

Práctica 3 RMI - Uso del IDE IntelliJ

Desarrollo de Sistemas Distribuidos

Jose Luis Gallego Peña - DSD2 A continuación vamos a explicar resumidamente cómo usar el IDE IntelliJ para trabajar en la práctica 3 con Java RMI. En este caso se trabaja sobre el ejemplo 3.

Una vez instalado, abrimos el programa y creamos un nuevo proyecto en: **File -> New -> Project** y elegimos **Java**, dándole a siguiente y eligiendo el nombre de la aplicación.

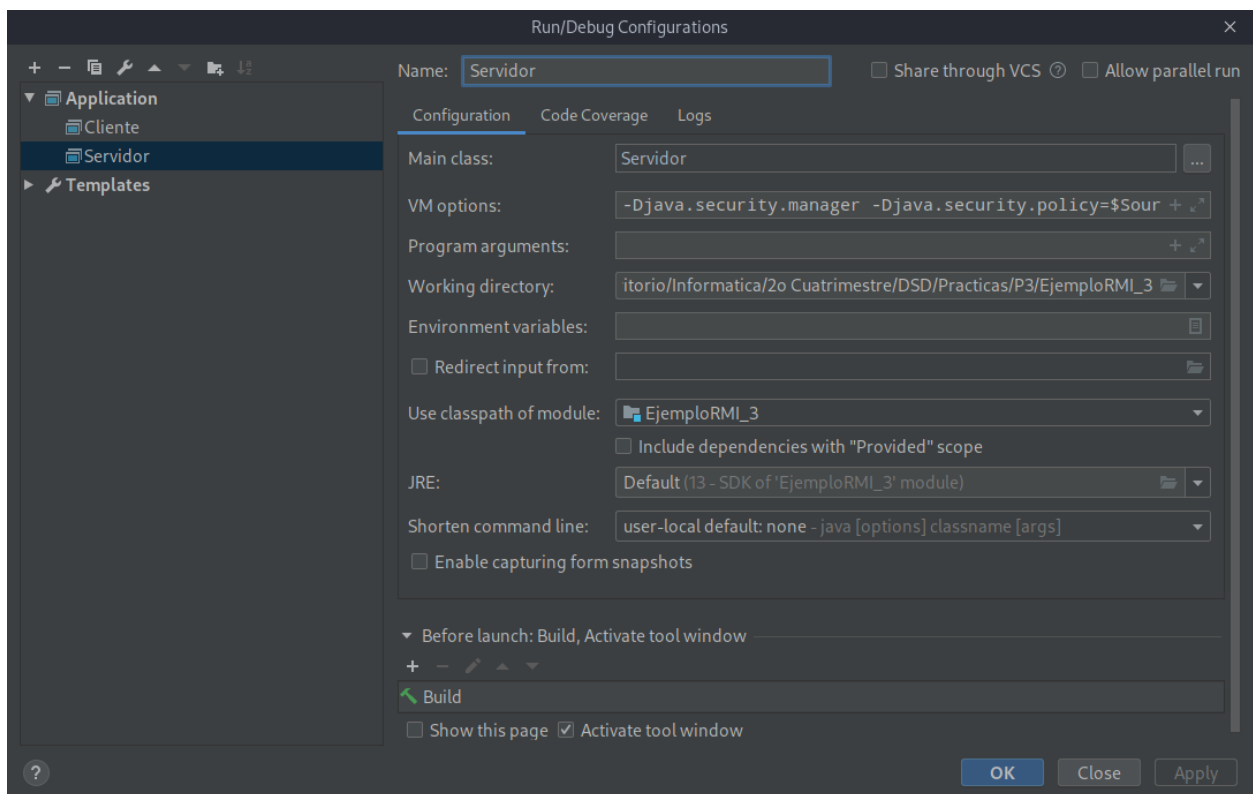
Esto creará un nuevo módulo con una estructura de carpetas. La más importante es src, en la cual crearemos los archivos haciendo click derecho: **New -> Java class** y ahí le ponemos el nombre a la clase.

Para poder compilar tanto el cliente como el servidor hay que crear una configuración especial de compilación. Para ello, nos vamos al menú superior y seleccionamos **Run -> Edit configurations**. En esta pestaña que se abre, pulsamos el botón **+** y seleccionamos **Application**. Hay que hacer esto dos veces, una para el cliente y otra para el servidor.

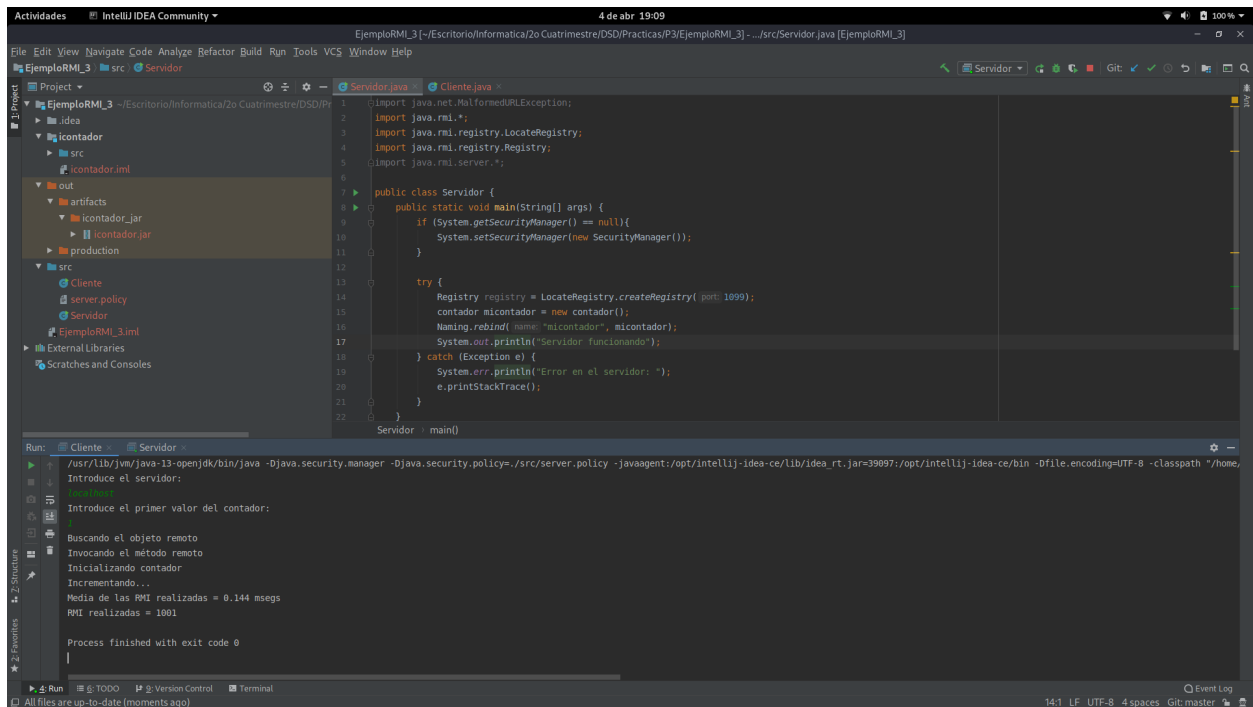
En la parte de main class le damos a los tres puntos y seleccionamos el fichero con la clase servidor o la clase cliente, y luego en VM options escribimos lo siguiente:

```
-Djava.security.manager -Djava.security.policy=./src/server.policy
```

Esto es para que al compilar coja el fichero server.policy.



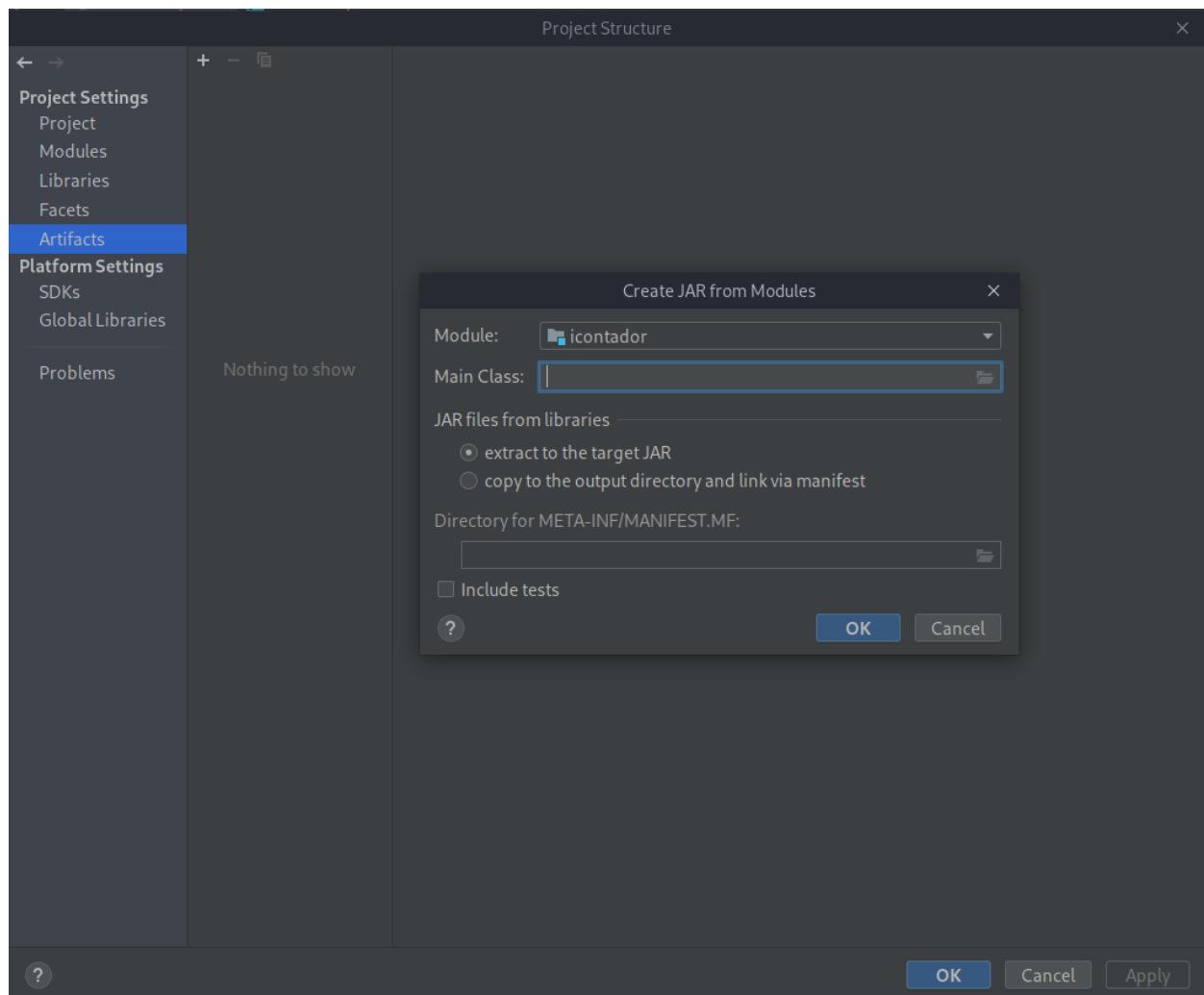
Una vez hecho esto, solo hay que darle al icono de play arriba a la derecha o en el menú run.

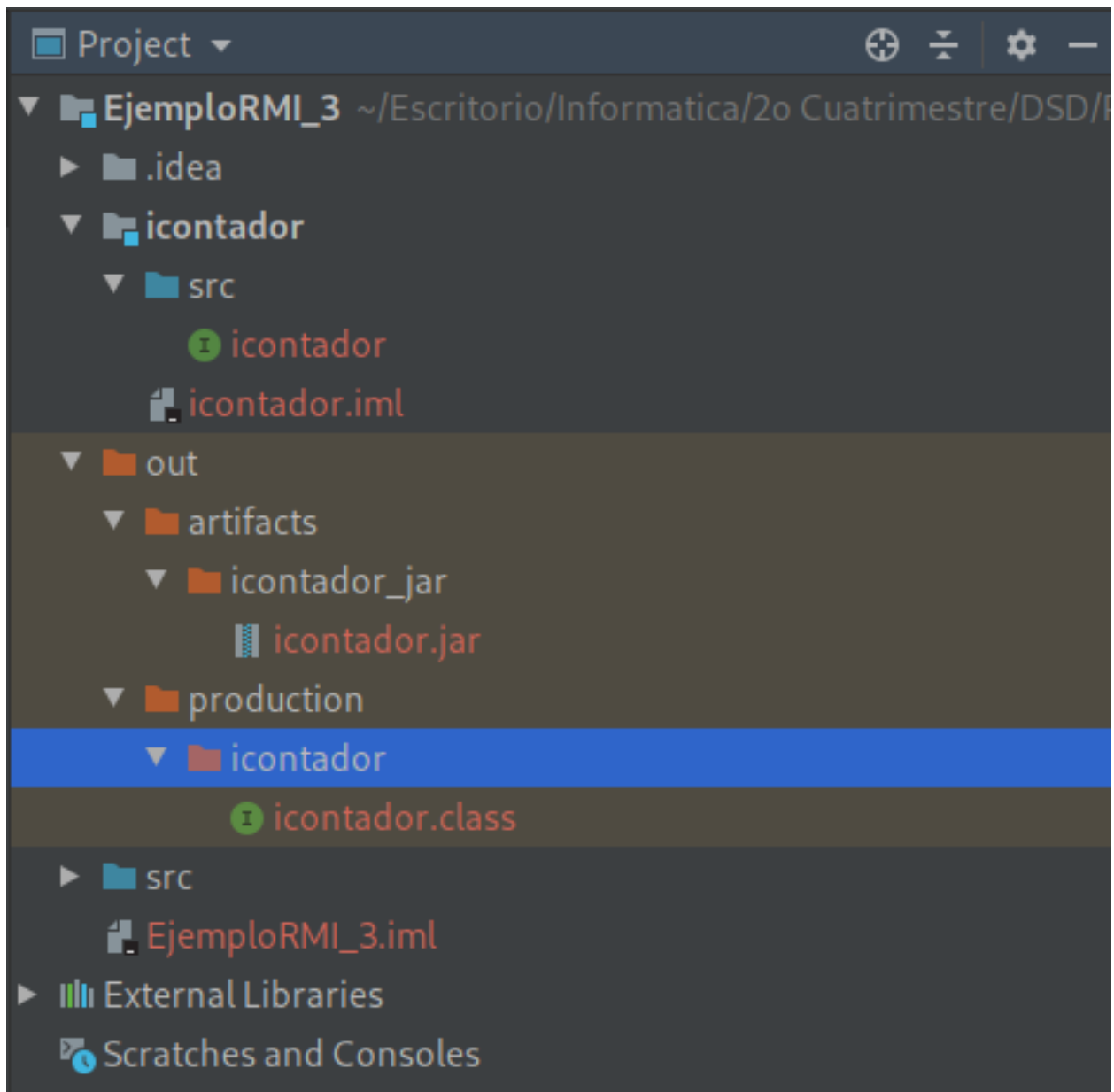


Sin embargo, en el caso del Ejemplo 3 tenemos que hacer algo especial.

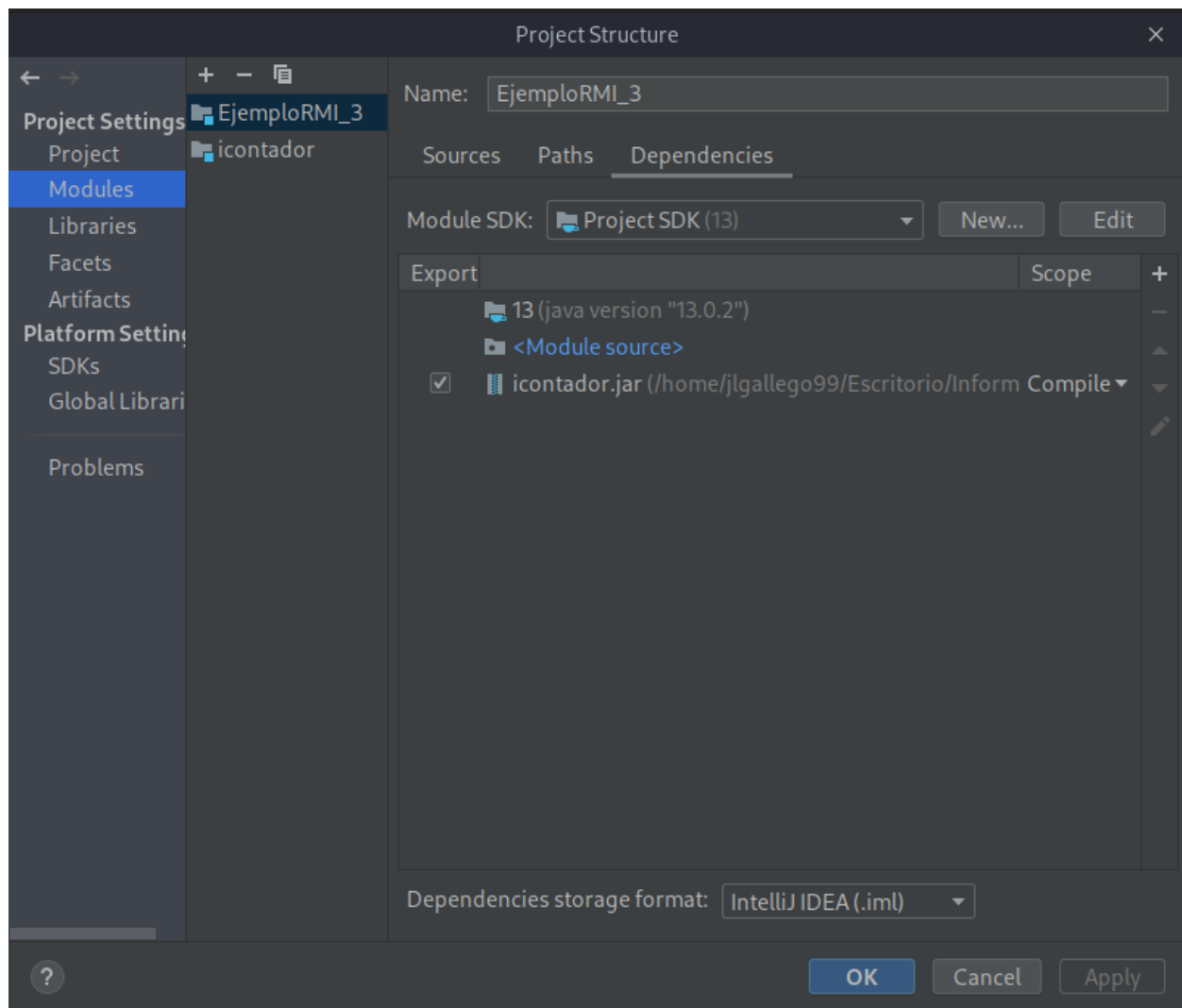
Puesto que queremos la interfaz en un nuevo modulo, hacemos click derecho en el fichero principal en la estructura de arbol y creamos un nuevo módulo llamado **icontador**, que tendrá su propia estructura de carpetas. En su src declaramos y definimos la interfaz.

Una vez hecho esto, para poder generar el .jar hacemos lo siguiente: **File -> Project Structure -> Artifacts** y ahí seleccionamos el icono **JAR file -> From modules with dependencies** y seleccionamos el módulo (la carpeta) que acabamos de crear. Finalmente para generar el .jar lo que hacemos es ir a **Build -> Build Artifacts -> icontador**.





Sin embargo, para poder crear una dependencia entre el modulo principal y esta nueva biblioteca hay que añadir este .jar. Para ello click derecho en el proyecto: **Open module settings -> Modules -> Dependencies -> + -> JARs or directories** y buscamos el archivo jar.



Con esto ya podemos ver los archivos en la carpeta **External Libraries** y podemos compilar bien el programa.

Nota: En mi caso, solo en el ejemplo 3, no me detectaba bien la variable *Sourcepath* así que cambié la opción de compilación poniendo la ruta exacta:

```
-Djava.security.manager -Djava.security.policy=./src/server.policy
```