******

***DOCUMENTACIÓN***

***Manual de referencia del proyecto JoinHub.***

***SNAPLIX Software©***

**ÍNDICE:**

1. Instalación del proyecto.
   1. Herramientas CASE necesarias.
   2. Proceso de instalación del proyecto.
   3. Ejecutar el proyecto en un servidor local.

**RECURSOS RELEVANTES:**

* Link Git with Eclipse IDE:

<https://wiki.eclipse.org/EGit/User_Guide/Getting_Started>

* Serving web content with Spring and Java:

<https://spring.io/guides/gs/serving-web-content/>

# Bibliographie

**CONEXIÓN DE BASE DE DATOS CON JAVA** [En ligne] / aut. Perez Aleksandr Paul Quito. - 19 03 2019. - https://fr.scribd.com/doc/13421106/CONEXION-DE-BASE-DE-DATOS-CON-JAVA-Aleksandr-Quito-Perez.

**Java Database Connectivity with 5 Steps** [En ligne]. - https://www.javatpoint.com/steps-to-connect-to-the-database-in-java.

1. **Instalación del proyecto.**

En este apartado se ofrece, a disposición de cualquiera, la posibilidad de modificar el código fuente y/o consultar el proyecto, indicando aquí las herramientas necesarias para ello. Cabe destacar que, algunas herramientas no son necesariamente obligatorias, pero sí recomendables, puesto que son aquellas con las que se fue llevado a cabo el proyecto.

# Herramientas CASE necesarias.

Para poder realizar modificaciones en el proyecto, son necesarias las siguientes herramientas:

* Entorno de Desarrollo Integrado ***Eclipse*** (Opcional, recomendado). Se puede descargar desde: <https://www.eclipse.org/ide/>.
* Plugin de Eclipse ***Spring Tools*** (Puede obtenerse desde el “Marketplace” de Eclipse; Dentro de Eclipse ir a: Help -> Eclipse Marketplace).
* Gestor de dependencias ***Maven***.

# Proceso de instalación del proyecto.

* + 1. En primer lugar, será necesario tener instalado el Kit de Desarrollo de Java en su última versión: **A partir de la versión JDK 12.0.1**.

Se puede obtener desde la página oficial de Java: <https://www.oracle.com/technetwork/java/javase/downloads/index.html>.

* + 1. Instalar el plugin de Git para Eclipse ***EGit***. Se puede obtener desde: Help -> Eclipse Marketplace. Consultar el apartado de “Recursos relevantes” para más información sobre el uso del plugin.
    2. Una vez instalada la anterior extensión, clonar el repositorio desde GitHub. Para ello, ir a: Window -> Show View -> Other.

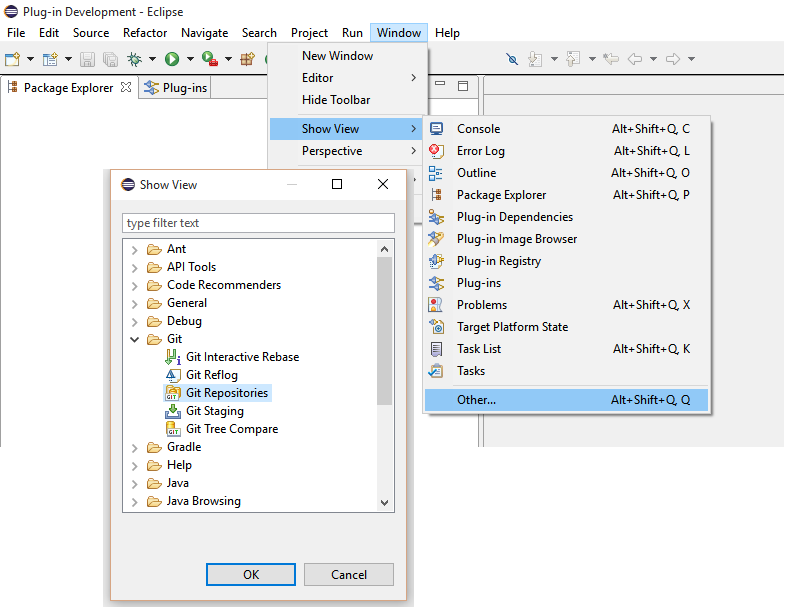


Fig. 1

Aparecerá una ventana secundaria, dentro de la cual seleccionar *Git Repositories* y, seguidamente, hacer click sobre *OK*.

A continuación, de forma automática, se abrirá una pestaña (*Git Repositories*) bajo la pestaña *Package Explorer*. En la misma, hacer click sobre: Clone a Git repository.

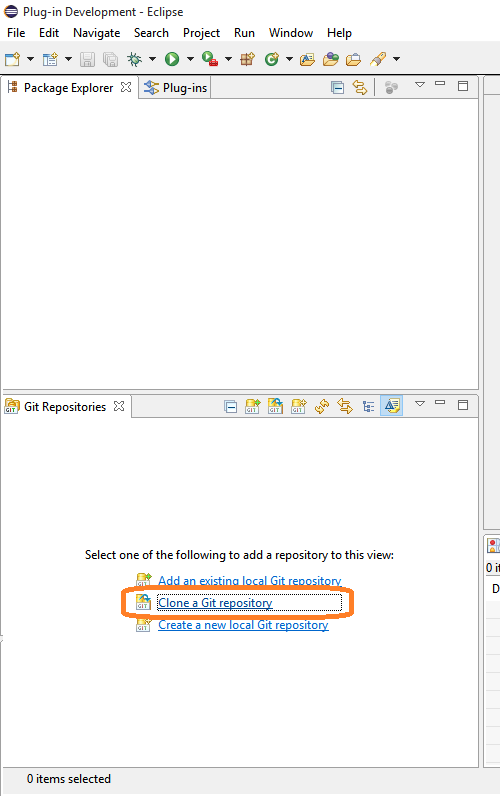


Fig. 2

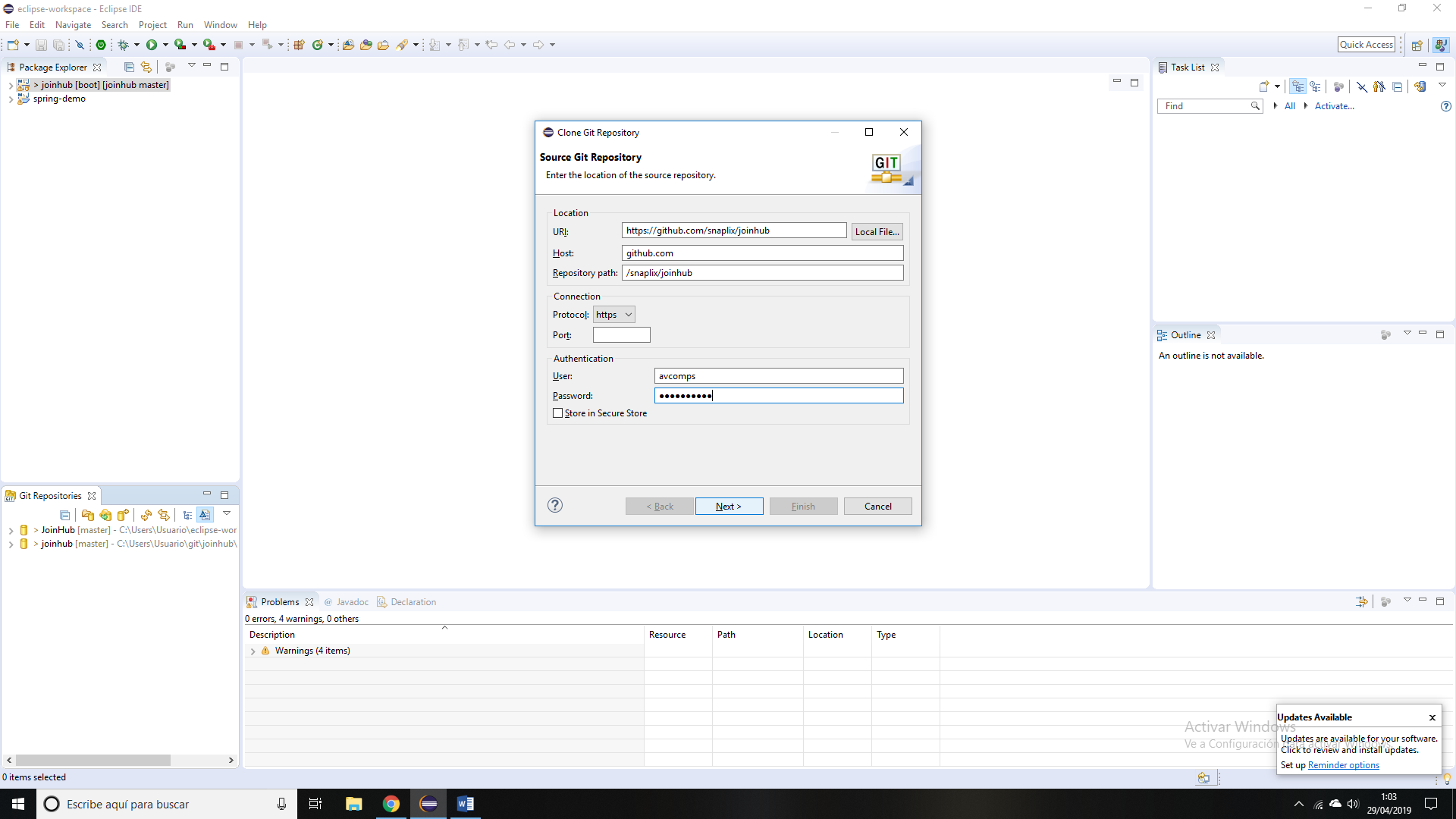


Fig. 3

Seguidamente, aparecerá una nueva ventana secundaria, en la que se debe introducir los datos de localización del repositorio, así como el nombre de usuario y la contraseña de una cuenta de Git contribuyente del repositorio.

Tras ello, hacer click sobre Next >.

Los datos a introducir son los siguientes:

1. **Localización:**
   1. **URI**: <https://github.com/snaplix/joinhub>.
   2. **Host**: github.com.
   3. **Repository path**: /snaplix/joinhub.
2. **Conexión:**
   1. **Protocol**: https.
   2. **Port**: 443 o dejar en blanco.
3. **Autenticación:**
   1. **User**: nombre de usuario de una cuenta de Git contribuyente.
   2. **Password**: contraseña de la correspondiente cuenta.

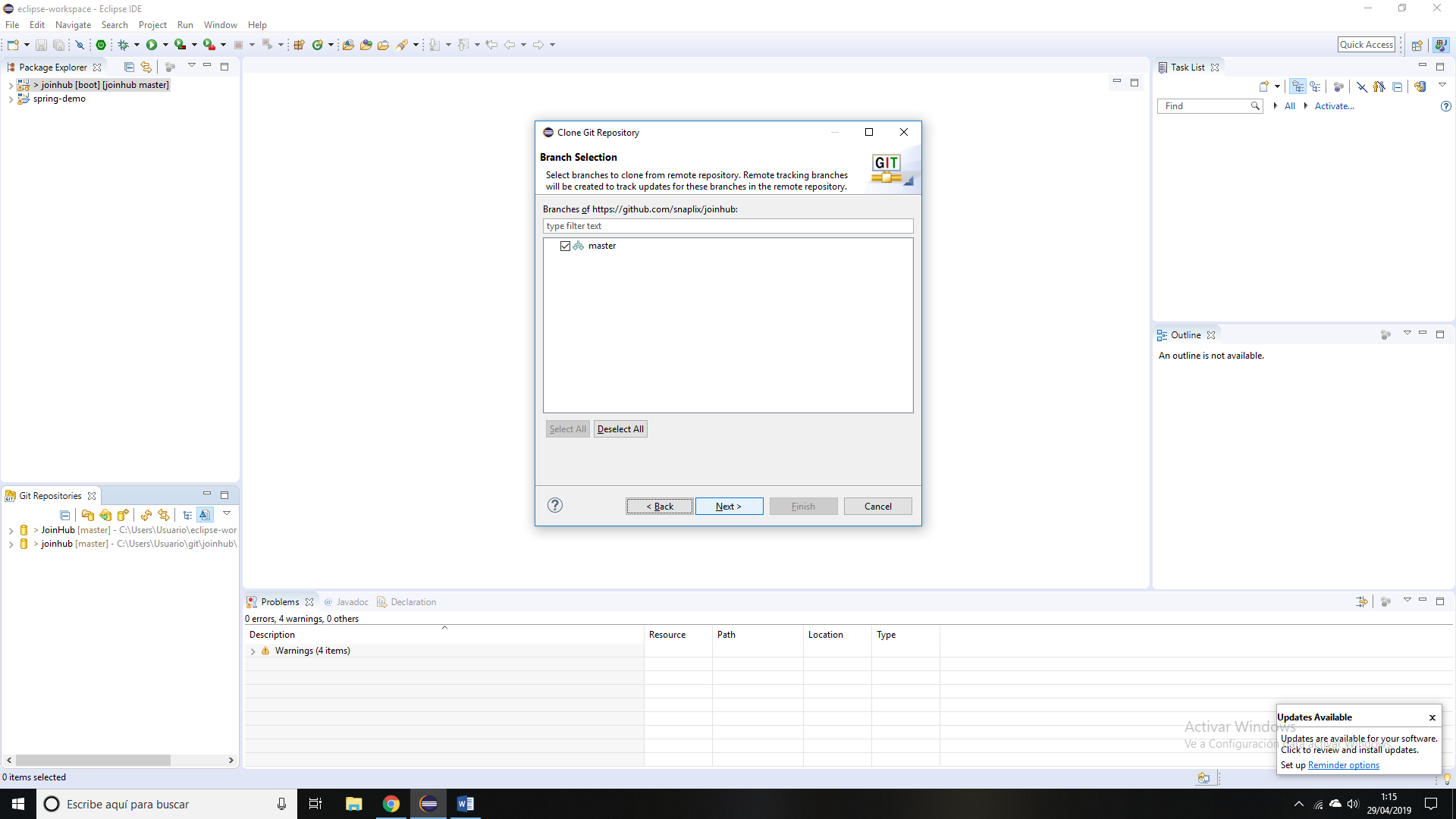


Fig. 4

A continuación, seleccionar la rama *“master”*. Seguidamente, hacer click sobre Next >.

Llegados a este punto, Eclipse generará una localización final para guardar el repositorio (Se puede elegir libremente el destino local del proyecto).

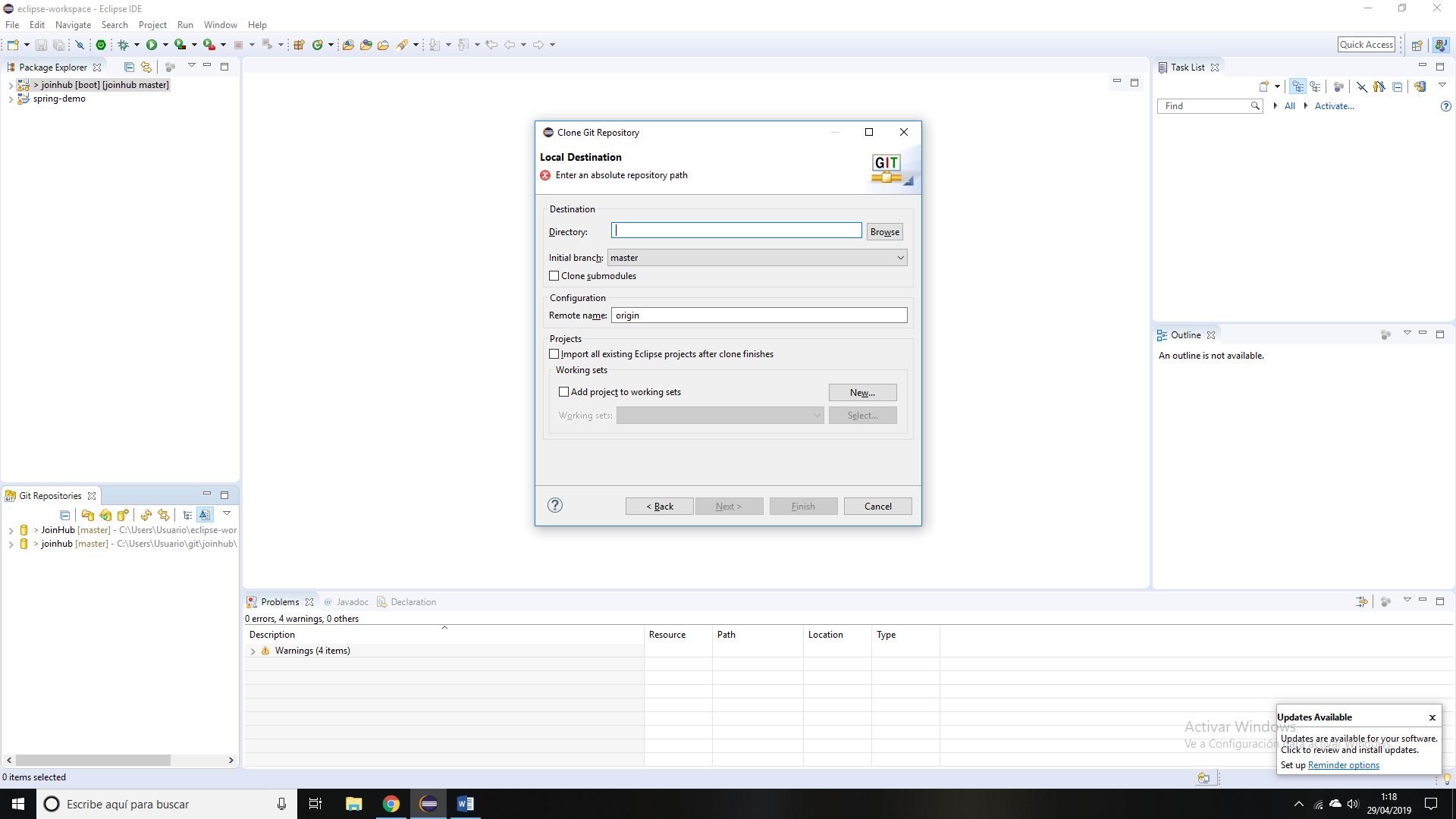


Fig. 5

En *“Remote name”* añadir: *origin*. Finalmente, hacer click sobre Finish.

Una vez realizados todos los anteriores pasos, deberá aparecer algo como lo siguiente:

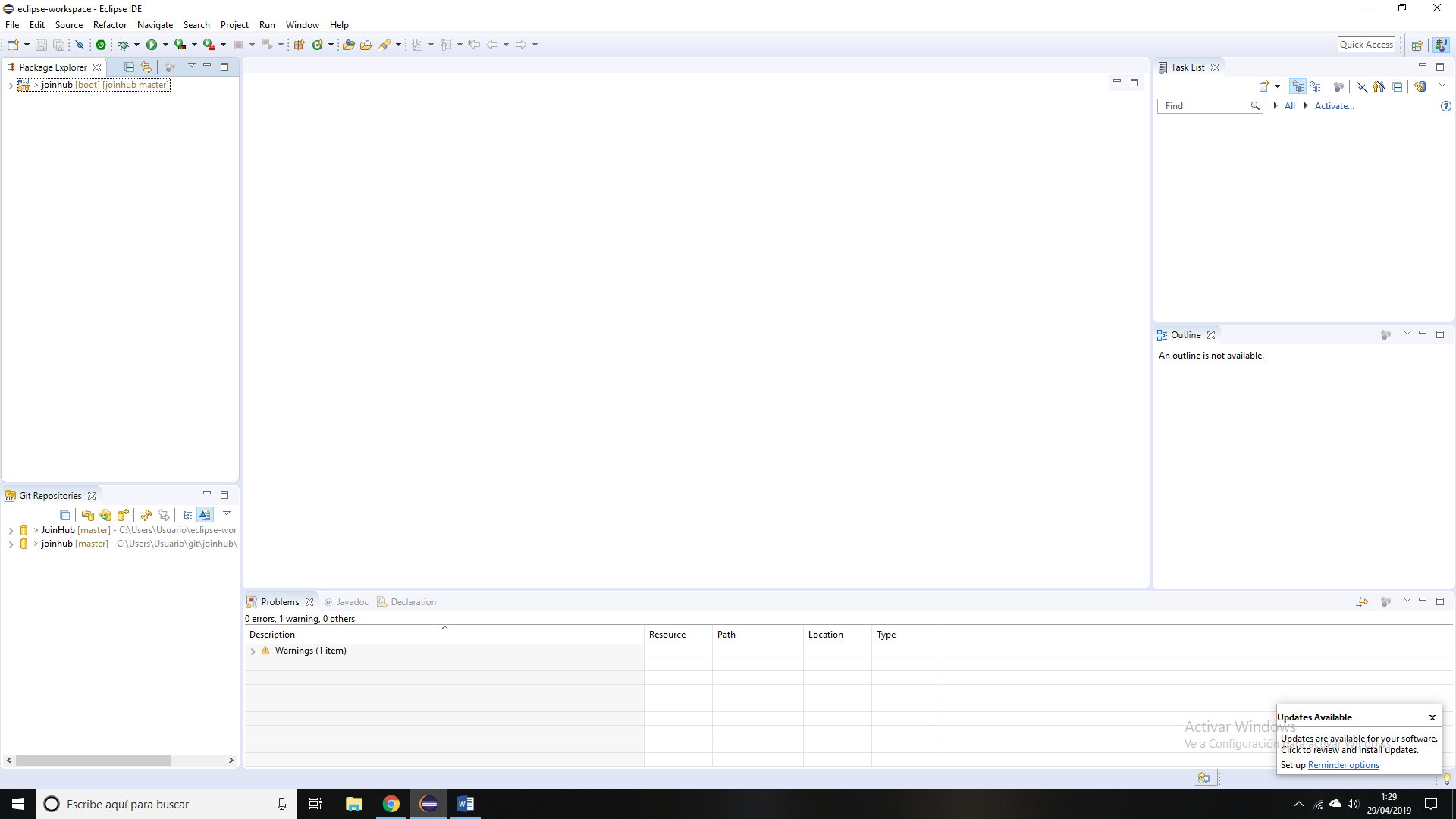


Fig. 6

Como se puede apreciar en la *Fig. 6*, aparece el nombre del proyecto, *“joinhub”*, junto a las etiquetas [boot] y [joinhub master]. La primera indica que se ha llevado a cabo el correcto análisis del proyecto y el mismo es ejecutable y, la segunda, indica la rama a la que pertenece el repositorio, la maestra (principal) en este caso.

El símbolo amarillo bajo la J de Java, el lenguaje principal del proyecto, indica que todos los cambios realizados han sido “comiteados” y mandados al repositorio remoto.

* + 1. Para poder realizar las operaciones de control de versión pertinentes dentro del proyecto, hacer click derecho sobre el proyecto en la pestaña *Package Explorer* y, a continuación, ir a Team. Ahí aparecerán todas las operaciones disponibles, así como otros parámetros de configuración.

# Ejecutar el proyecto en un servidor local.

Para ejecutar el proyecto en cualquier servidor local, en el que se encuentre almacenado el proyecto, realizar lo siguiente:

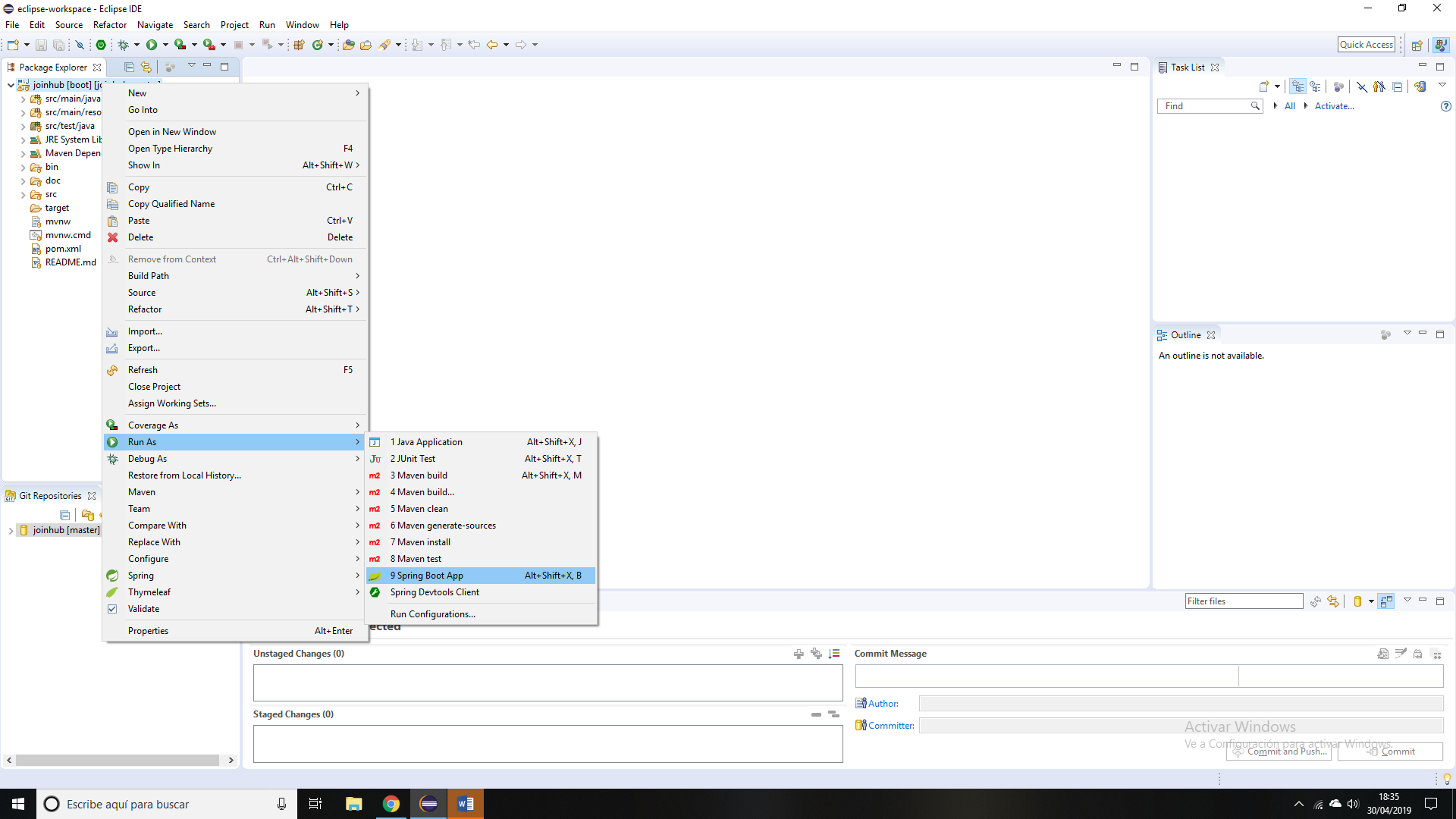


Fig. 7

Hacer click derecho sobre el proyecto en la pestaña del explorador de paquetes o explorador de proyectos, indistintamente, e ir a: Run as -> Spring Boot App. Tras ello, esperar unos segundos hasta que Spring haya terminado de configurar el servidor y otros ajustes.

Una vez realizado lo anterior, se puede acceder, desde cualquier navegador, a la aplicación web a través de la siguiente dirección:

<localhost:8080>

En caso de que salte algún error relacionado con el archivo “**tools.jar**” del JDK, comprobar que se haya instalado correctamente el JDK, operación que se puede realizar accediendo a: Window -> Preferences -> Java -> Installed JREs.