

## Sessió 1: Introducció a IntelliJ

### Objectius:

1. Instal·lació de IntelliJ com a IDE de desenvolupament
2. Crear un primer programa
3. Conèixer com importar un projecte de netbeans en l'entorn IntelliJ

### 1. Introducció

IntelliJ IDEA Community Edition és un IDE desenvolupat per l'empresa JetBrains molt adient per desenvolupar aplicacions Java i Android. Podeu obtenir una còpia per instal·lar la darrera versió amb llicència educacional per un any a l'enllaç:

<https://www.jetbrains.com/community/education/?fromMenu>

Després del primer any, es pot anar renovant anualment.

### 2. Principals diferències entre Netbeans i IntelliJ

A diferència de NetBeans, IntelliJ treballa sobre *modules*. Un projecte pot doncs contenir un o més *module* dins del mateix projecte IntelliJ. Cada *module* tindrà les dependències i el seu fitxers fonts i podran ser independents.

Posem per exemple una aplicació client i una aplicació servidor. Cada aplicació podria ser un *module* independent i podríem tenir per exemple una llibreria com un tercer *module* que dones suport a les altres dues aplicacions. Un altre ús seria partir una única aplicació desenvolupada en diferents *modules* sota el clàssic patró MVC on cada una de les parts podria ser un *module* dins d'un projecte.

En la següent taula es poden veure els canvis de nomenclatura entre els projectes de [netbeans](#) i els de [IntelliJ](#). Fixeu-vos que el que abans eren Projectes ara són mòduls i es reserva la paraula Projecte per a definir tot el conjunt de mòduls.

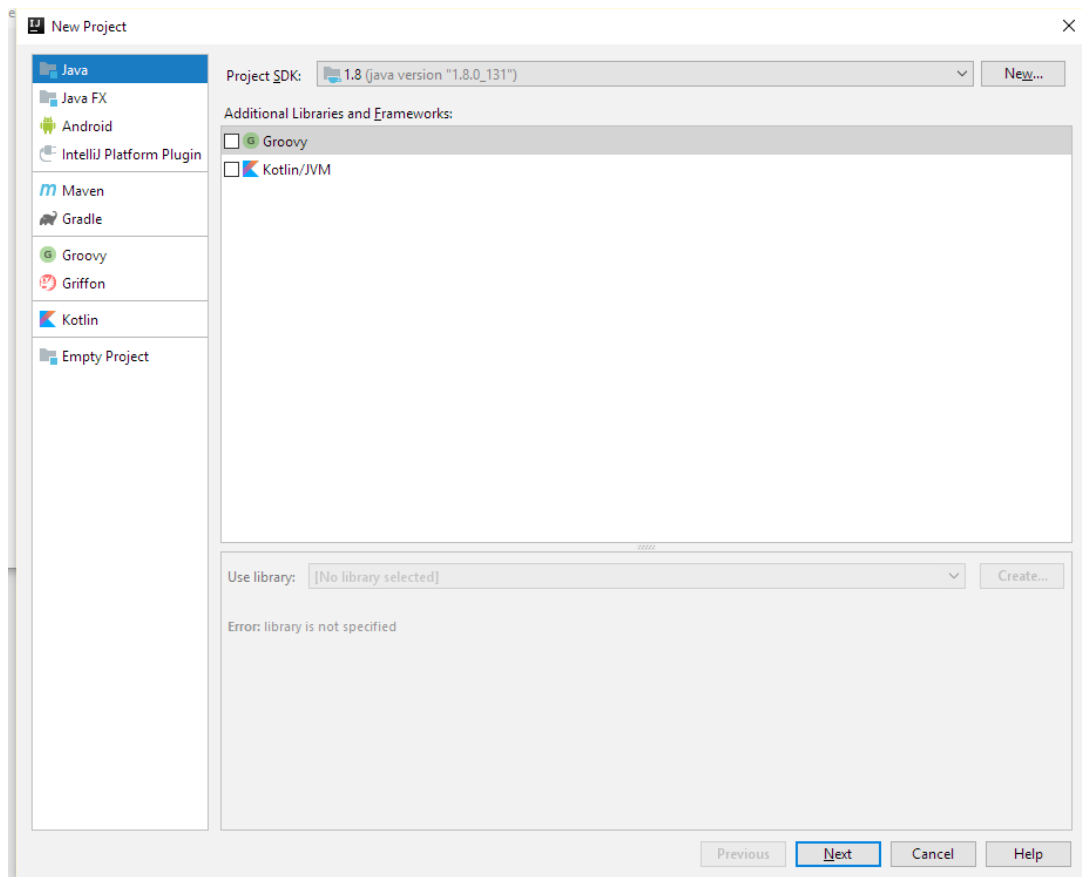
NetBeans	IntelliJ IDEA
Project	Module
Global library	Global library
Project library	Module library
Project dependency	Module dependency

### 3. Obrint IntelliJ (construint un nou projecte)

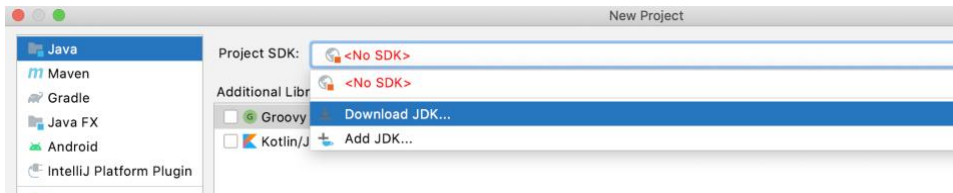
Per a construir un nou projecte, es defineix des del menú principal o des del menú de Files. Per ara es fa un projecte amb un mòdul únic. En iniciar-se IntelliJ surt un menú principal on ja es pot crear el projecte:



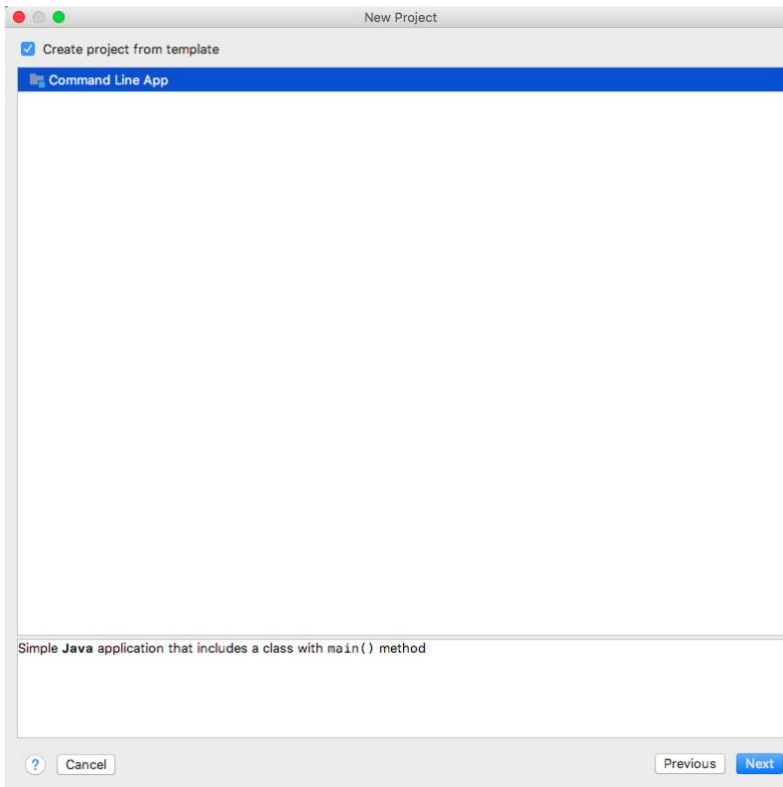
En aquest cas, cal fer *New Project* i seleccionar Java



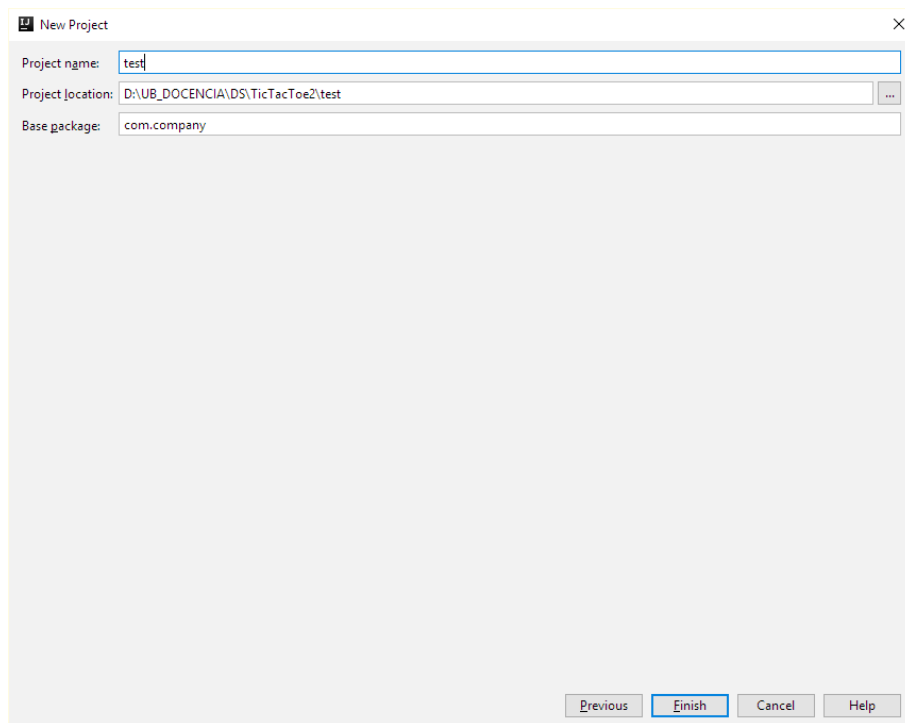
Cal revisar que teniu un JDK instal·lat en el vostre ordinador. Sinó, en aquest pas, us permet instal·lar-lo.



Un cop baixat i instal·lat el SDK, prémer Next i apareixerà la següent finestra. Aquí, pots triar si crear el projecte des d'un Template. Aquí escollirem Command Line App per a crear per defecte un main.



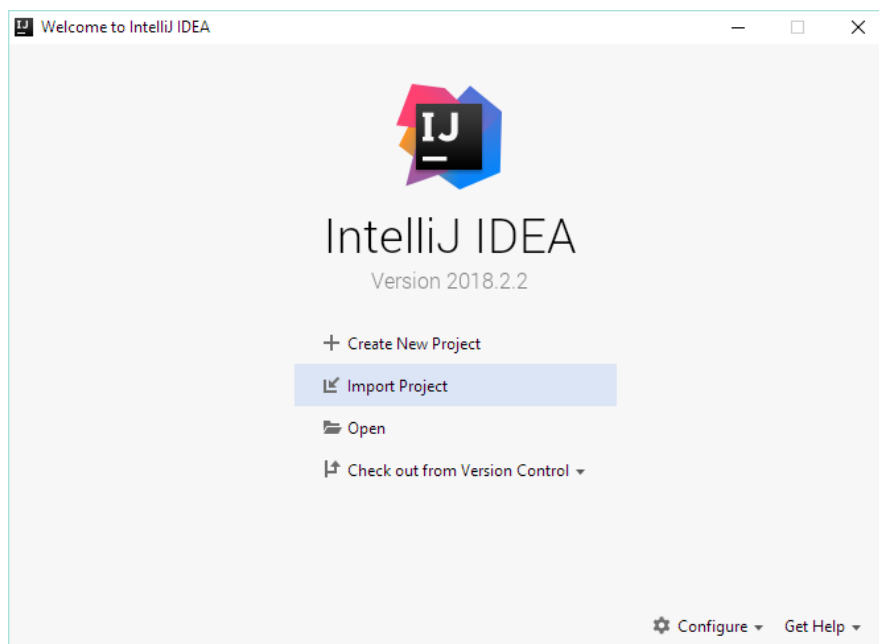
Finalment es pregunta el nom del Projecte. Sinó es selecciona un nom diferent del mòdul que el del projecte, el projecte és equivalent al mòdul. Recordeu que en IntelliJ un projecte pot tenir diferents mòduls amb diferents configuracions i dependències.



Per a més informació veu el vídeo: [https://youtu.be/c0efB\\_CKOYo](https://youtu.be/c0efB_CKOYo)

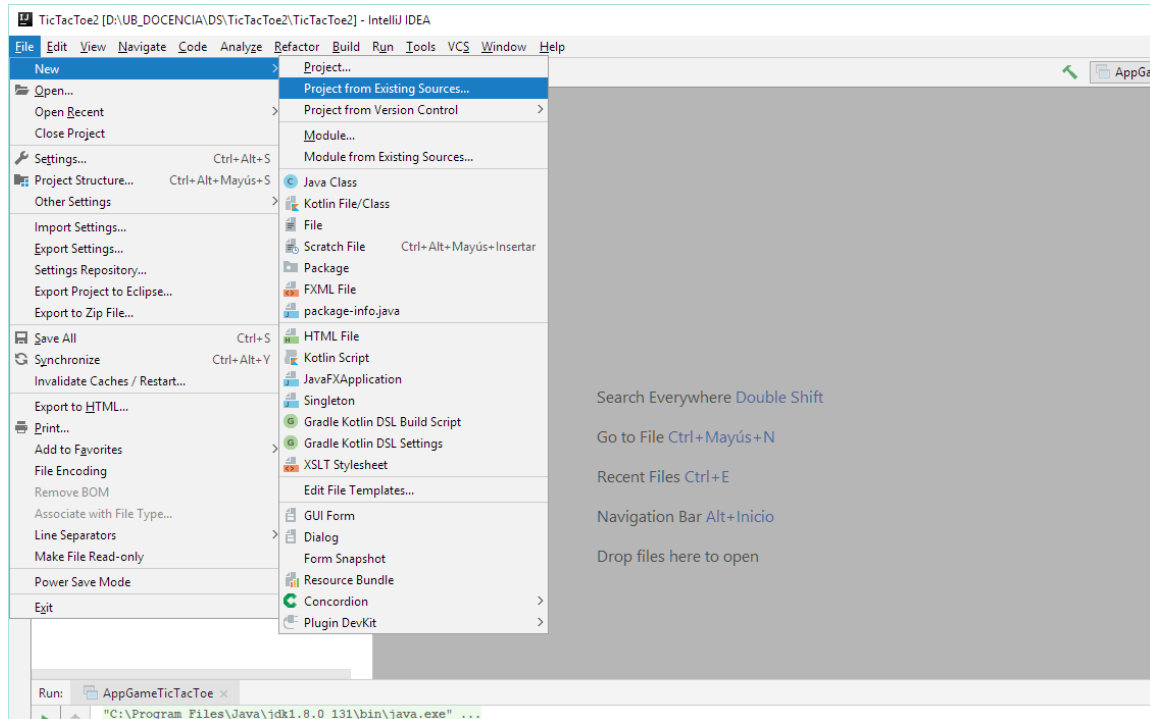
## 4. Importació de projectes i configuració de llibreries

Des de la pantalla principal, seleccionant Import Project:

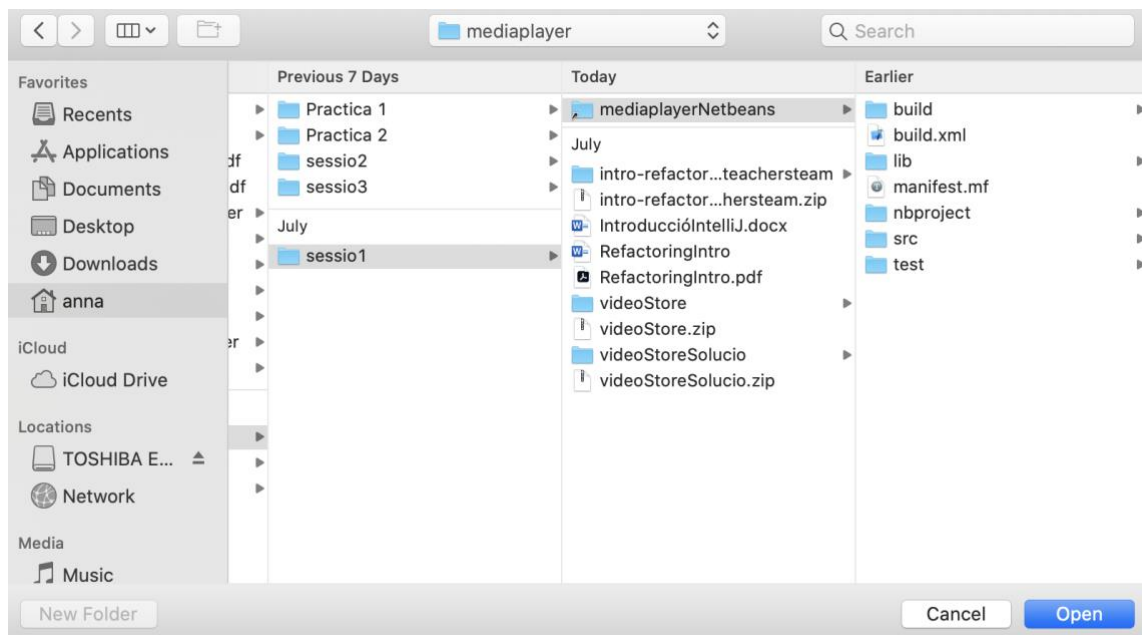


O si ja es té un projecte obert, per importar un projecte NetBeans al nostre IDE haurem de seguir aquestes passes:

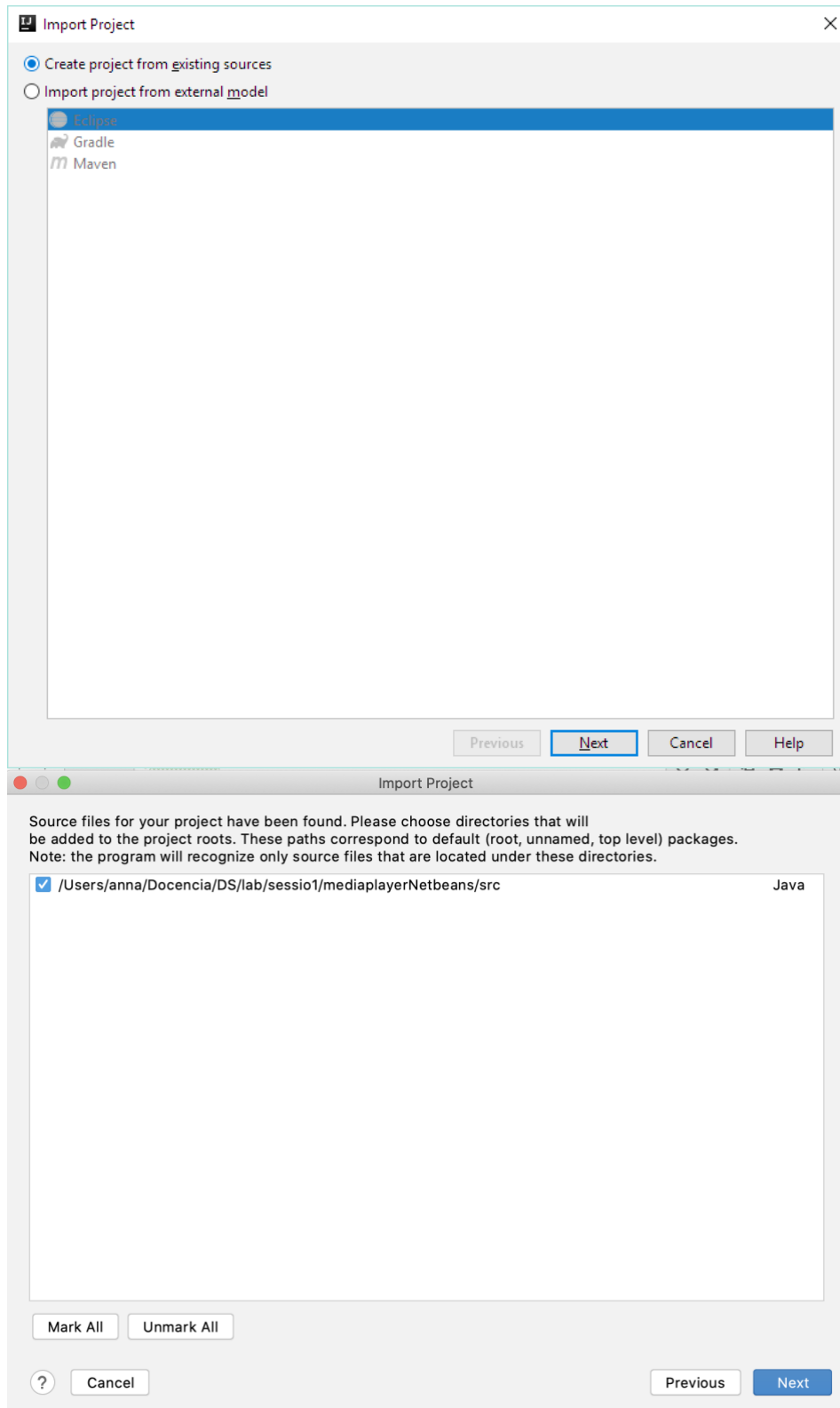
- Anem a (*File*→*New*→*Project from Existing Source*)



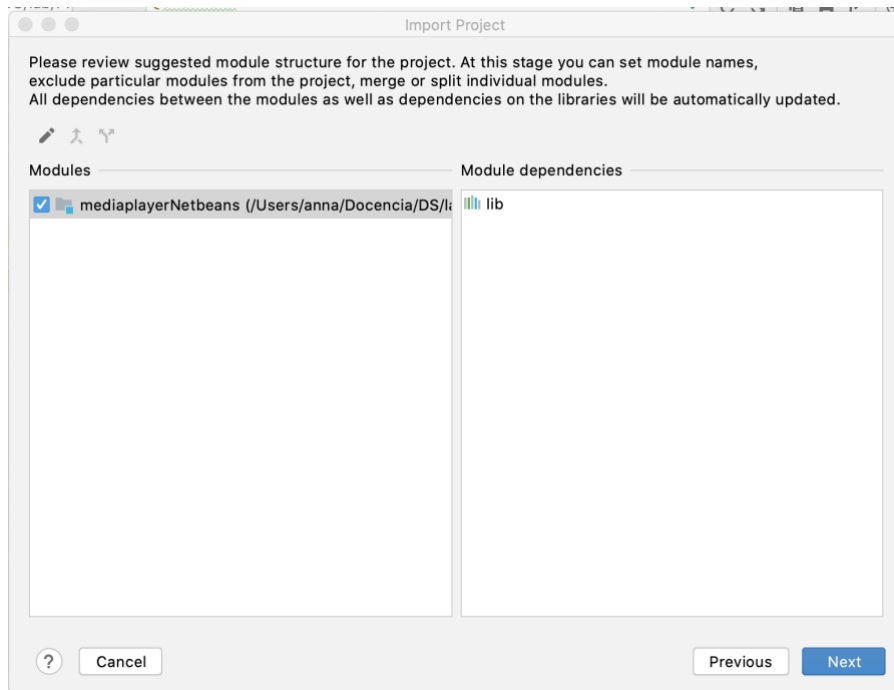
- Busquem el directori arrel del nostre projecte de NetBeans (busqueu un projecte d'un any anterior que tingueu disponible, per exemple) o el projecte de [mediaplayerNetbeans](#) que teniu al campus.



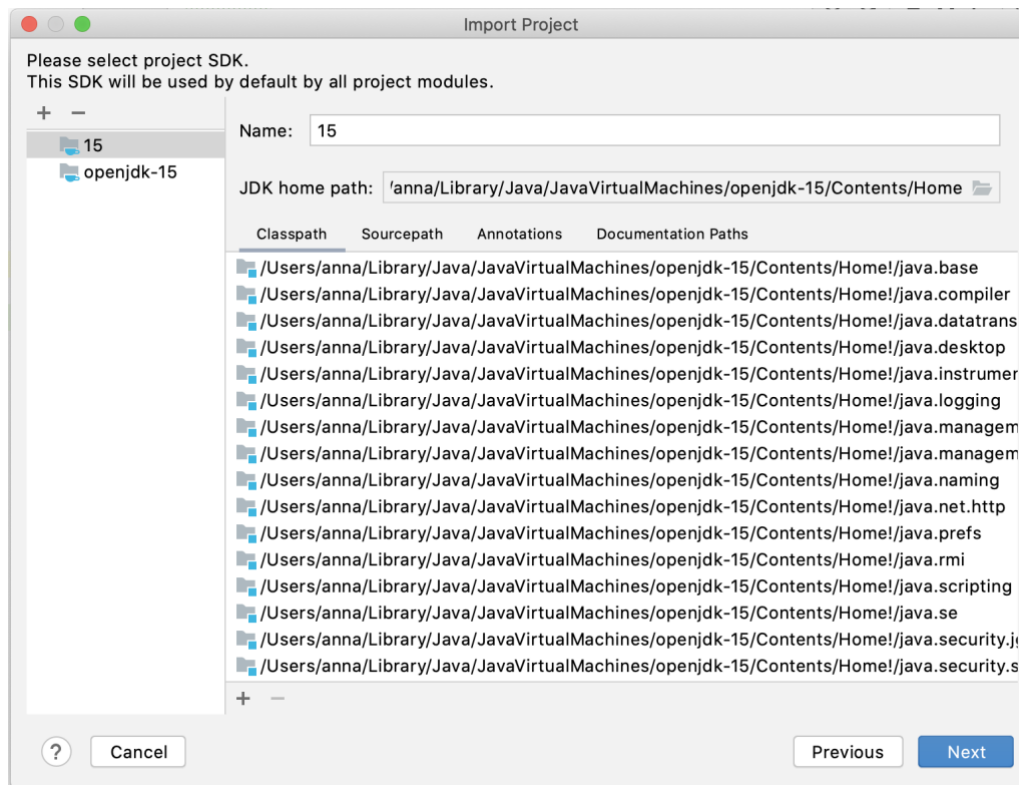
- Seleccionem la opció *Create project from existing sources*



Afegim les dependències al mòdul:

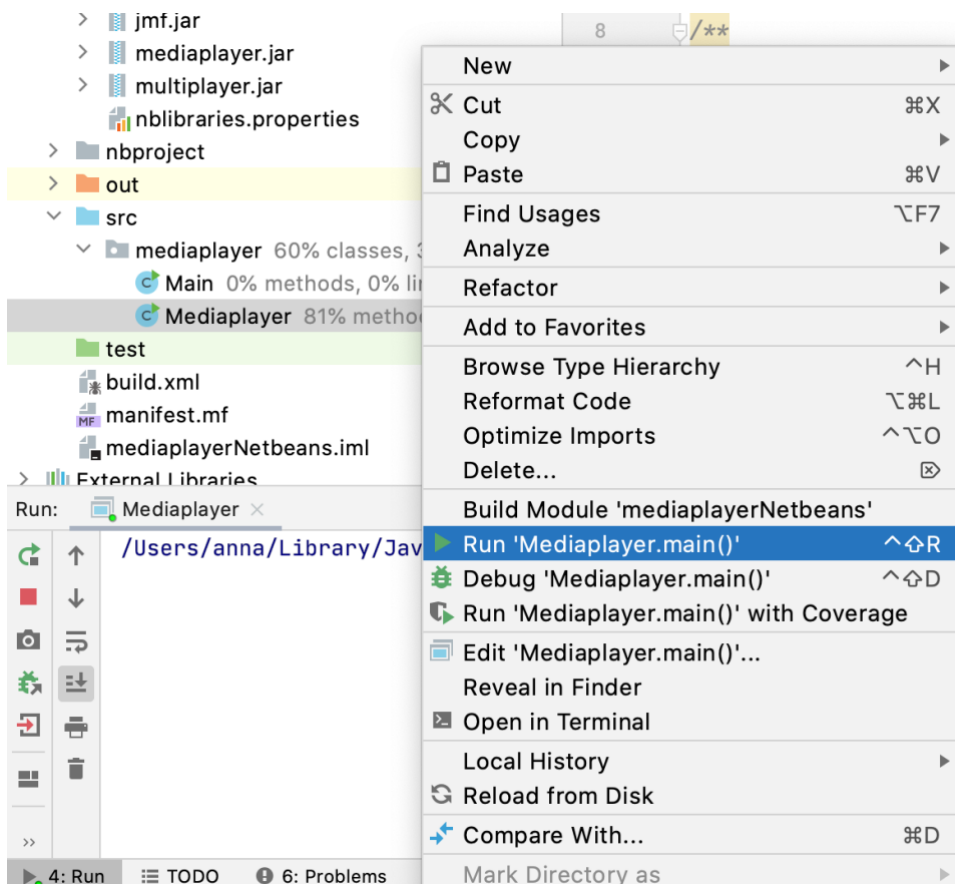


Triem el SDK que volem usar:



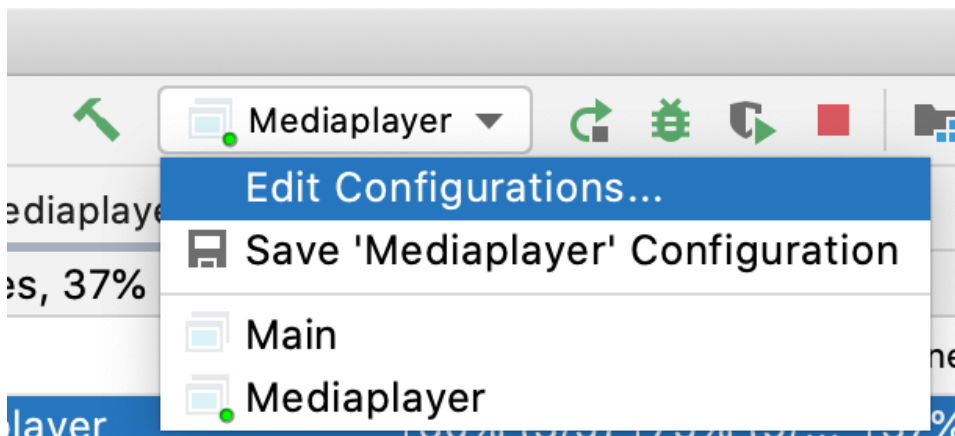
En aquest exemple podem executar directament la classe MediaPlayer per a reproduir música.



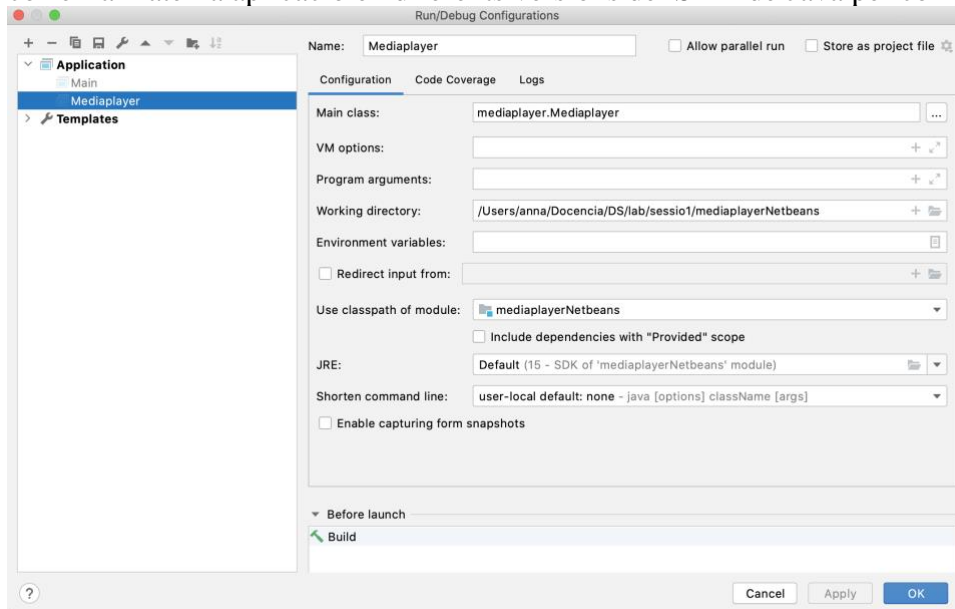


## 5. Executant un projecte

Per defecte a cada mòdul tindrem les classes executables amb Run/Debug per defecte. Si no el tinguéssim o volguéssim modificar-lo hauríem d'entrar al menú de navegació i fer un *Edit Configurations*.

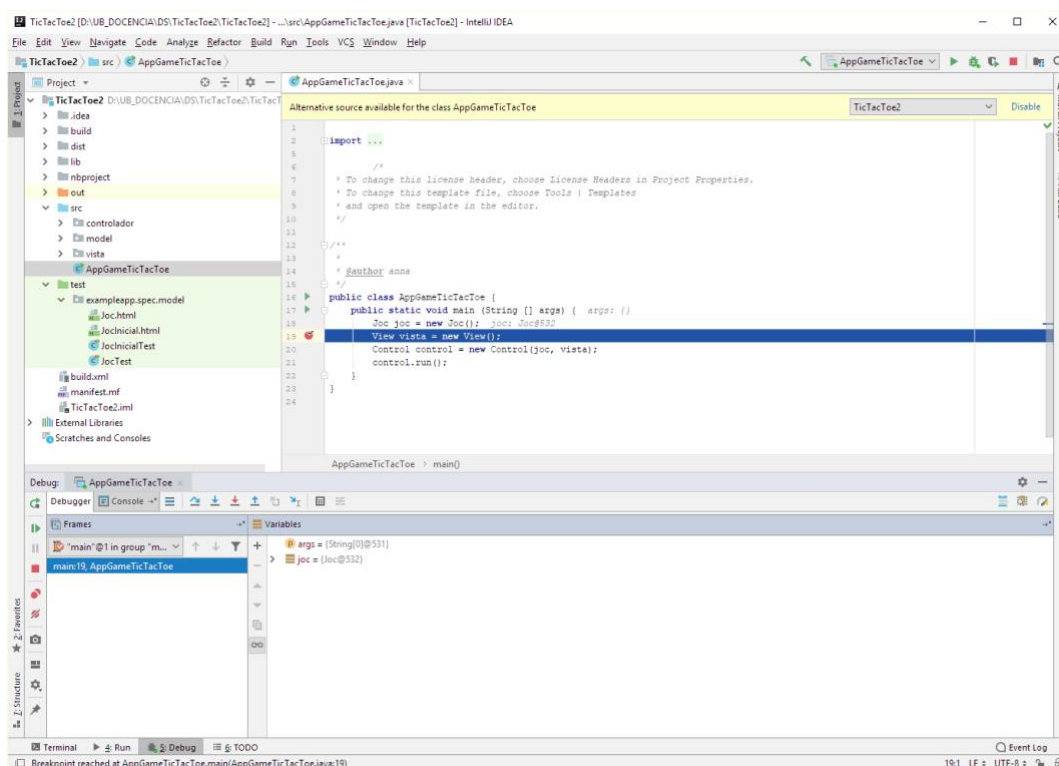


Aquí podrem modificar paràmetres i crear noves configuracions. Un ús possible seria si volguéssim córrer la mateixa aplicació en diferents versions del SDK de Java per comprovar el seu funcionament.



## 6. Debugant el projecte:

Podrem usar breakpoints al nostre codi per a poder tenir un traçabilitat en el mode de debug del IntelliJ



```

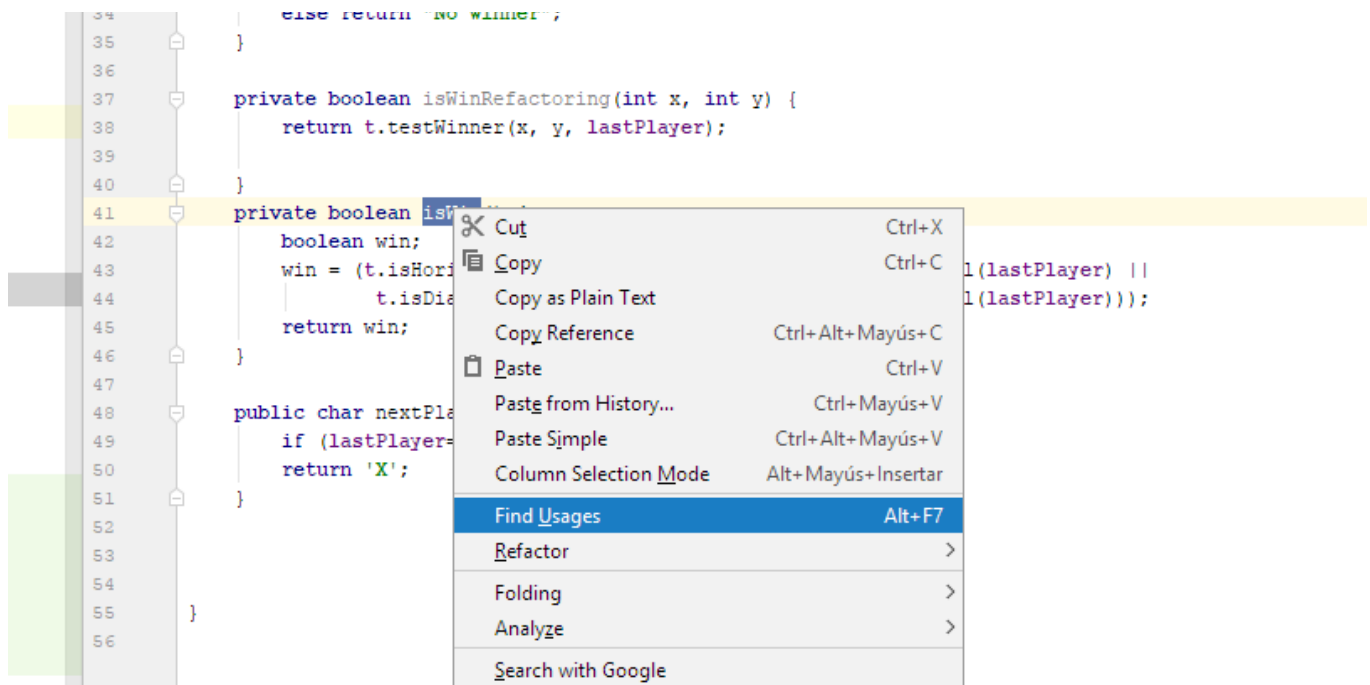
* @author anna
*/
public class AppGameTicTacToe {
    public static void main (String [] args) {    args: {}
        Joc joc = new Joc();    joc: Joc@532
        View vista = new View();
        Control control = new Control(joc, vista);
        control.run();
    }
}

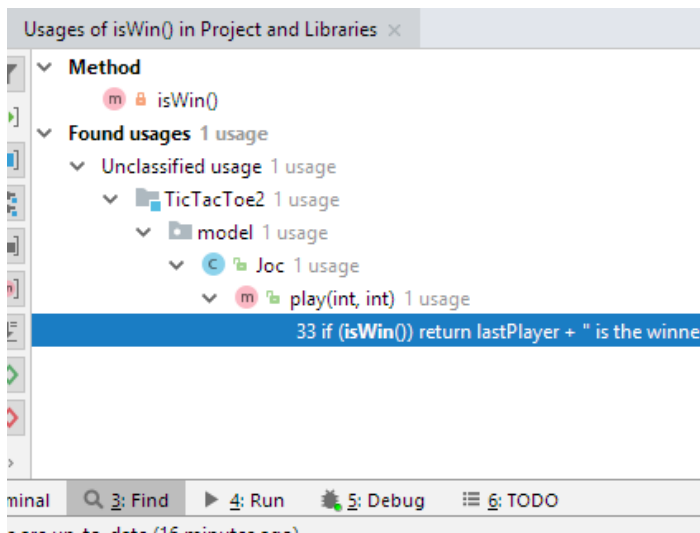
```



## 7. Find Usages

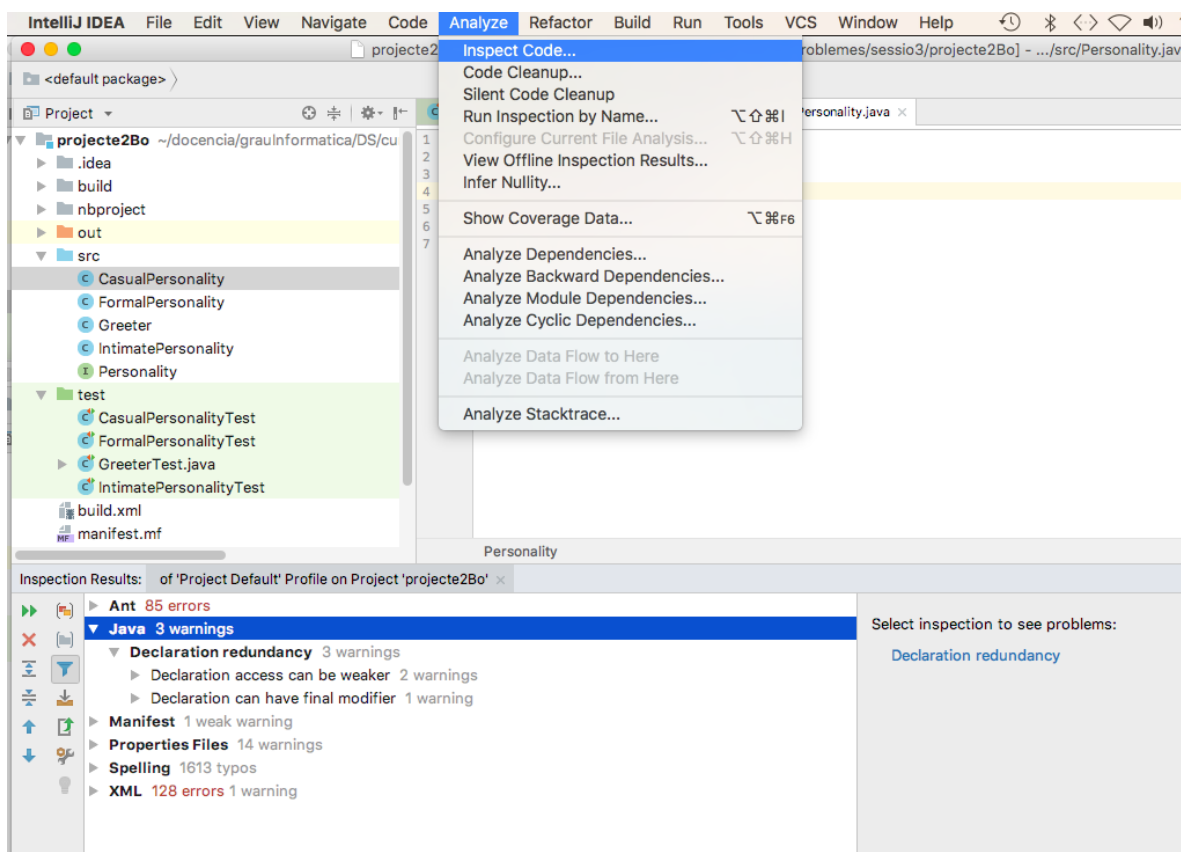
Un funcionalitat molt útil que ens proporciona IntelliJ és la possibilitat de fer un *Find usages* d'un element del nostre codi.





## 8. Inspeccionant el codi:

En aquest IDE també hi ha la possibilitat d'inspeccionar el codi i que ens doni suggeriments de fer codi no redundant o comprova els criteris de públic, privat del projecte. Quan feu *Analyze* > *Inspect code* ... us fa una anàlisi a nivell de classe o de projecte, analitzant les línies redundants (per exemple, construccions alternatives no necessàries, o mètodes que no han de ser públics, així com parts del codi que es poden refactoritzar, etc.)



Trobaràs més informació en el vídeo: <https://youtu.be/iZRdQtTxS9A>

## **9. Referències i més documentació.**

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

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

<https://intellij-support.jetbrains.com/hc/en-us/community/posts/206887325-Difference-between-project-and-module>

<https://www.jetbrains.com/help/idea/2018.1/run-debug-configuration.html>

<https://www.jetbrains.com/help/idea/finding-usages-in-project.html>

Preguntes més freqüents de la importació de projectes de Netbeans a IntelliJ:

<https://www.jetbrains.com/help/idea/netbeans.html#d1440169e98>