
IDE ☺

ÍNDICE

Instalación

Creación de archivos

Funcionalidades avanzadas (Si las hubiera)

Creación de proyectos

Herramientas de debugging

IDE seleccionado: IntelliJ IDEA

¿Qué entendemos por IDE?

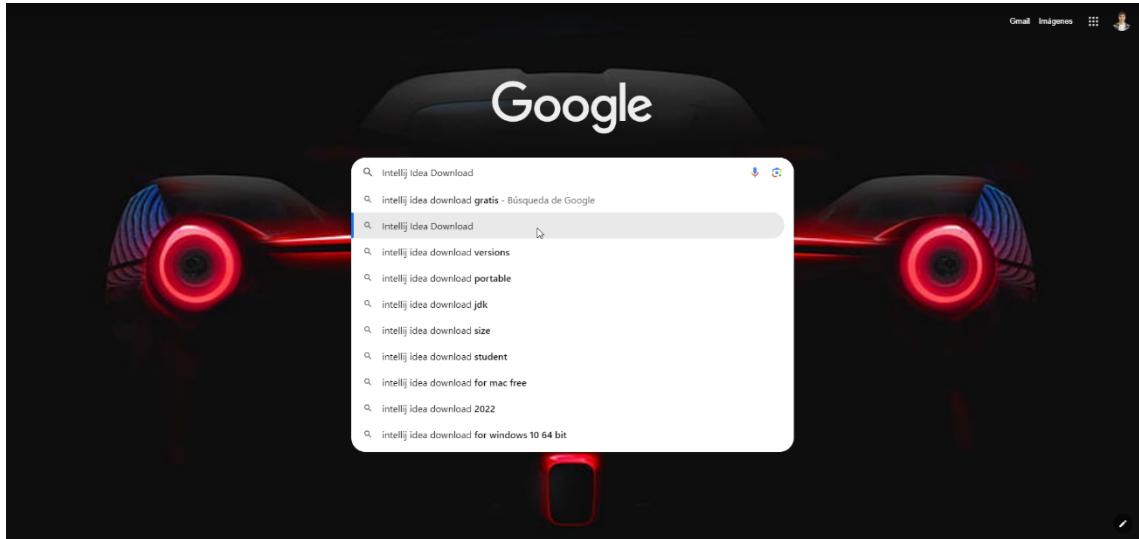
Aplicación de software que **centraliza** todas las **herramientas esenciales**

(editor, traductor de código y el detector de errores) para que los programadores

puedan escribir, probar y construir **software** de forma eficiente.

1) Instalación

Vamos al motor de búsqueda

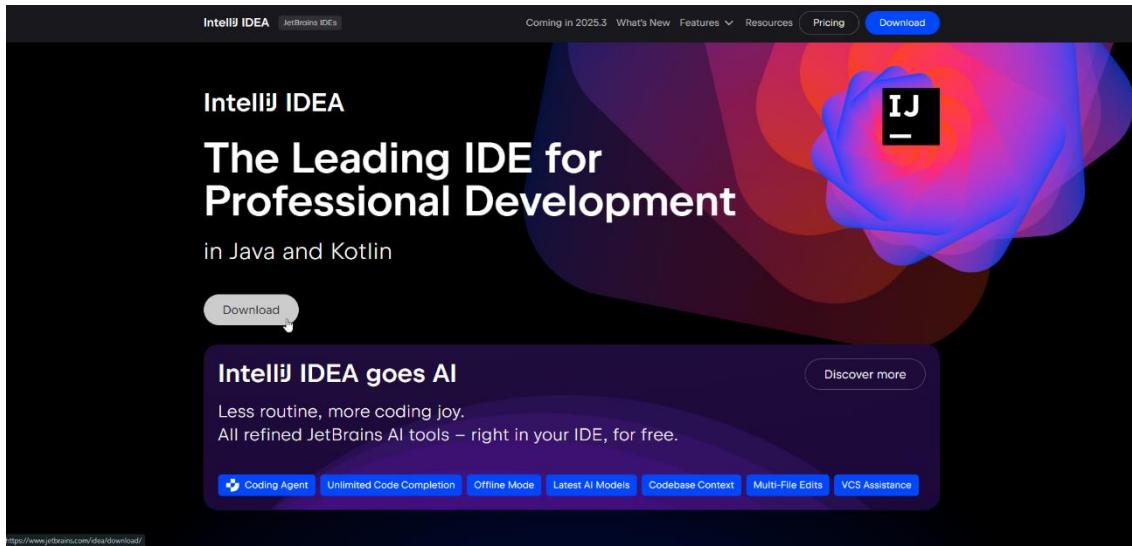


Accedemos al siguiente enlace

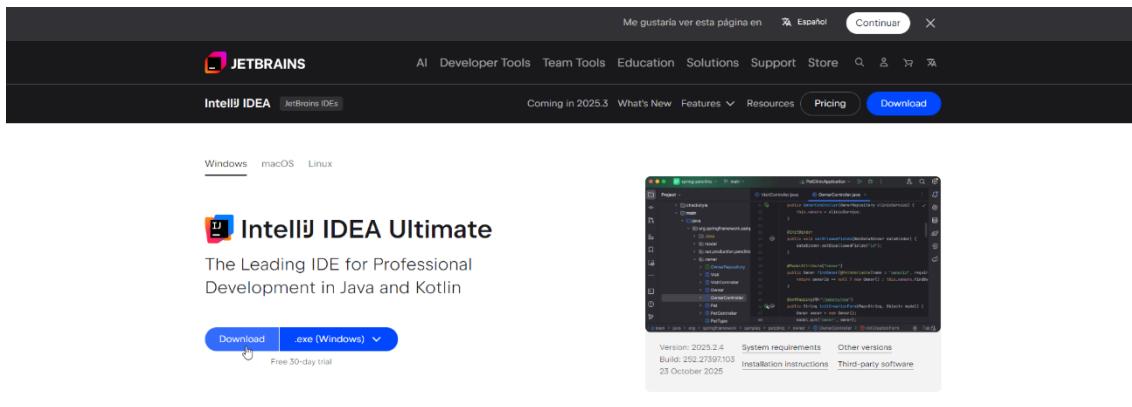
A screenshot of a web browser displaying the JetBrains website for IntelliJ IDEA. The page has a dark background. At the top left is the JetBrains logo, followed by the text "JetBrains" and the URL "https://www.jetbrains.com/idea". To the right of the URL is a purple link "Traducir esta página" and a three-dot menu icon. Below this, there is a large heading "IntelliJ IDEA - in Java and Kotlin" with a hand cursor icon pointing at the "IDEA" part. Underneath the heading, there is a brief description: "IntelliJ IDEA helps you write code faster with tools that eliminate tedious tasks and let you focus on what matters – building great software." The overall layout is clean and professional.

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

Clicamos en *Download*



Último clic en *Download*



Lo encontramos en local */descargas*

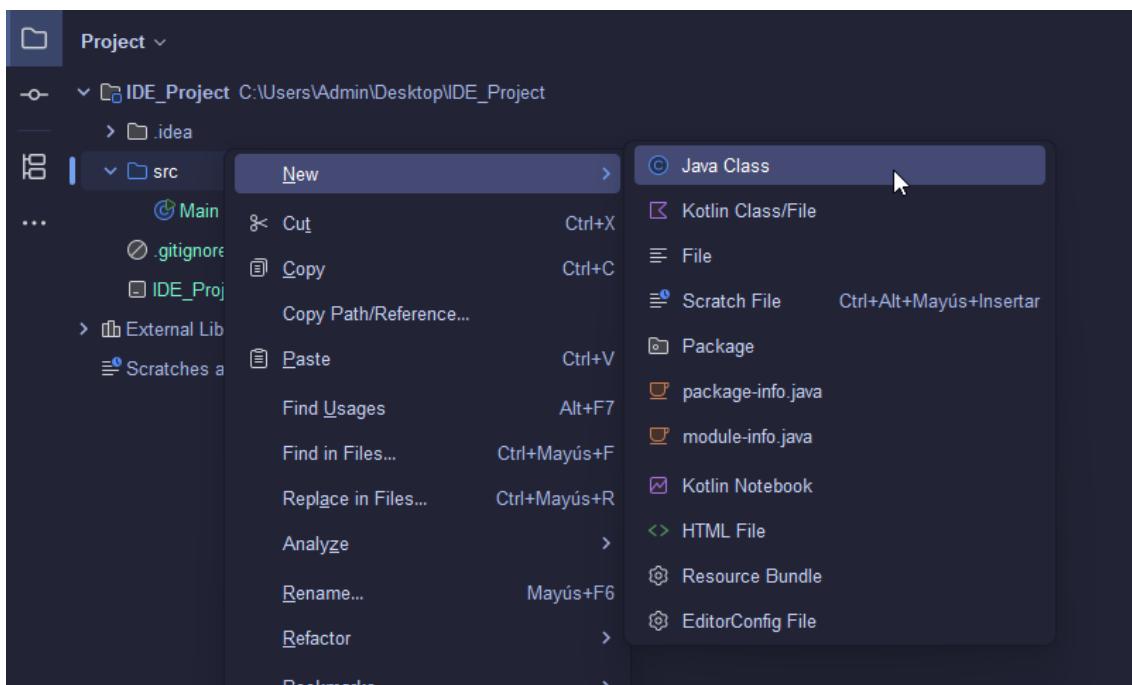


2) Creación de archivos

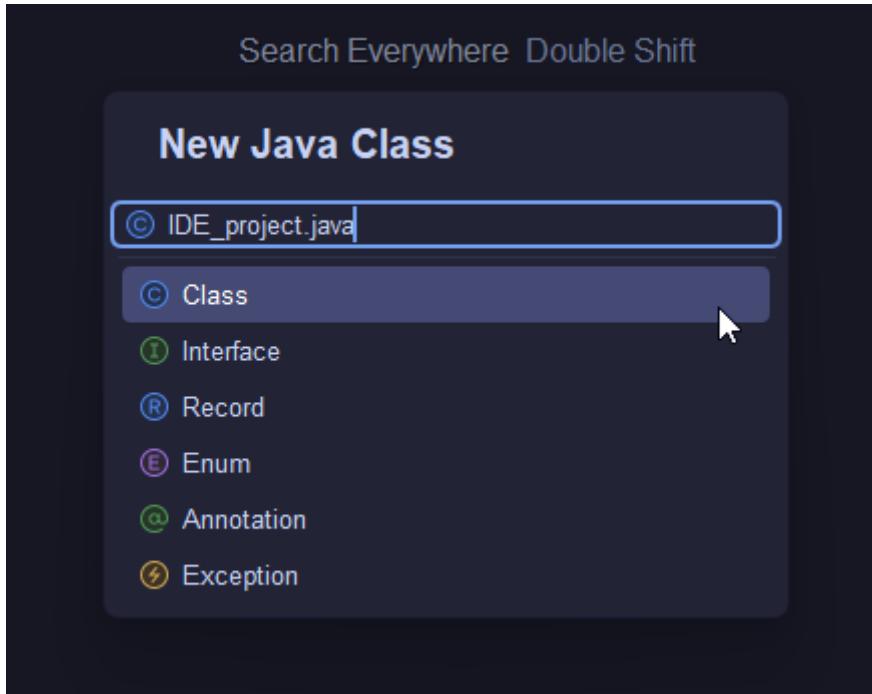
Voy a generar un proyecto nuevo para ver de forma clara esa creación de archivos.

Archivo .java

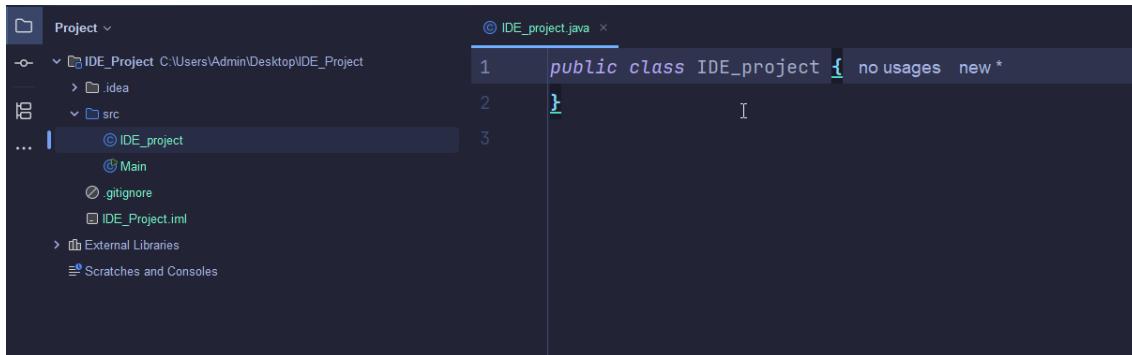
- src (clic derecho) > New > Java class



Asignamos un nombre a nuestro archivo `.java`



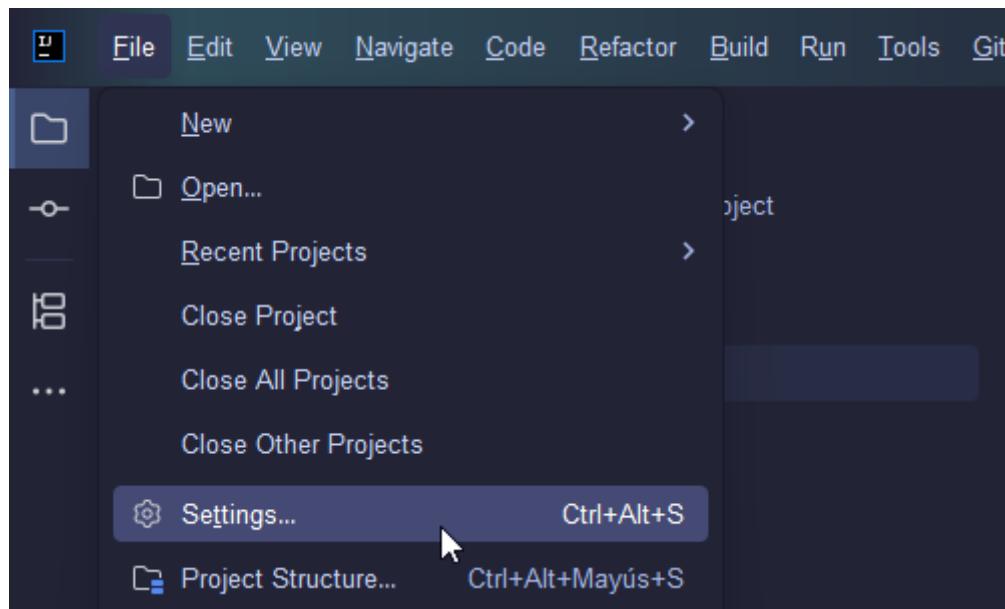
¡Archivo `.java` creado!



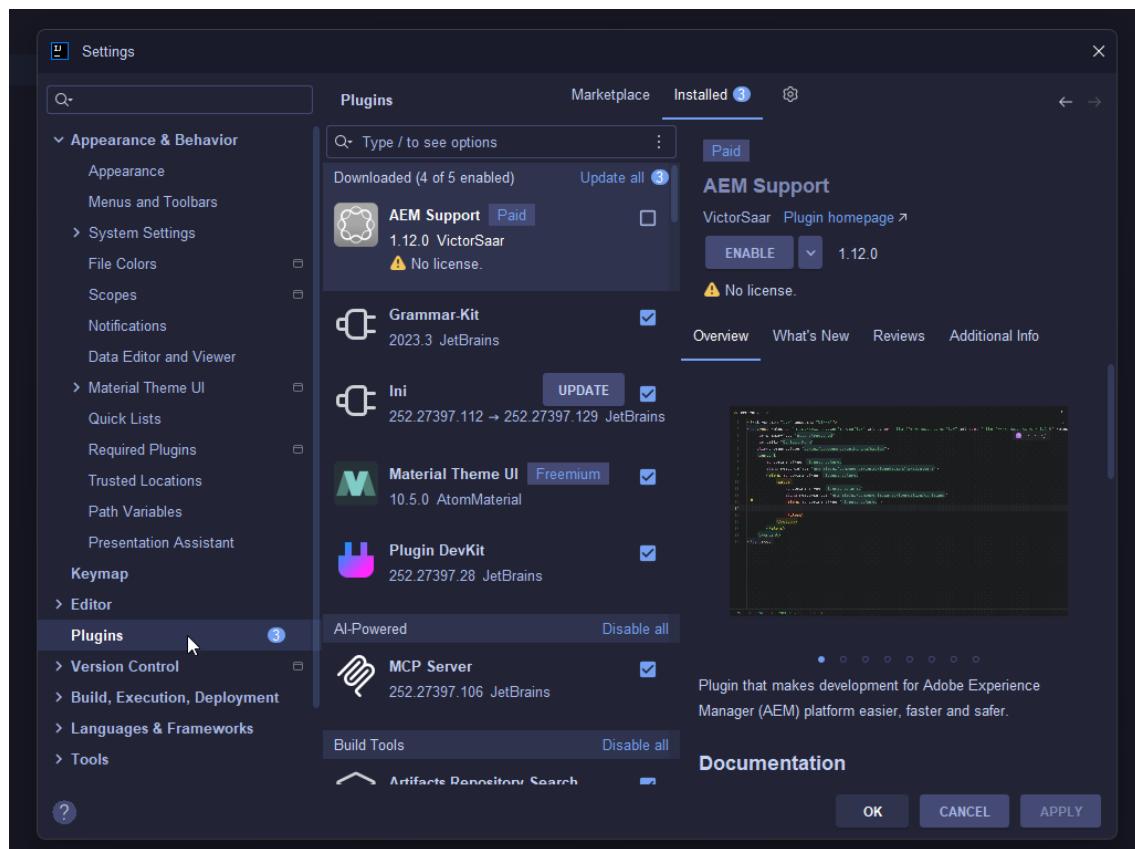
3) Funcionalidades avanzadas (pluggins)

Instalaremos por pasos una funcionalidad o plugin de fondo de nuestro IDE

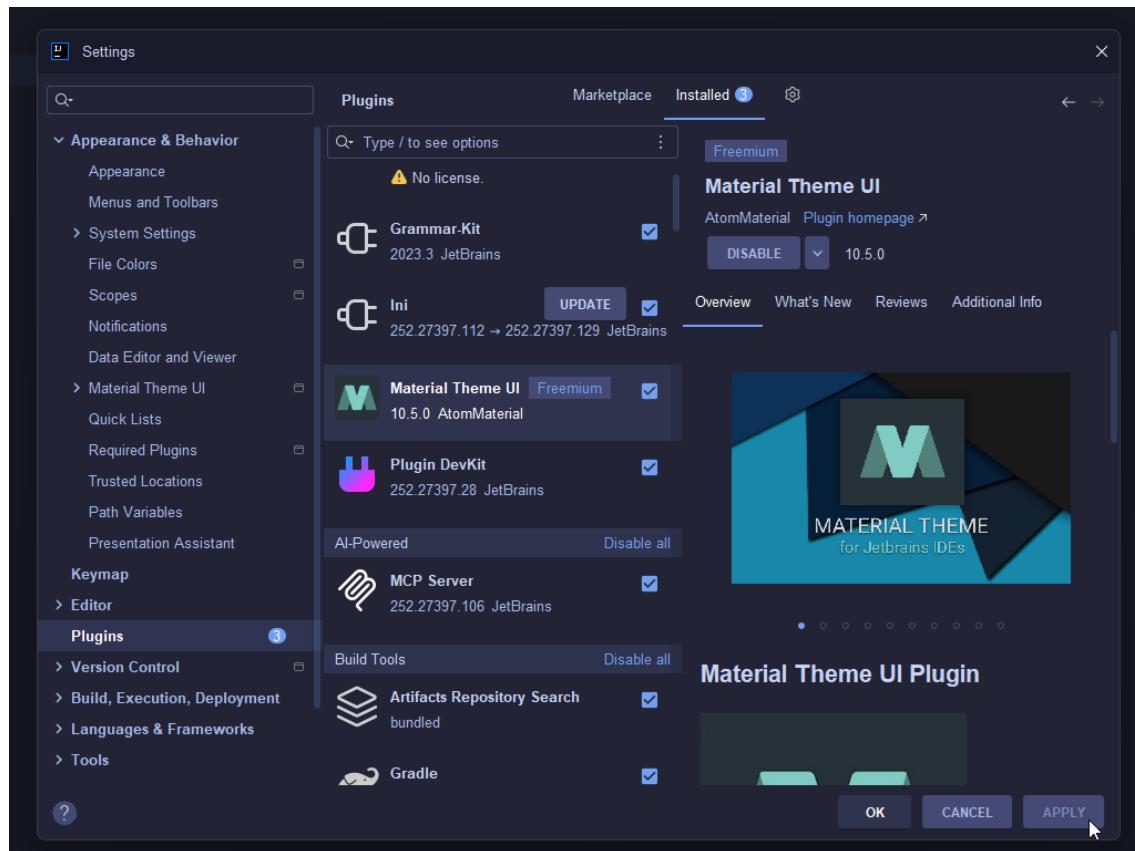
File > Settings



Plugins > Material Theme UI



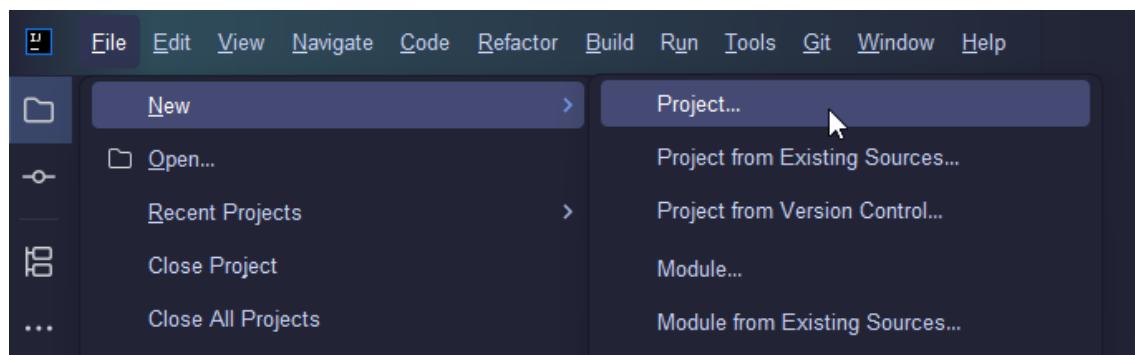
Apply



4) Creación de proyectos

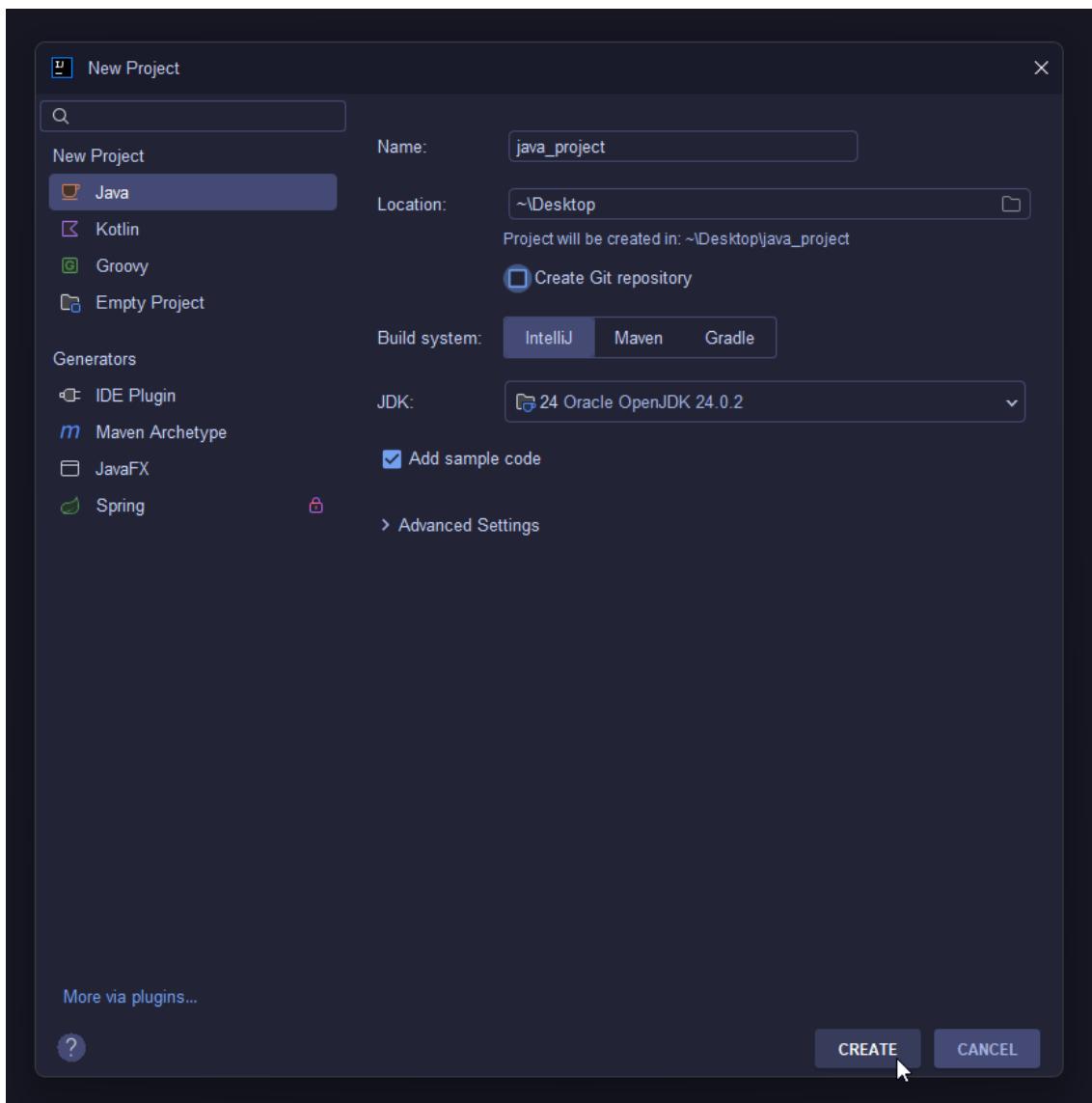
Volvemos a nuestro IDE de los dos puntos anteriores y generamos un proyecto. Será Java project

File > New > Project

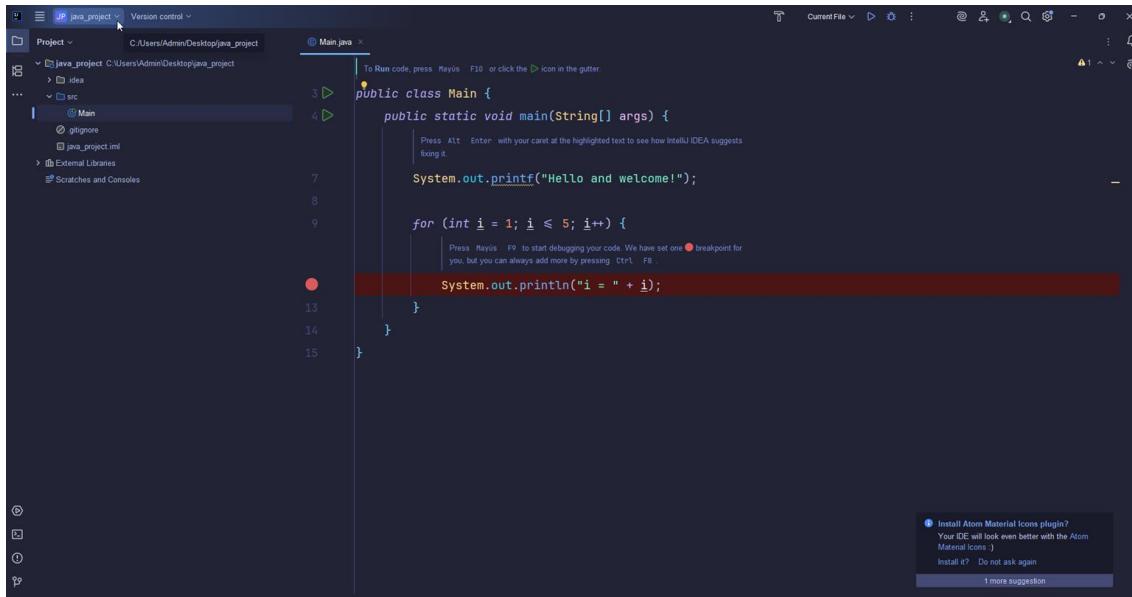


Desarrollamos un nombre y guardamos el proyecto en nuestra local.

Create



¡Proyecto Java_project creado!



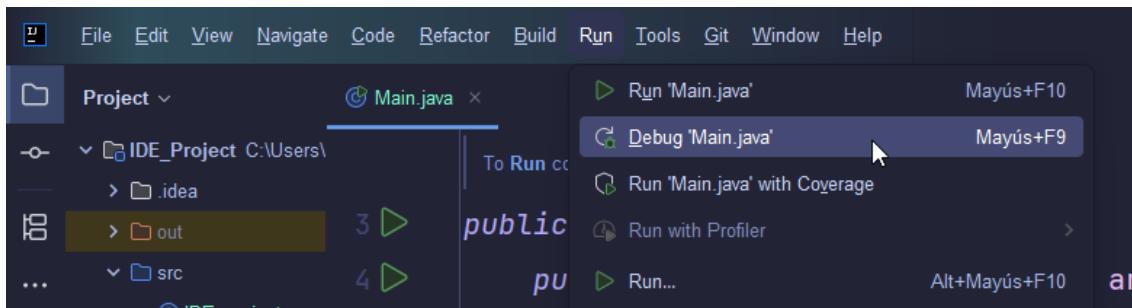
```
public class Main {
    public static void main(String[] args) {
        System.out.printf("Hello and welcome!");
        for (int i = 1; i <= 5; i++) {
            System.out.println("i = " + i);
        }
    }
}
```

5) Herramientas de debugging

Vamos a hacer que nuestro programa funcione.

Antes nos posicionamos sobre el archivo que queramos arrancar. Comprobaremos tambien que el código está bien estructurado. y va a funcionar.

Run > Debug 'nombreArchivo'



En la parte inferior, vemos que la terminal muestra correctamente el código por pantalla.

The screenshot shows the IntelliJ IDEA interface with a Java project named "IDE_Project". The "Main.java" file is open, displaying the following code:

```
public class Main { new*
    public static void main(String[] args) { new*
        System.out.printf("Hello and welcome!");

        for (int i = 1; i <= 5; i++) {
            System.out.println("i = " + i);
        }
    }
}
```

The code editor has syntax highlighting and code completion suggestions. Below the editor is the "Debug" tool window, which shows the output of the running application:

```
Hello and welcome!
i = 1
i = 2
i = 3
i = 4
i = 5
```

The status bar at the bottom indicates "Process finished with exit code 0".