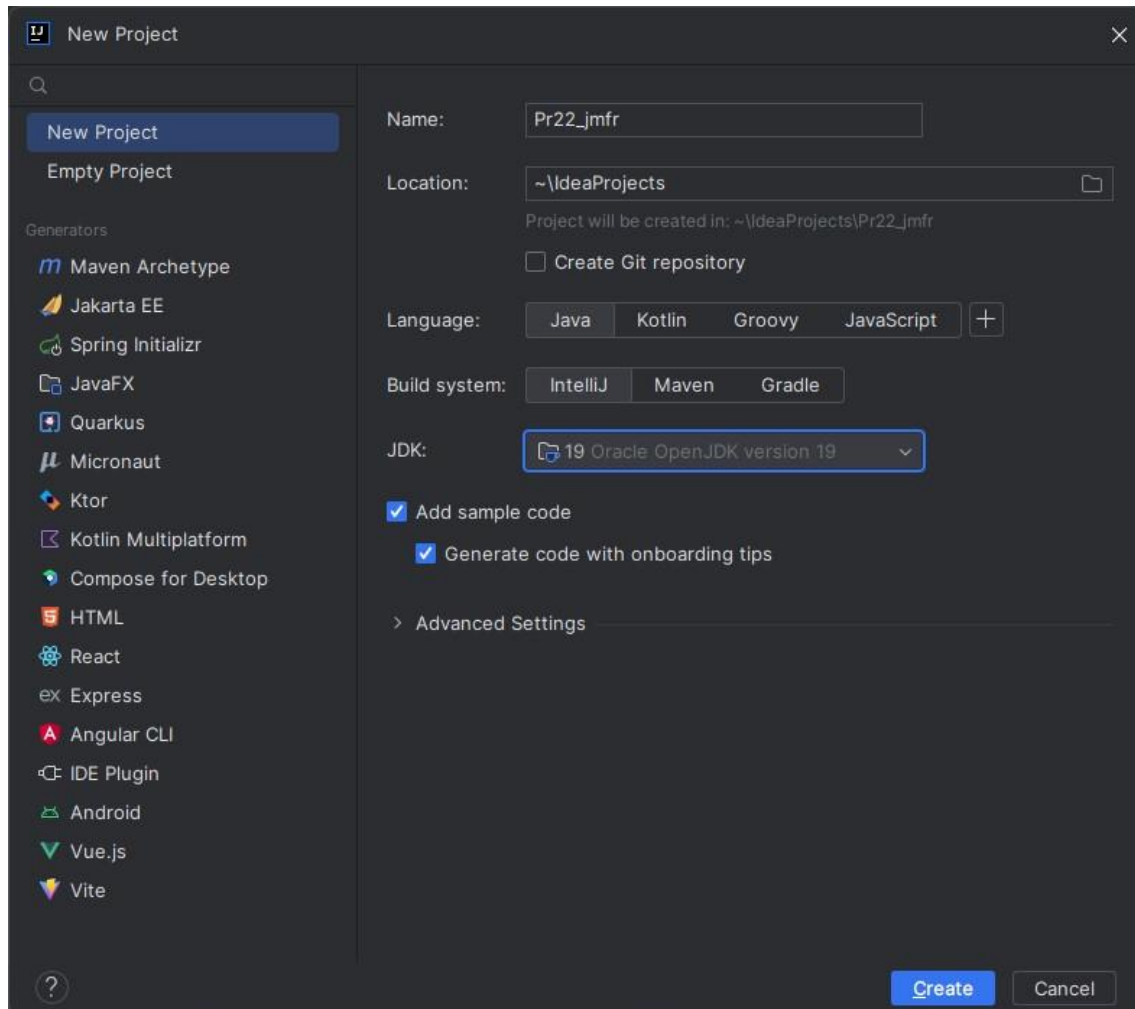
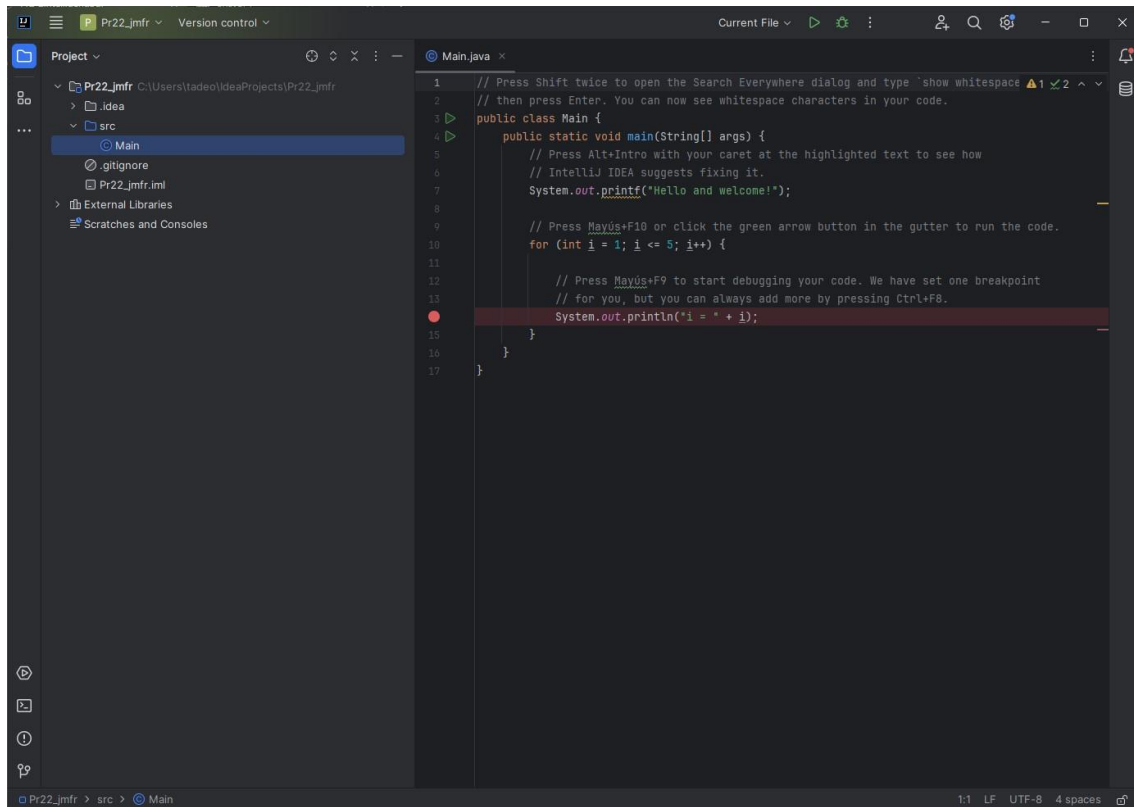


Práctica 2.2 – Manejo básico de IntelliJ IDEA. Instalación de plugins

Ejercicio 1 – Crear un proyecto

Creas un nuevo proyecto en IntelliJ IDEA llamado Pr22 seguido de tus iniciales. Incluye un fichero main con la estructura de la función main.





- ¿Qué atajo de teclado puedes usar para crearlo?

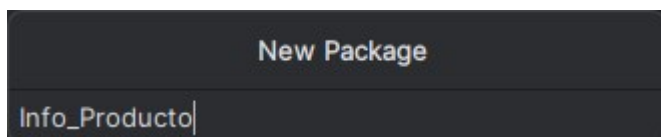
He usado el atajo “Alt” + “Insert”.

- ¿Cómo te puedes mover a la ventana del proyecto mediante el teclado?

Te puedes mover a la ventana con el atajo de teclado “Alt” + “1”.

Ejercicio 2 – Crear un paquete y una clase

- Crea un paquete Info_Producto. ¿Qué teclas usas para crearlo?



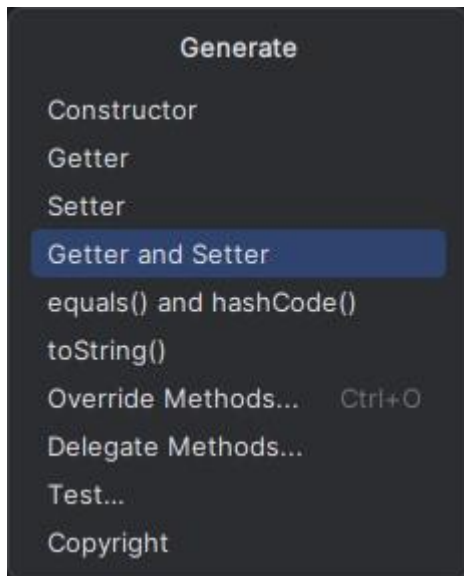
Click derecho encima de la carpeta del proyecto, seleccionamos paquete y crear.

- En el paquete Info_Producto, crea una nueva clase llamada Producto. Incluye los siguientes atributos:

```
private String codigo;  
private String nombre;  
private float precio;
```

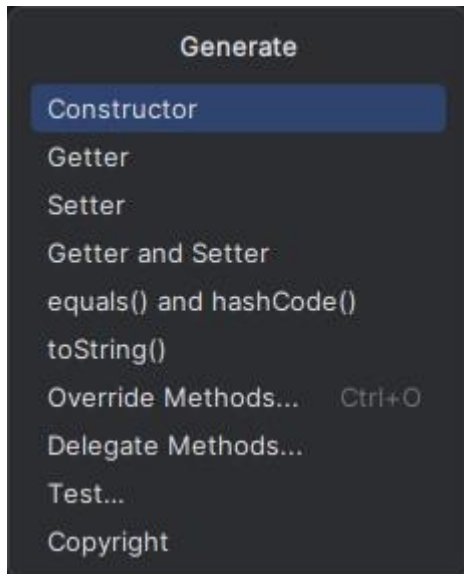
- Mediante atajos de teclado, crea los métodos getters/setters para cada uno de los campos.
¿Cómo accedes a la opción?

Usaremos las teclas “Alt + “Insert”.

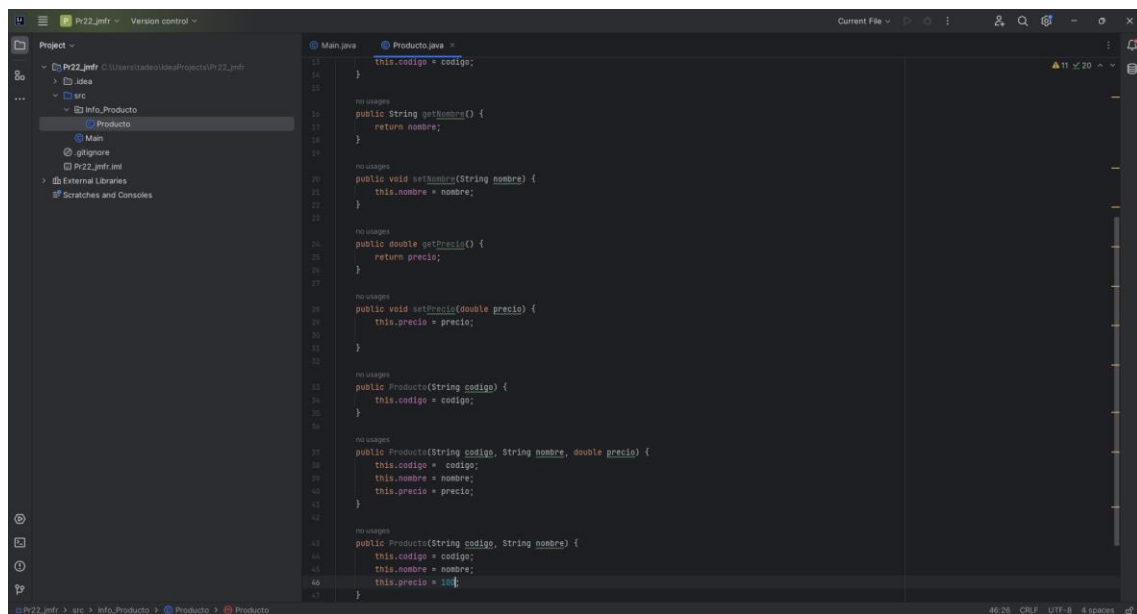


Seleccionaremos este campo

- Crea los siguientes constructores:



- Por defecto (sin parámetros)
- . Establece código como “00”, nombre “producto 0” y precio=0
- Con tres valores
- Con valores para código y nombre. Precio tomará por defecto el valor 10



Así quedaría el código habiendo añadido el constructor

- ¿Qué atajos de teclado usas para crearlos?

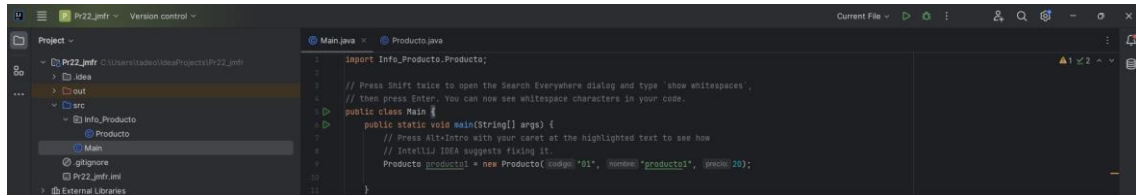
Usaremos en todos el atajo “Alt”+ “Inser”

Ejercicio 3 – Crear un programa de prueba

- Muévete al archivo main.java mediante el teclado. ¿Qué atajo usas?

Usaremos el comando “Ctrl” + “Tab”.

- Inserta la siguiente línea de código: `Producto producto1 = new Producto("01","producto1",20);`



- Intenta compilar usando el teclado. ¿Qué atajo usas?

Con la tecla “Ctrl” + “F9”. • Incluye una captura de pantalla con los errores que obtienes

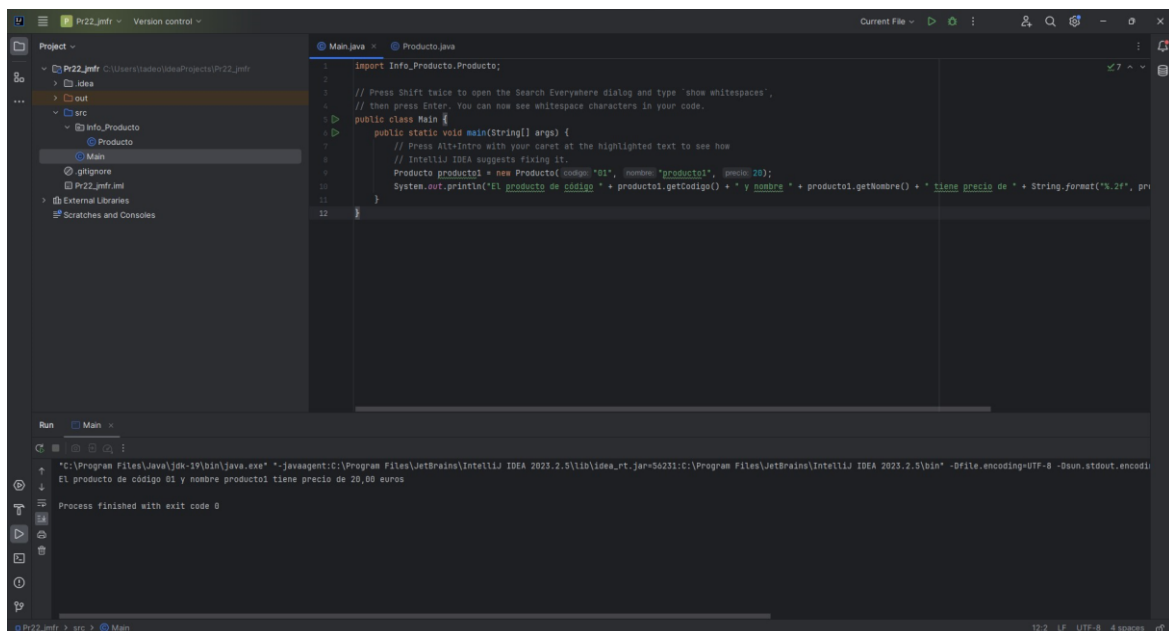
A la hora de compilar en mi caso no me ha dado erro

- Soluciona los errores usando solo el teclado. Indica qué teclas o comandos has usado y describe con tus palabras cuál era el problema para no compilar

- Inserta una línea de código para que muestre el siguiente resultado:

El producto de código 01 y nombre producto1 tiene precio 20.0 euros

- Compila y ejecuta el programa. Indica los atajos de teclado que has usado

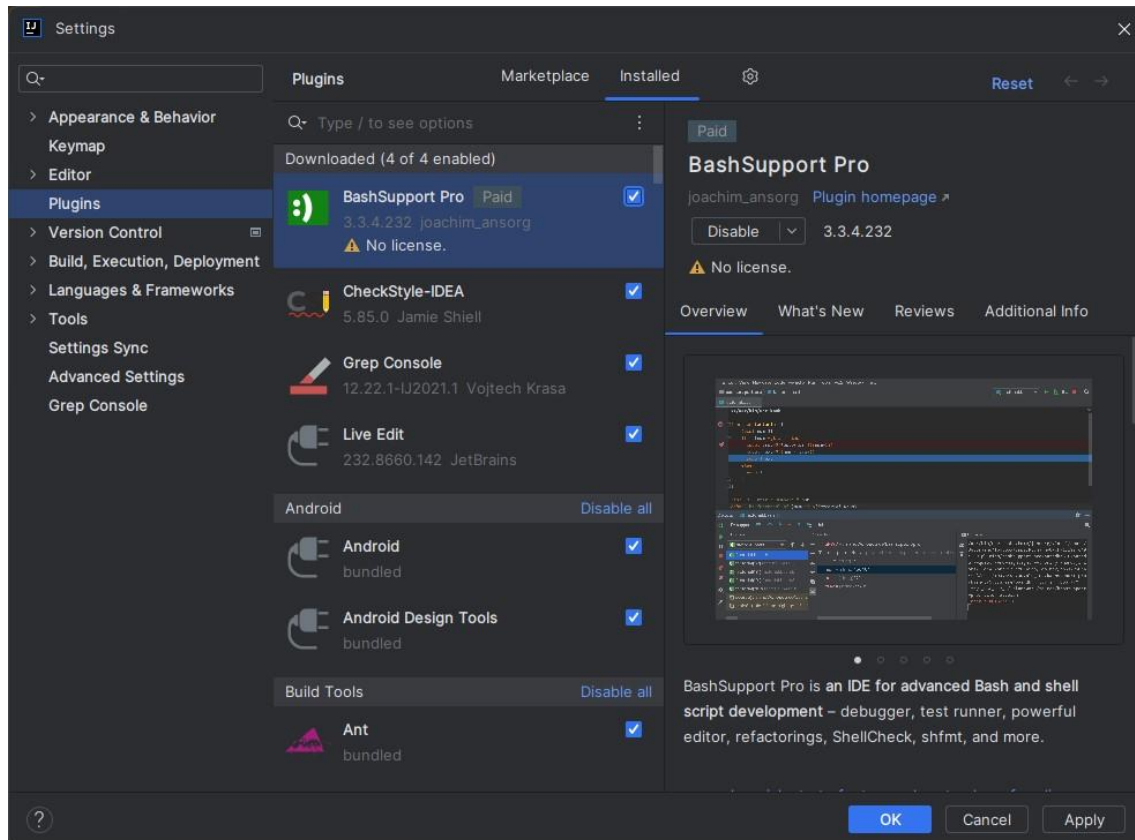


Lo haremos con la tecla “Shift” + “F10”.

Ejercicio 4 – Instalar plugins.

Instala los siguientes plugins en IntelliJ. Busca información de cada uno de ellos y describe qué hacen brevemente. Muestra ejemplos con capturas de pantalla

- BashSupport
- LiveEdit
- Grep console
- Check-Style



BashSupport:

Este plugin ofrece soporte para el lenguaje de scripting Bash en IntelliJ IDEA. Proporciona resaltado de sintaxis, autocompletado, navegación, y otras características para mejorar el desarrollo de scripts Bash.

LiveEdit:

LiveEdit es una característica de IntelliJ IDEA que permite la edición en vivo de código HTML, CSS y JavaScript. Se actualiza automáticamente en el navegador web sin necesidad de recargar la página.

Grep Console:

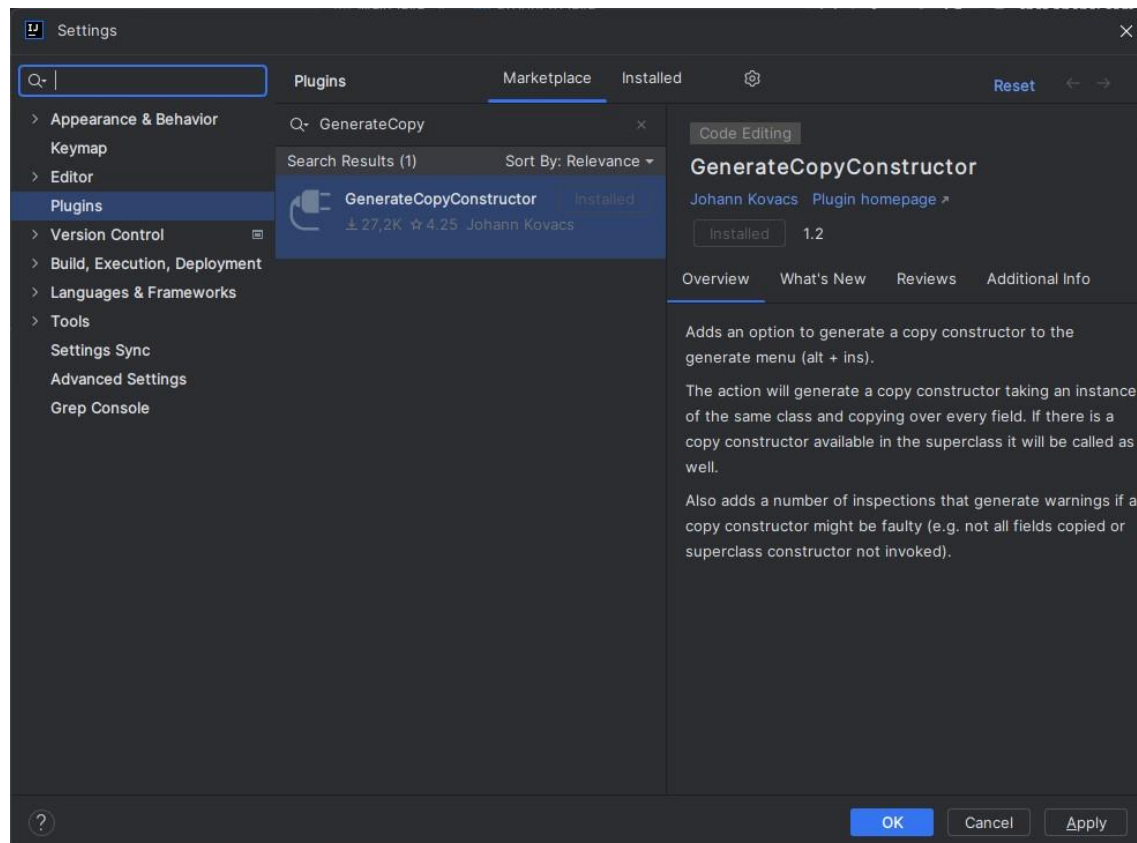
Este plugin agrega una consola especializada en la que puedes filtrar y resaltar líneas de texto en función de patrones de expresiones regulares (regex). Es útil para analizar registros o resultados de la consola.

Checkstyle:

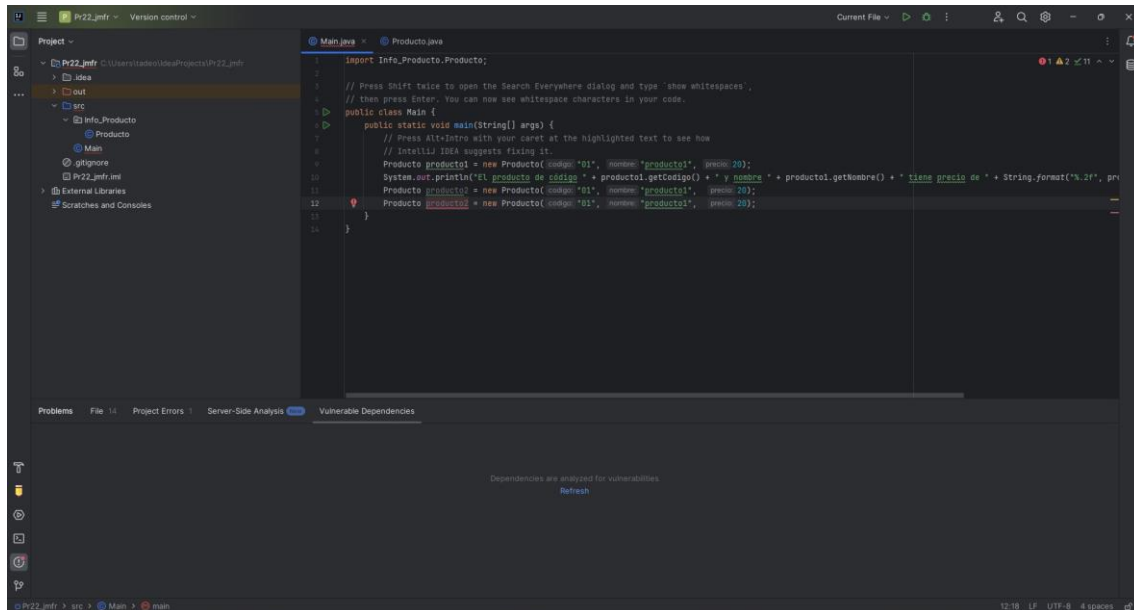
Checkstyle es una herramienta de desarrollo que ayuda a garantizar que el código Java cumple con un conjunto específico de reglas de codificación. Este plugin integra Checkstyle en IntelliJ IDEA para analizar y mostrar problemas de estilo en el código.

Ejercicio 5. Constructor copia

- Abre la ventana de plugins e instala GenerateCopyConstructor. Este plugin te permitirá crear constructores copia, que generan un objeto con los mismos datos que el que se pasa como parámetro.



- En el programa principal, debajo de la instrucción `system.out.println` crea un objeto `producto2` con los mismos datos que `producto1`.

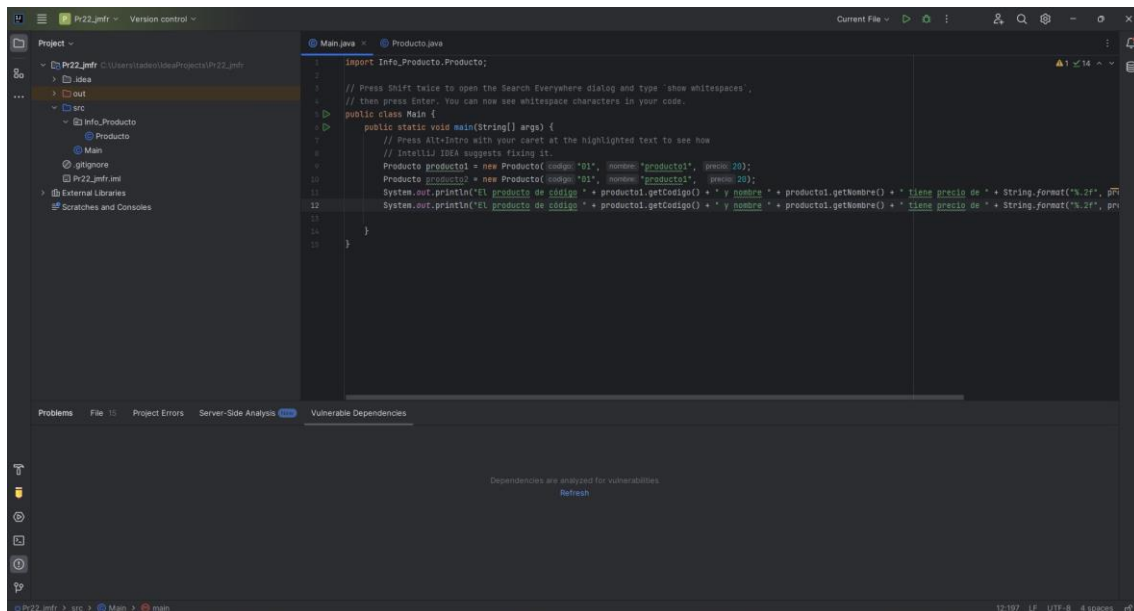


- Sitúate en la línea `system.out.`. Pulsa la tecla `shift+alt+down` (`shift+alt+up`). ¿Qué hace? Sitúa la línea detrás de la creación del objeto `producto2`

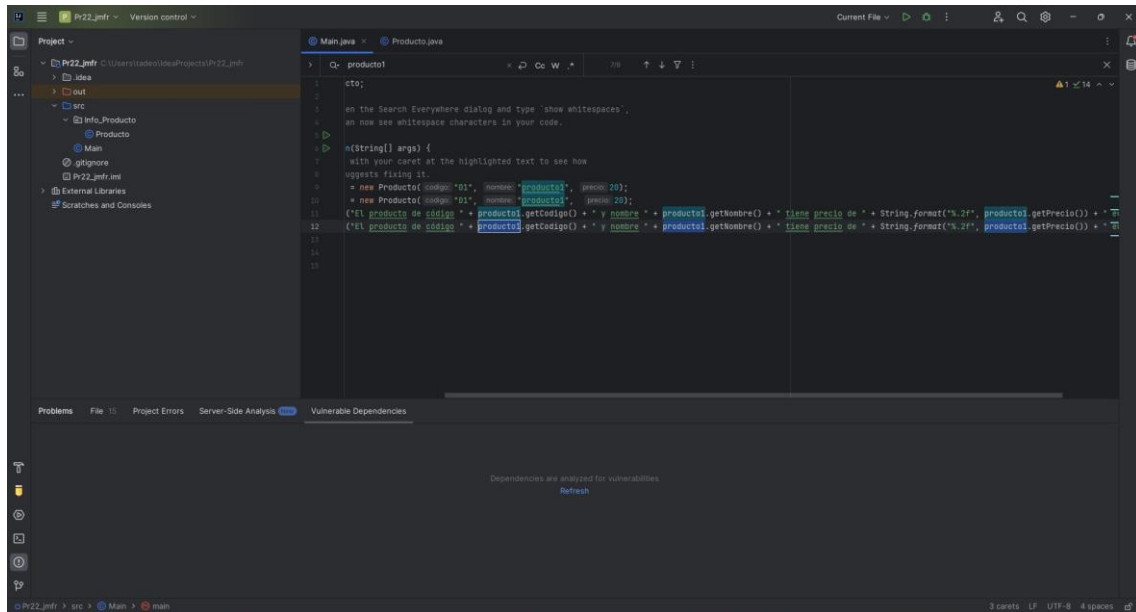
Mueve esa línea a la línea de arriba o a la de abajo, depende que flecha pulses.

- Duplica la línea `system.out.`. ¿Qué tecla has usado?

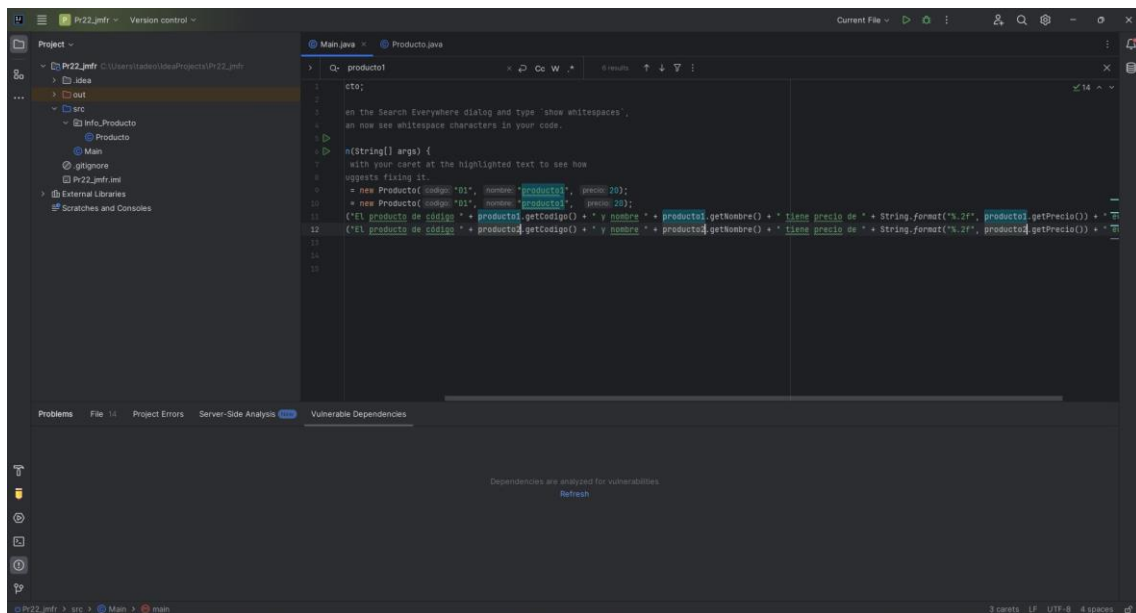
La tecla `Ctrl + D`.



- En la segunda línea system.out, selecciona el texto producto1 usando la tecla F3, pulsa ALT+J dos veces para seleccionar las dos veces que se llama a producto1 en dicha línea de código. Una vez que estén las tres ocurrencias seleccionadas, cambia producto1 por producto2.



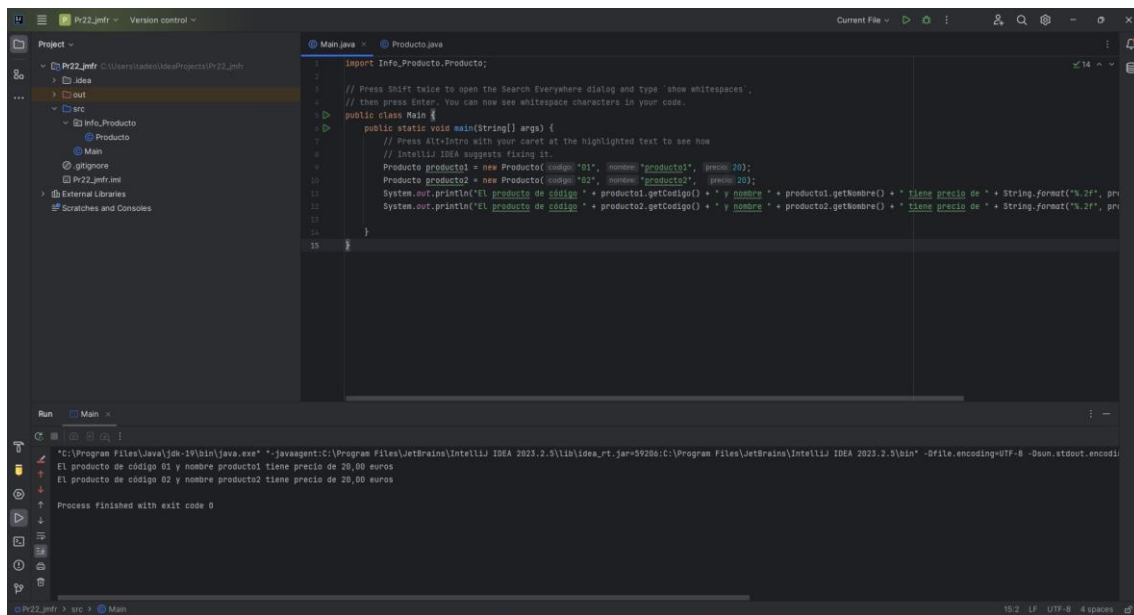
```
1  cto;
2
3  on the Search Everywhere dialog and type 'show whitespaces',
4  an now see whitespace characters in your code.
5
6  n(String[] args) {
7      with your caret at the highlighted text to see how
8      suggests fixing it.
9      = new Producto( "01", nombre: "producto1", precio: 20);
10     = new Producto( "01", nombre: "producto1", precio: 20);
11     ("El producto de código " + producto1.getCodigo() + " y nombre " + producto1.getNombre() + " tiene precio de " + String.format("%.2f", producto1.getPrecio()) + " €")
12     ("El producto de código " + producto1.getCodigo() + " y nombre " + producto1.getNombre() + " tiene precio de " + String.format("%.2f", producto1.getPrecio()) + " €")
13
14
15
```



```
1  cto;
2
3  on the Search Everywhere dialog and type 'show whitespaces',
4  an now see whitespace characters in your code.
5
6  n(String[] args) {
7      with your caret at the highlighted text to see how
8      suggests fixing it.
9      = new Producto( "01", nombre: "producto1", precio: 20);
10     = new Producto( "01", nombre: "producto1", precio: 20);
11     ("El producto de código " + producto1.getCodigo() + " y nombre " + producto1.getNombre() + " tiene precio de " + String.format("%.2f", producto1.getPrecio()) + " €")
12     ("El producto de código " + producto1.getCodigo() + " y nombre " + producto1.getNombre() + " tiene precio de " + String.format("%.2f", producto1.getPrecio()) + " €")
13
14
15
```

Cambiamos el texto producto1, y le cambiamos por producto2.

- Compila el programa y comprueba que los dos objetos que has creado son iguales.



- Genera el código para crear un tercer objeto y mostrar sus datos por pantalla. En este caso se le pedirán los datos al usuario (código, nombre y precio)

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Producto producto1 = new Producto("01", "producto1", 20);
```

```
        Producto producto2 = new Producto(producto1);
```

```
        System.out.println("Producto 1: " + producto1);
```

```
        System.out.println("Producto 2 (copia de Producto 1): " + producto2);
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Ingresa el código del producto:");
```

```
        String codigo = scanner.nextLine();
```

```
        System.out.println("Ingresa el nombre del producto:");
```

```
        String nombre = scanner.nextLine();
```

```
        System.out.println("Ingresa el precio del producto:");
```

```
        double precio = scanner.nextDouble();
```

```
        Producto producto3 = new Producto(codigo, nombre, precio);
```

```
        System.out.println("Producto 3 (creado por el usuario): " + producto3);
```

```
    }
```

```
}
```

Ejercicio 6 – Instalar librerías externas

- Crea un proyecto nuevo llamado pr22-math seguido de tus iniciales con un fichero con el programa main

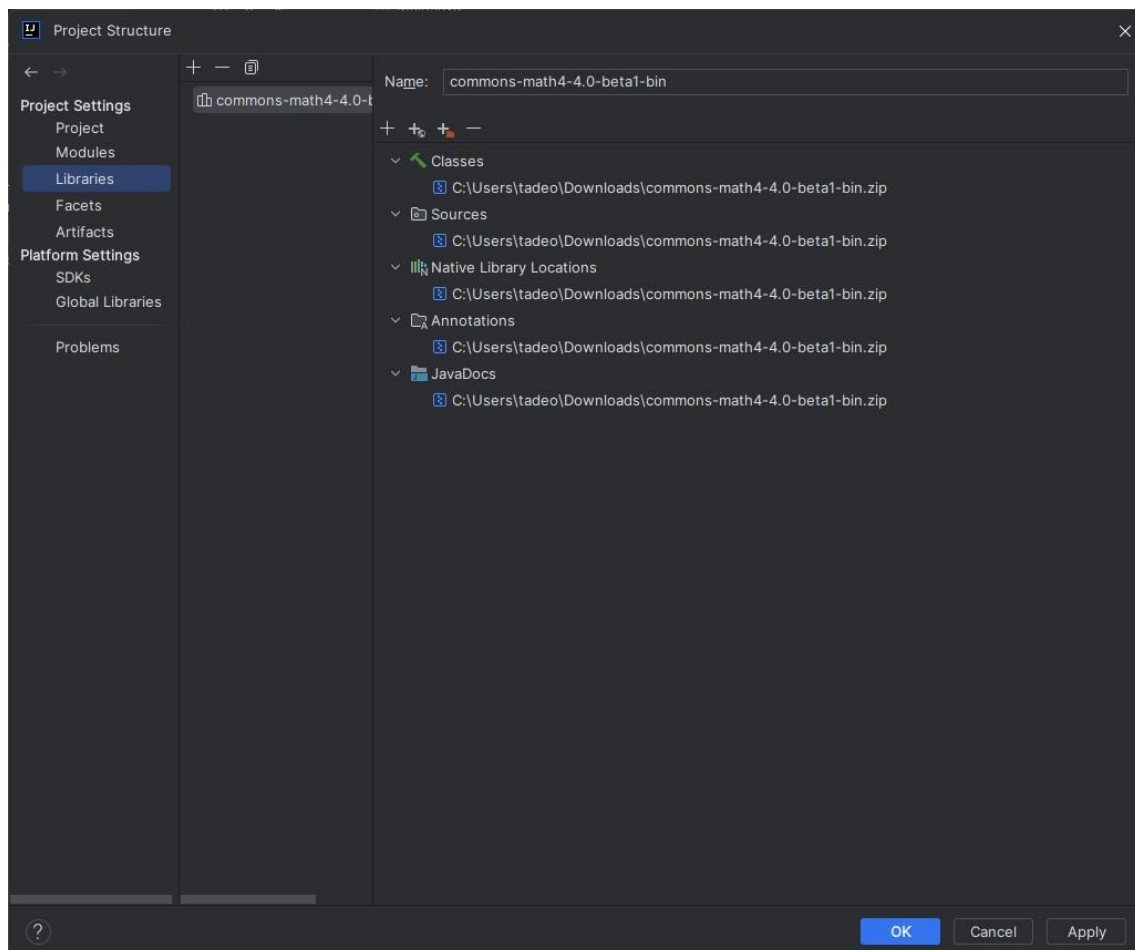
Añadimos el proyecto como hicimos en el primer ejercicio.

- Descarga la librería commons-math de la página oficial.

Descargamos la que es el .zip

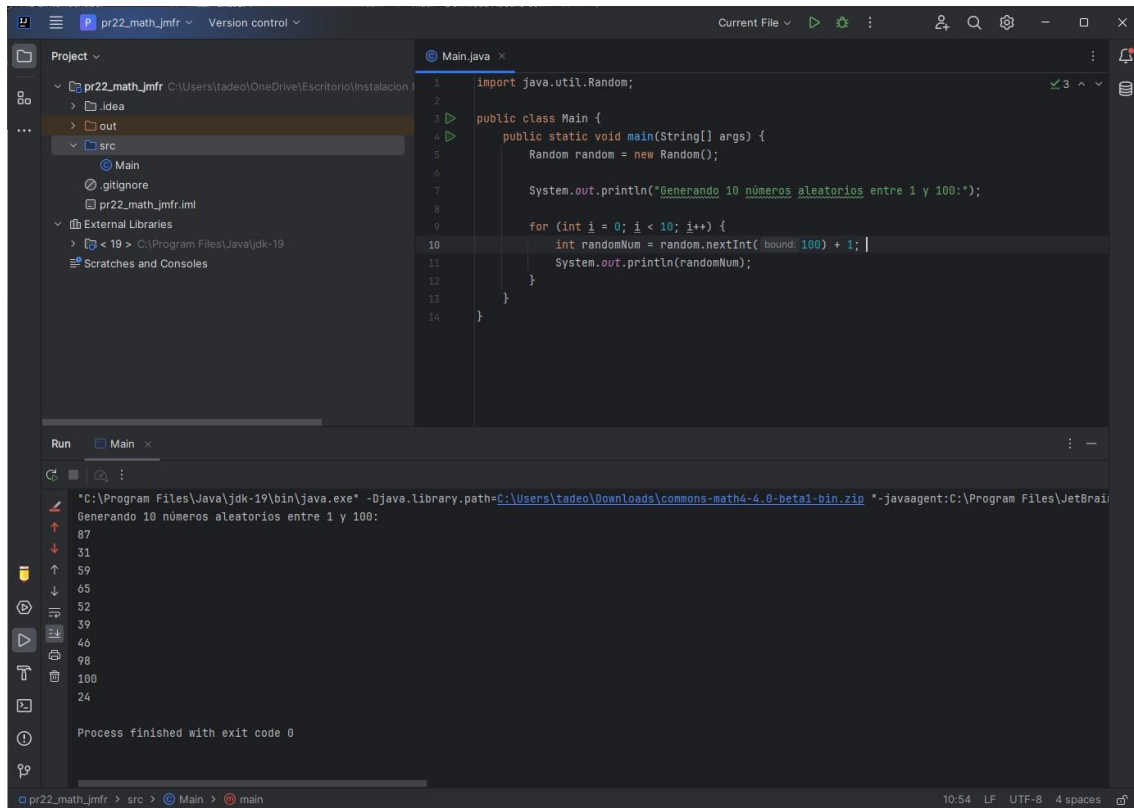
[commons-math4-4.0-beta1-bin.zip](#)

- Incluye la librería en el proyecto (fichero jar, sources y documentación). Crea una captura de pantalla de dicha ventana. ¿Con qué atajo puedes acceder a la configuración del proyecto?



Con el comando Ctrl + Alt + Shift + S, podemos entrar directamente al apartado “Project Settings”

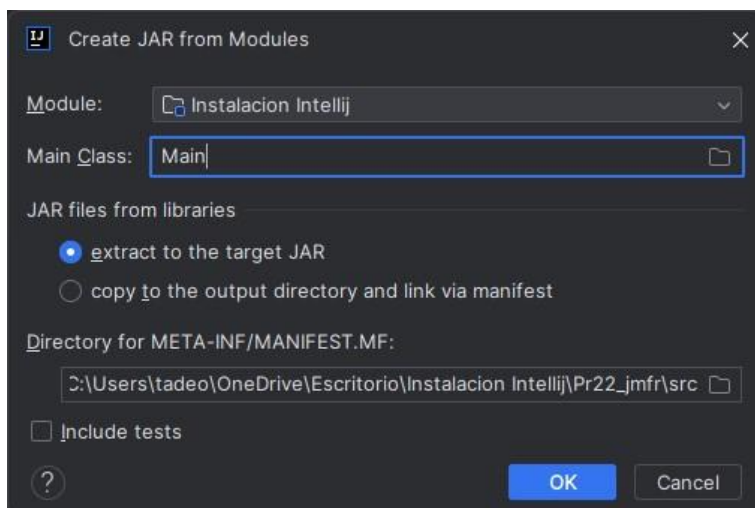
- En el programa principal, crea el código que permita generar y mostrar 10 números aleatorios (random en inglés) entre el valor 1 y 100. Genera una captura de pantalla con la ejecución del programa.



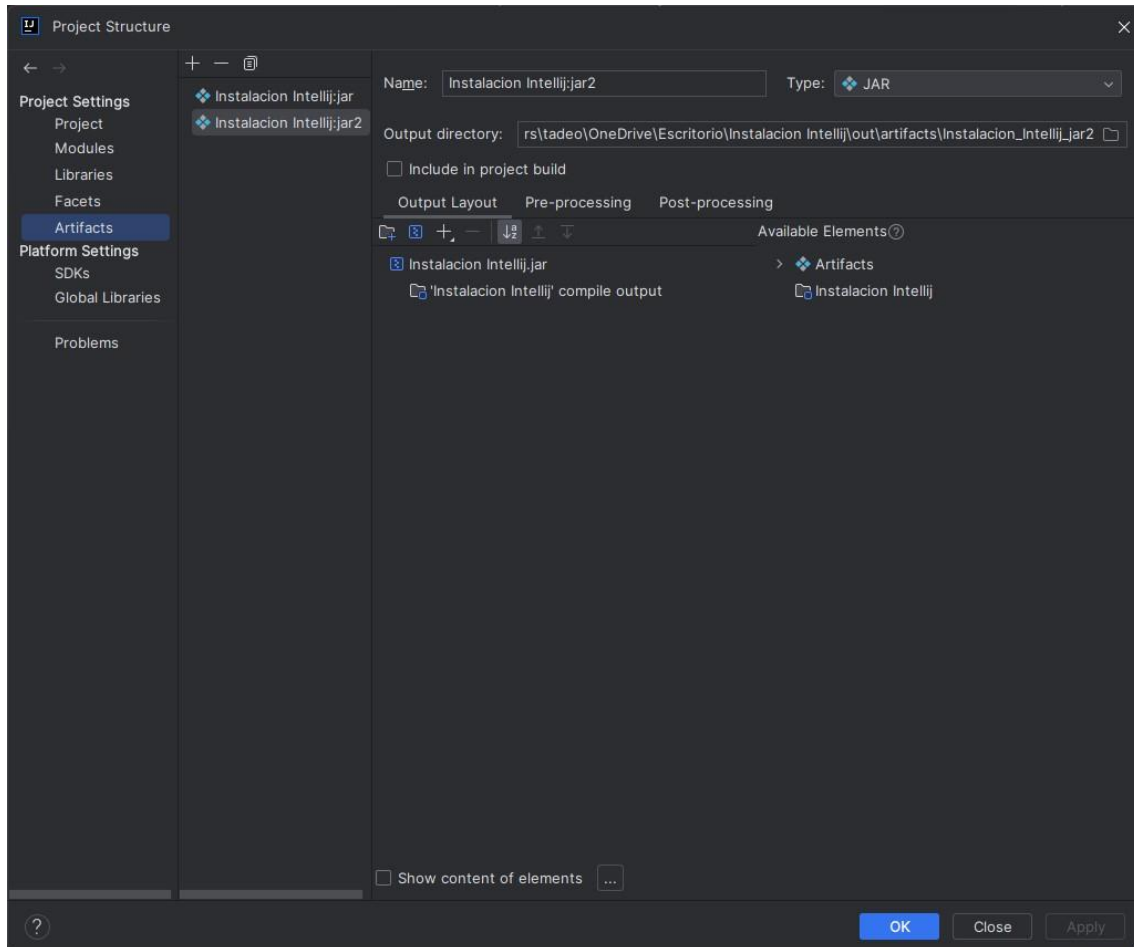
Ejercicio 7 – Generar fichero jar

- Genera un fichero jar para cada uno de los dos proyectos que has creado en la práctica. ¿Cómo accedes a la opción de generación? Indica los atajos que usas

Entramos en Project Settings, con el comando que vimos anteriormente Ctrl + Alt + Shift + S, seleccionamos la ventana "Artifacts", dentro le damos clic en el botón "+" y seleccionamos "JAR" "From modules with dependencies.



Seleccionaremos los dos proyectos.



Cuando los hayamos añadido nos saldrá algo así.