.39681\bin\windows-x86-64>StartSonar.bat
Wrapper --> Wrapper Started as Console
Wrapper Launching a JVM...
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 |
jvm 1 | 2021.01.08 00:37:48 INFO app[[o.s.a.AppFileSystem] Cleaning or creating temp directory D:\Descargas\sonarqube-8.6.0.39681\temp
jvm 1 | 2021.01.08 00:37:48 INFO app[[o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:55738]
jvm 1 | 2021.01.08 00:37:48 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenamePrefix=es]] from [D:\Descarga
s\sonarqube-8.6.0.39681\elasticsearch]: C:\Program Files\Java\jdk-11.0.7\bin\java -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:
+UseCMSInitiatingOccupancyOnly -Djava.io.tmpdir=D:\Descargas\sonarqube-8.6.0.39681\temp -XX:ErrorFile=../logs/es_hs_err_pid%p.log -Des.networkaddres
s.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true
-XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dio.netty.a
lllocator.numDirectArenas=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.locale.providers=COMPAT -Xmx512m -Xms512m -XX:MaxDirec
tMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=D:\Descargas\sonarqube-8.6.0.39681\elasticsearch -Des.path.conf=D:\D
escargas\sonarqube-8.6.0.39681\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future
release.
jvm 1 | 2021.01.08 00:37:48 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | 2021.01.08 00:38:20 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2021.01.08 00:38:20 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilenamePrefix=web]] from [D:\Descar
gas\sonarqube-8.6.0.39681]: C:\Program Files\Java\jdk-11.0.7\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=D:\Descargas\s
onarqube-8.6.0.39681\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --a
dd-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.n
onProxyHosts=localhost[127.*][:1] -cp ./lib/common/*;D:\Descargas\sonarqube-8.6.0.39681\lib\jdbc\h2\h2-1.4.199.jar org.sonar.server.app.WebServer D
:\Descargas\sonarqube-8.6.0.39681\temp\sq-process3582634360640815907properties
jvm 1 | 2021.01.08 00:38:37 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2021.01.08 00:38:37 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenamePrefix=ce]] from [D:\Descarga
s\sonarqube-8.6.0.39681]: C:\Program Files\Java\jdk-11.0.7\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=D:\Descargas\son
arqube-8.6.0.39681\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError
-Dhttp.nonProxyHosts=localhost[127.*][:1] -cp ./lib/common/*;D:\Descargas\sonarqube-8.6.0.39681\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServ
er D:\Descargas\sonarqube-8.6.0.39681\temp\sq-process13476843797561319948properties
jvm 1 | 2021.01.08 00:38:41 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2021.01.08 00:38:41 INFO app[[o.s.a.SchedulerImpl] SonarQube is up

```

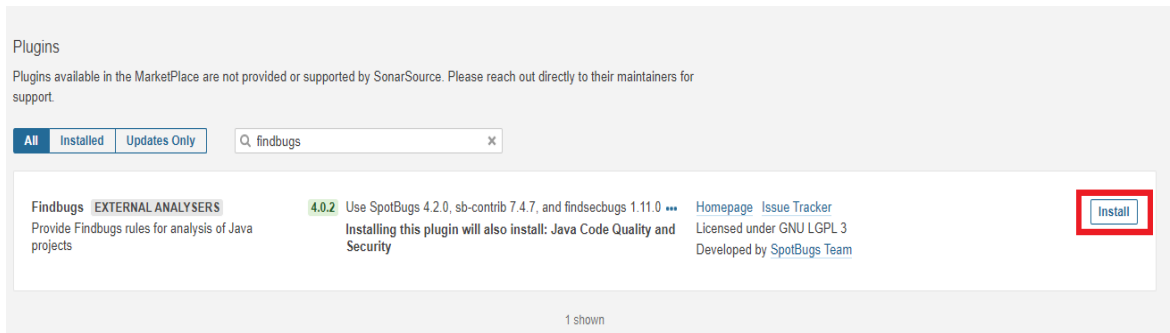
Ahora nuestro servidor local está desplegado, por defecto esta accesible en la URL <http://localhost:9000>, cuando entremos a esta dirección, se nos pedirá un usuario y una contraseña, por defecto ambas son “admin”. Una vez dentro de sonar nos dirigiremos a la pestaña “Administration”:



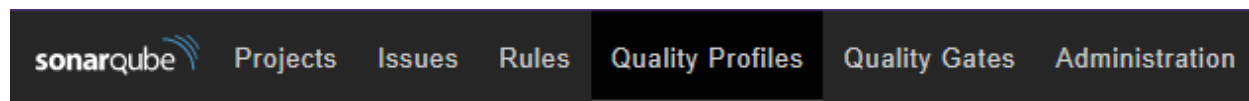
Dentro de “Administration” pincharemos sobre la pestaña “Marketplace”



Dentro de “Marketplace”, nos situaremos el apartado que dice plugins en el buscador escribiremos “findbugs” y descargaremos este plugin, pulsando sobre “Install”.



Una vez instalado el plugin será necesario reiniciar el servidor. El momento que se reinicie el servidor no redirigiremos a la pestaña “Quality Profiles”.



Si navegamos por esta pestaña veremos que vienen una serie de lenguajes de programación, y para cada uno, existen uno o más perfiles de calidad, esto son un conjunto de reglas que sonar utiliza para hacer el análisis estático del código. En este caso nos interesa el perfil “FindBugs Security Audit”, podemos apreciar que este perfil solo está disponible para el lenguaje Java. Una vez localizado dicho perfil pulsamos sobre el engranaje que vemos a la derecha y seleccionamos la opción “Set as Default”.

Java, 5 profile(s)	Projects ?	Rules	Updated	Used	
FindBugs BUILT-IN	0	444	4 days ago	4 days ago	
FindBugs + FB-Contrib BUILT-IN	0	751	4 days ago	4 days ago	
FindBugs Security Audit BUILT-IN	0	123	4 days ago	4 days ago	
FindBugs Security Minimal BUILT-IN	0	93	4 days ago		
Sonar way BUILT-IN	DEFAULT	422	4 days ago	4 da	
Java, 5 profile(s)	Projects ?	Rules	Updated	Used	
FindBugs BUILT-IN	0	444	4 days ago	4 days ago	
FindBugs + FB-Contrib BUILT-IN	0	751	4 days ago	4 days ago	
FindBugs Security Audit BUILT-IN	DEFAULT	123	4 days ago	4 days ago	
FindBugs Security Minimal BUILT-IN	0	93	4 days ago	Never	
Sonar way BUILT-IN	0	422	4 days ago	4 days ago	

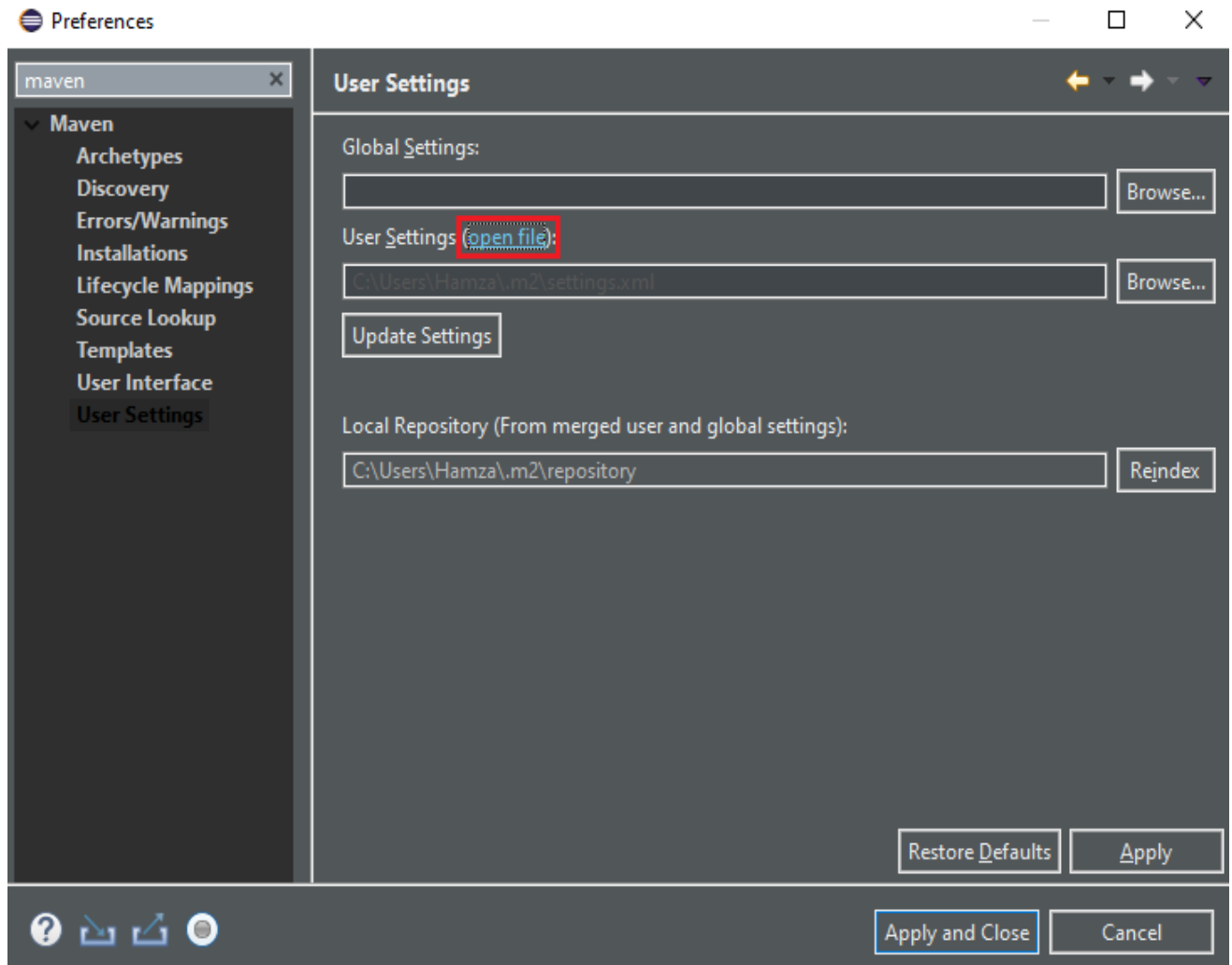
Después seleccionar la opción “Set as Default”, vemos que ahora la etiqueta “DEFAULT” esta sobre dicho perfil, esto significa que cuando lancemos un análisis de Sonar este utilizará las reglas definidas en este perfil para buscar vulnerabilidades en el código.

### 3. Maven

Finalmente, para poder lanzar Sonar con Maven necesitaremos configurar el archivo “settings.xml” de la carpeta .m2 que Maven genera en el directorio de nuestro usuario (C:\Users\User\.m2). Es posible que no tengamos el archivo “settings.xml”, en tal caso debemos crearlo nosotros manualmente.

Para editar el archivo podemos hacerlo de dos maneras, la que más cómoda resulta en cada momento.

- Dirigiéndonos directamente a dicho directorio y editando el archivo.
- Desde Eclipse. En este segundo caso, debemos en la barra principal debemos dirigirnos a la pestaña Window -> Preferences, en el buscador tecleamos “Maven” y pinchamos sobre la opción “User Settings”, en la ventana derecha vemos que tenemos una opción que dice “open file” (si no está creado el archivo “settings.xml” esta opción no será visible) pinchamos sobre ella y se abrirá el archivo “settings.xml” dentro del editor xml de eclipse.





Una vez abierto el archivo “settings.xml” añadiremos el siguiente contenido:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    https://maven.apache.org/xsd/settings-1.0.0.xsd">
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
  <profiles>
    <profile>
      <id>sonar</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <sonar.host.url>
          http://localhost:9000
        </sonar.host.url>
        <sonar.login>
          usuario
        </sonar.login>
        <sonar.password>
          contraseña
        </sonar.password>
      </properties>
    </profile>
  </profiles>
</settings>
```