

# android studio



Iván Díaz Miragaya  
Manuel Pérez Menéndez  
Cristian Rodriguez Prado  
Hernan Dario Coronel

# Indice

- 1. Android studio como IDE**
- 2. Guía de instalación en Linux**
- 3. Instalación en windows y recomendaciones**
- 4. Interfaz del IDE**
- 5. Importación y exportación de configuraciones**
- 6. Crear un nuevo proyecto en Java**
- 7. Creación de paquetes y clases**
- 8. Ejecución y depuración**
- 9. Integración con la IA y plugins**
- 10. Integración con GIT**

# Android Studio como IDE

## 1. Resumen del IDE

Android Studio es el IDE oficial para el desarrollo de aplicaciones Android. Está basado en IntelliJ y ofrece herramientas especializadas para el diseño, desarrollo, prueba y depuración de aplicaciones para Android./ Fue lanzado por Google en 2013 reemplazando a Eclipse como el IDE oficial para el desarrollo de aplicaciones android y es una de las plataformas más completas para desarrolladores móviles./

Como características principales cuenta con:

- Finalización de código inteligente, /sugerencias basadas en patrones, plantillas de código../
- Vista previa en tiempo real para diferentes dispositivos y /tamaños de pantalla./
- Android Emulator, /simulación de dispositivos con pruebas de sensores, llamadas, y ubicaciones./
- Cuenta con una consola de desarrollador: /consejos de optimización, ayuda para la traducción, estadísticas de uso./

# Instalación en Linux

## 1 - Requisitos y dependencias

Antes de hacer nada debemos abrir una consola e instalar los siguientes paquetes, tal y como se aprecia en la imagen (necesitaremos permisos de administrador)

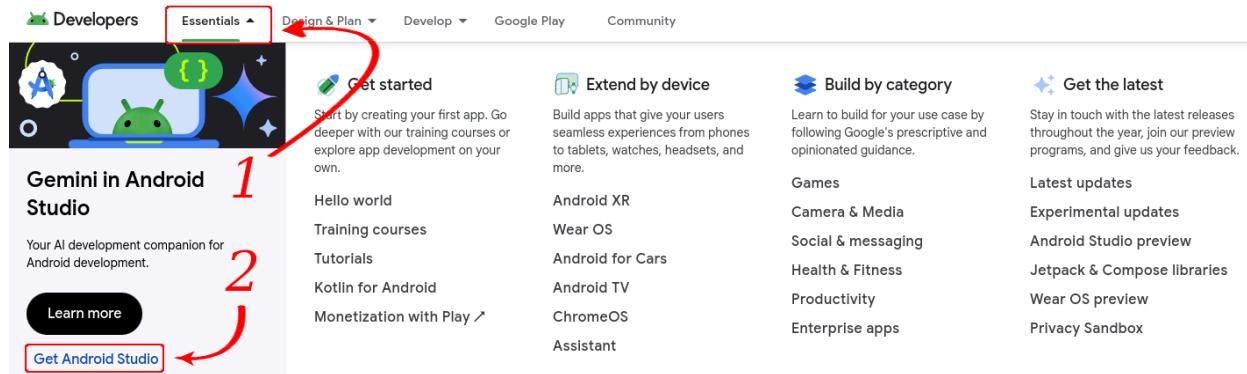
```
$ sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

Tras pulsar 'enter' nos dirá qué paquetes se han instalado y cuales estaban ya en su última versión

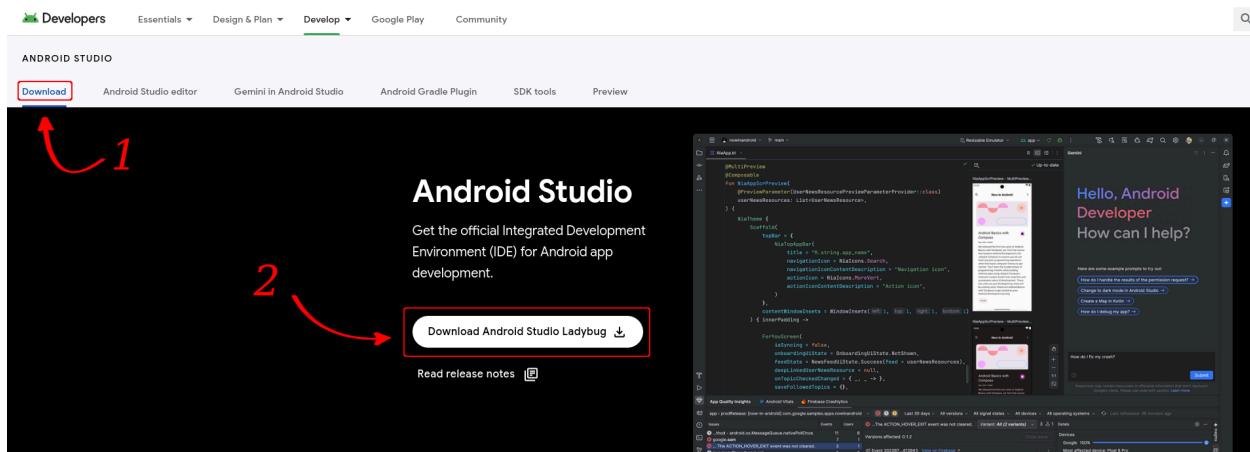
```
manuel@pclinux23:~$ sudo apt install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
libbz2-1.0:i386 ya está en su versión más reciente (1.0.8-2).
lib32z1 ya está en su versión más reciente (1:1.2.11.dfsg-2ubuntu1.5).
libc6:i386 ya está en su versión más reciente (2.31-0ubuntu9.16).
libstdc++6:i386 ya está en su versión más reciente (10.5.0-1ubuntu1~20.04).
libncurses5:i386 ya está en su versión más reciente (6.2-0ubuntu2.1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 67 no actualizados.
manuel@pclinux23:~$
```

## 2 - Descargando Android Studio

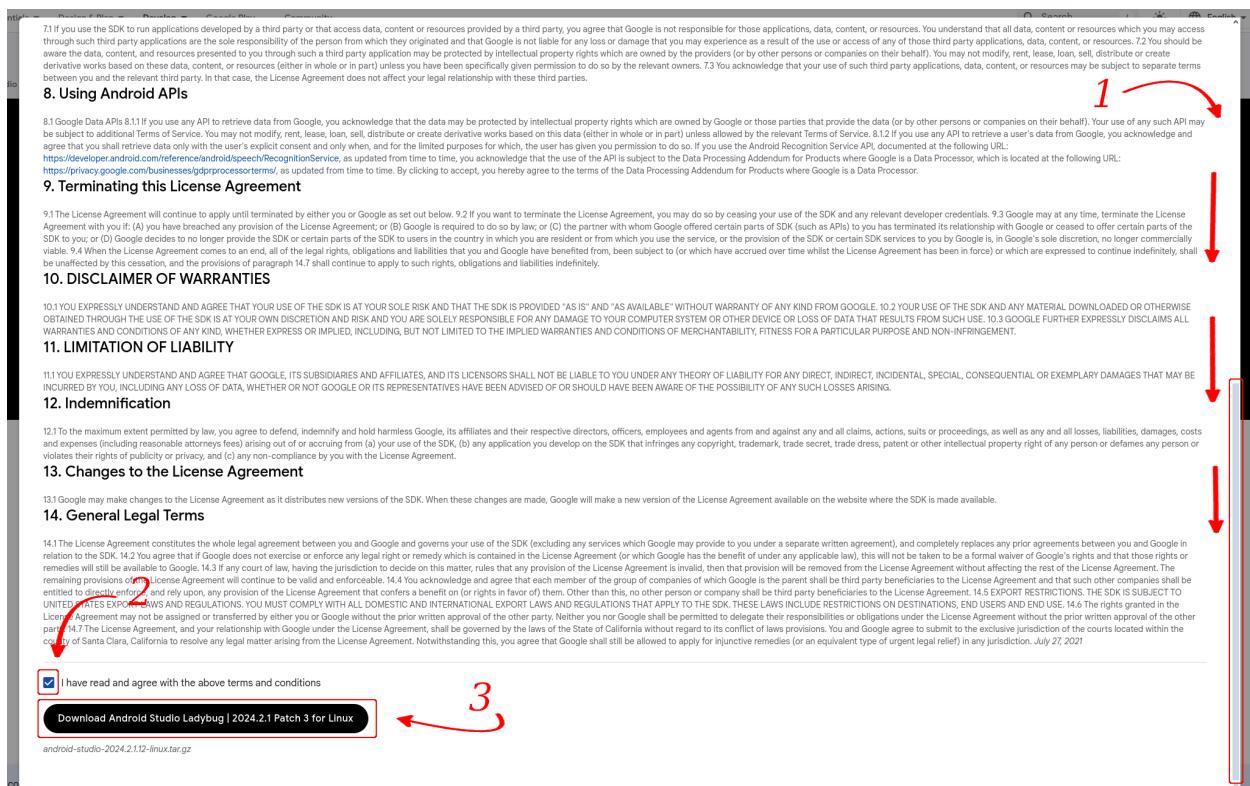
Para hacernos con una copia del IDE deberemos ir a la página web oficial ([developers.android.com](https://developer.android.com/studio)) y hacemos lo siguiente: Essentials > Get Android Studio



## Lo cual nos llevará a la siguiente página

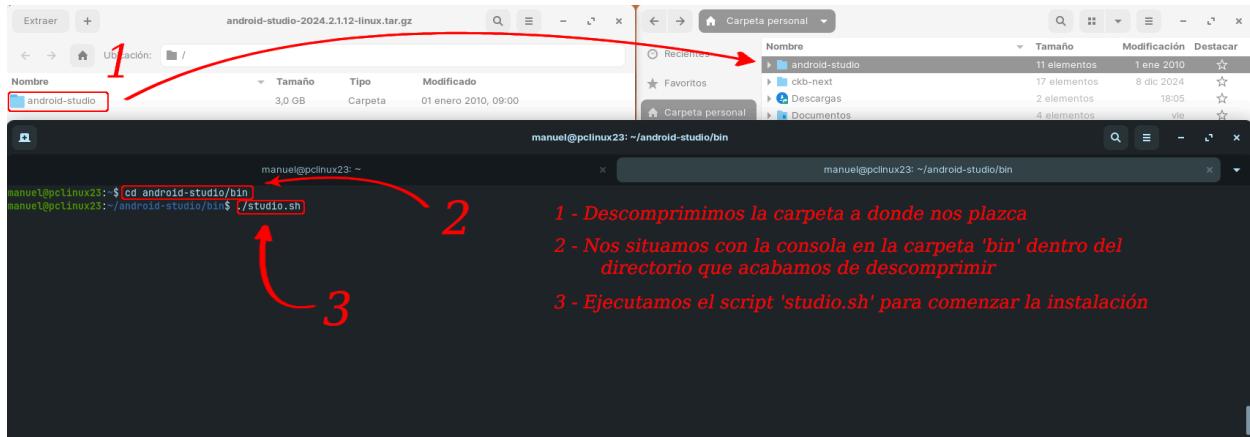


Nada más pinchar en botón (2) nos saltará la siguiente pantalla. La cual no sólo debemos leer sino comprender para ser totalmente conscientes de qué estamos aceptando... To pa'bajo y aceptar (seamos sinceros)

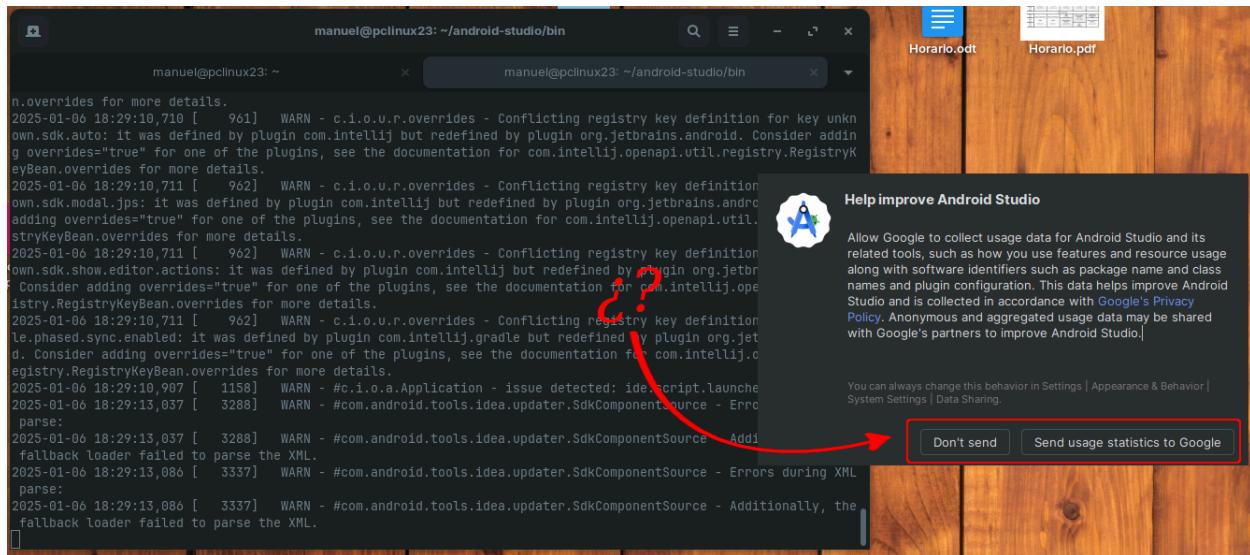


### 3 - Instando Android Studio

Nos descargará un archivo comprimido, el cual debemos descomprimir y hacer uso de la consola de la siguiente manera

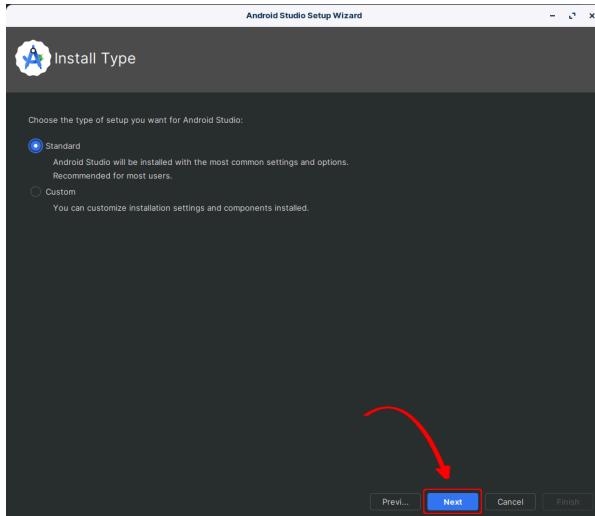


Una vez ejecutamos el *script* se nos dará la opción de enviar o no información de uso a Google... Queda al albur del usuario qué hacer

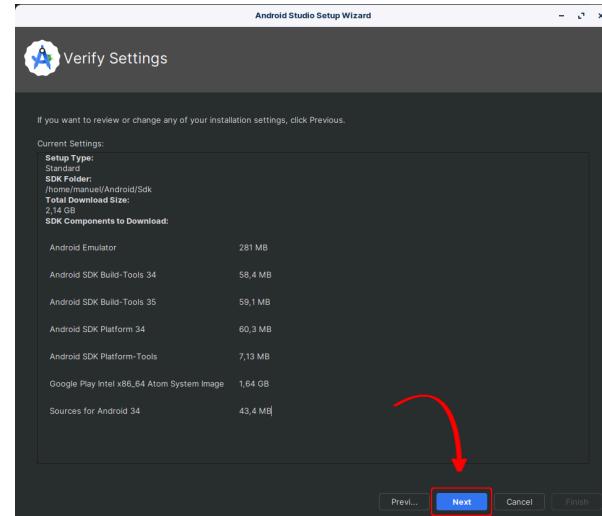


A lo largo de las siguientes pantallas iremos pinchando en el botón *next* haciendo alguna pausa en el caso de querer revisar algún dato

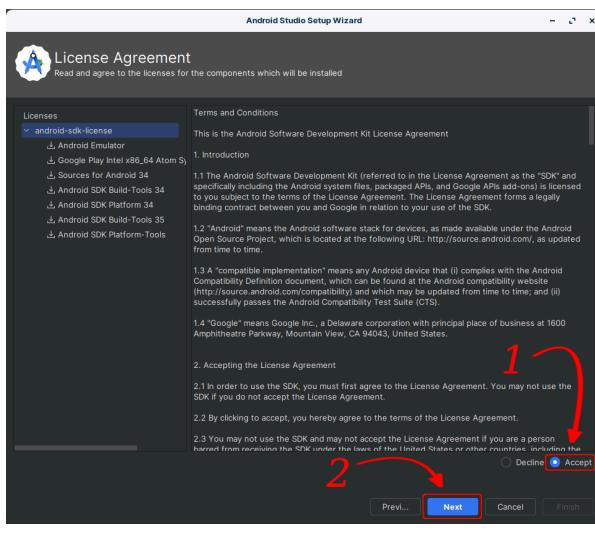
1



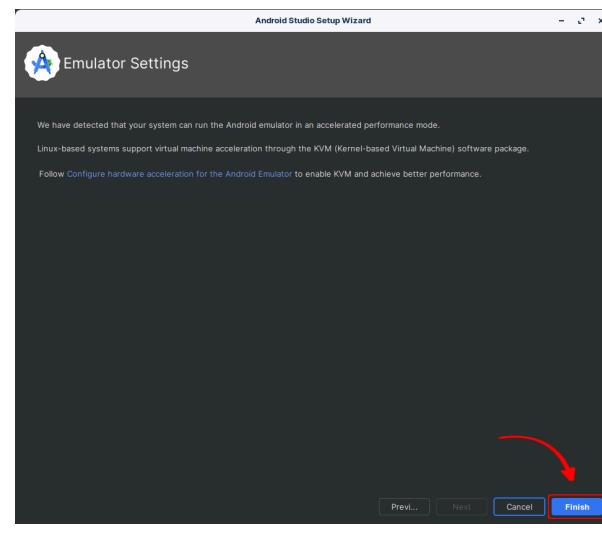
2



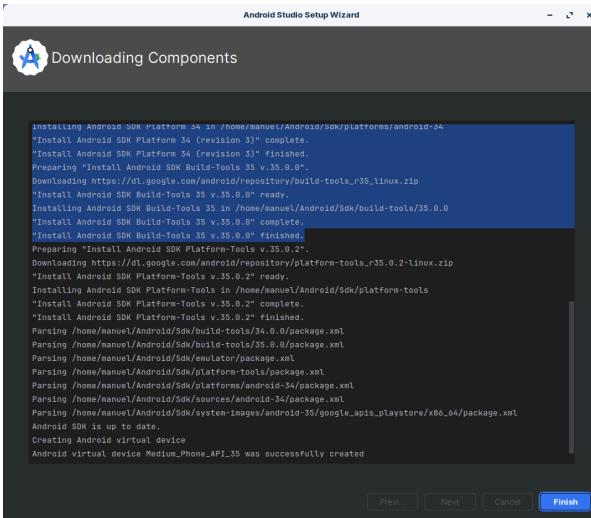
3



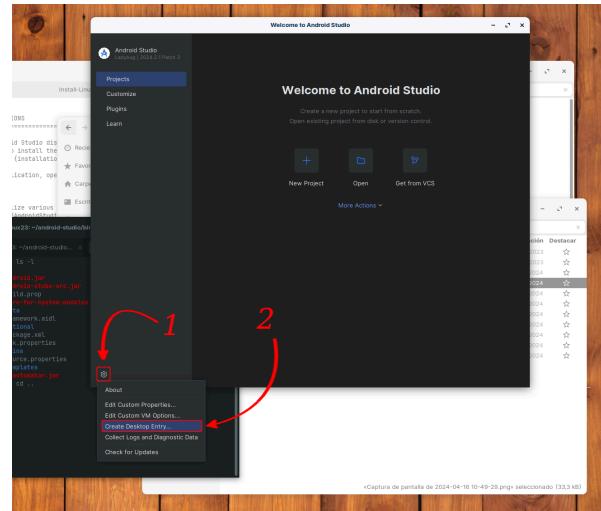
4



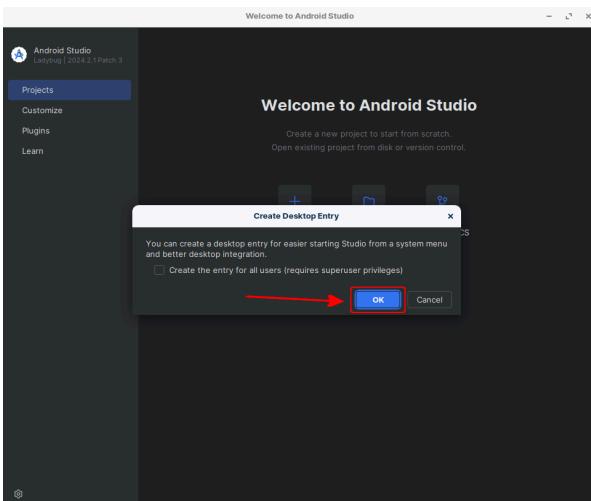
5



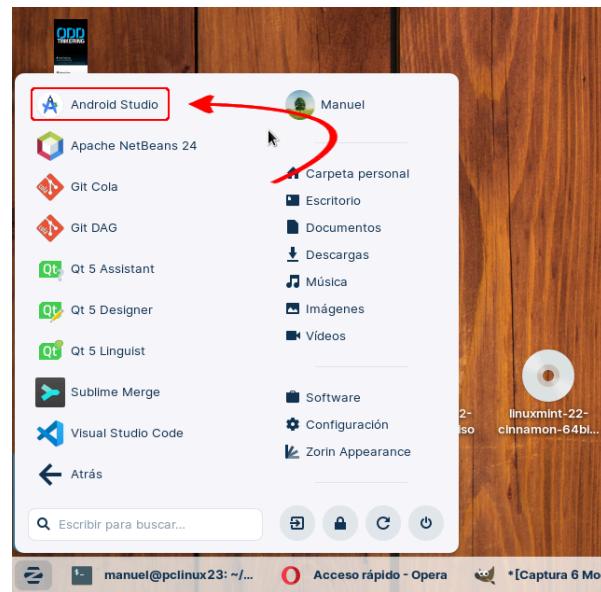
6 ( si queremos que aparezca en el menú de inicio)



7



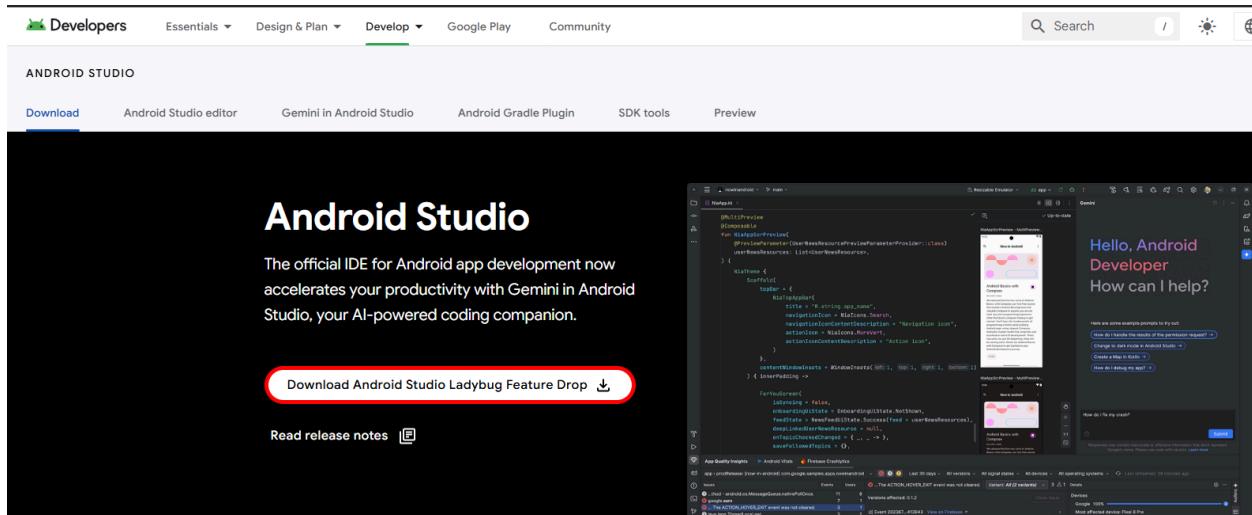
8



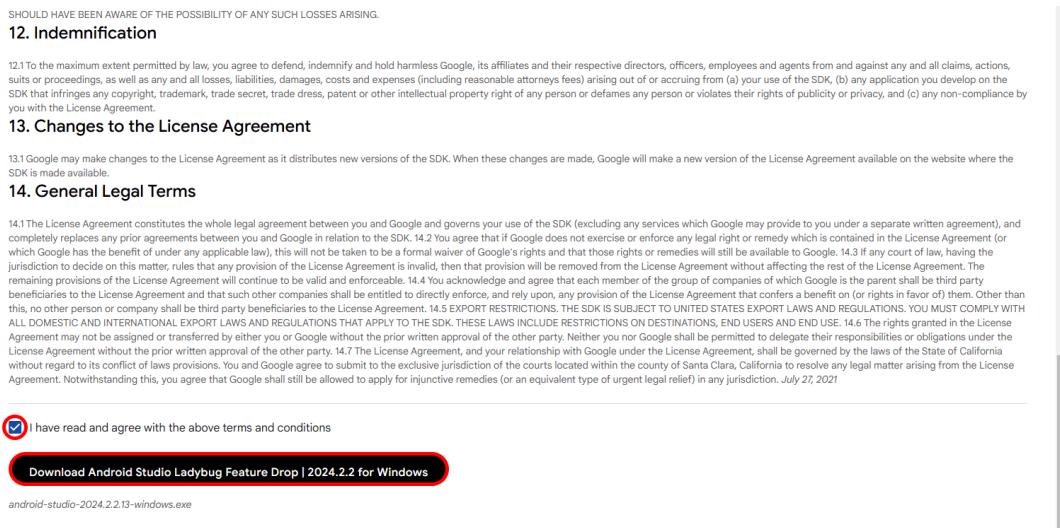
Y con esto ya sólo quedaría iniciar la aplicación como otra cualquiera

# Instalación en Windows

Para instalar Android Studio en Windows iremos a la siguiente [URL](#) donde veremos algo similar a los siguiente



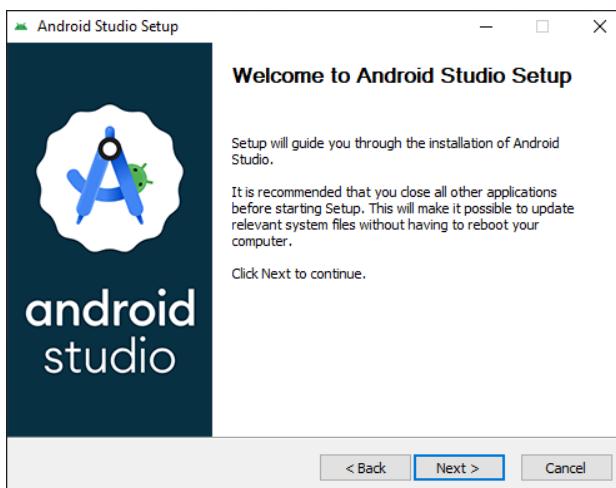
Aquí le daremos click al botón de “descargar” y se nos abrirá una ventana emergente en la que nos desplazamos hasta el final donde veremos lo siguiente



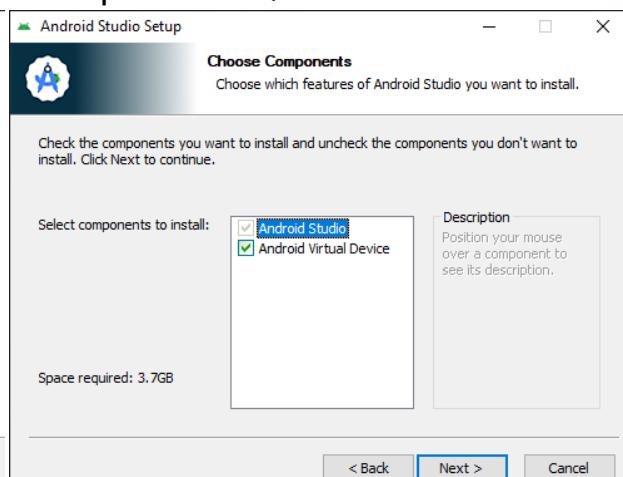
Aquí marcaremos la casilla de los términos y condiciones de uso y nos habilitará el botón de descarga.

Una vez terminada la descarga procederemos a abrir el ejecutable que descargamos. Luego de una breve espera, nos pedirá poder hacer cambios en el dispositivo y a continuación veremos las opciones de instalación

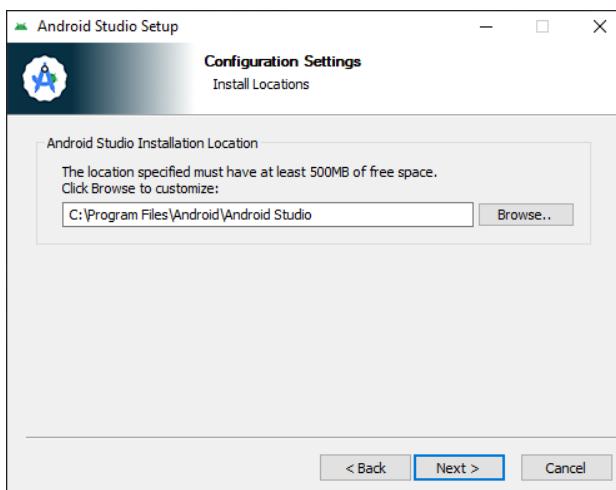
**1. Nos presentará el instalador.**



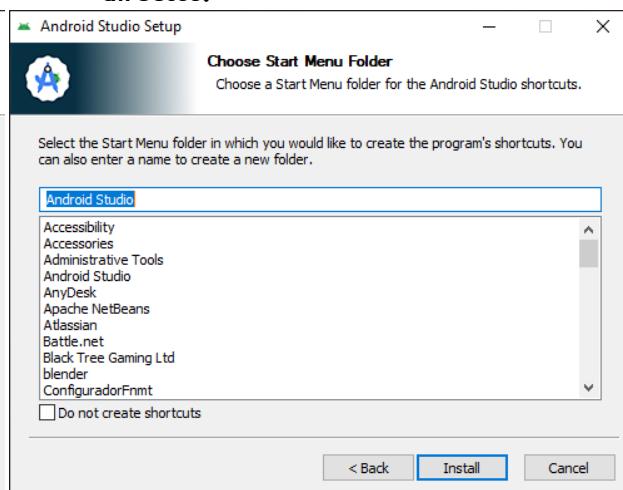
**2. Nos dará la opción de instalar un android phone virtual.**



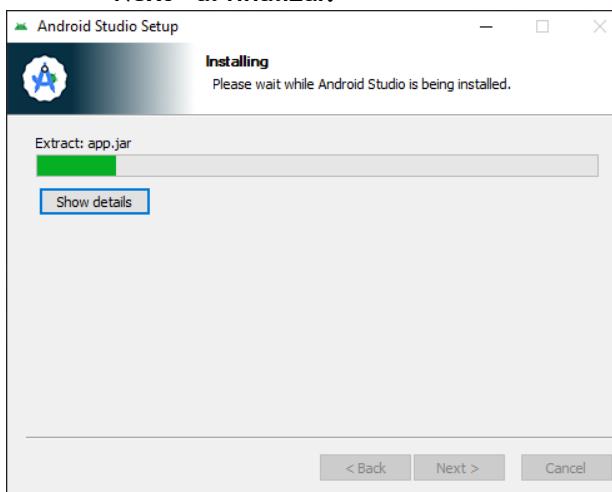
**3. Elegiremos la ruta de instalación.**



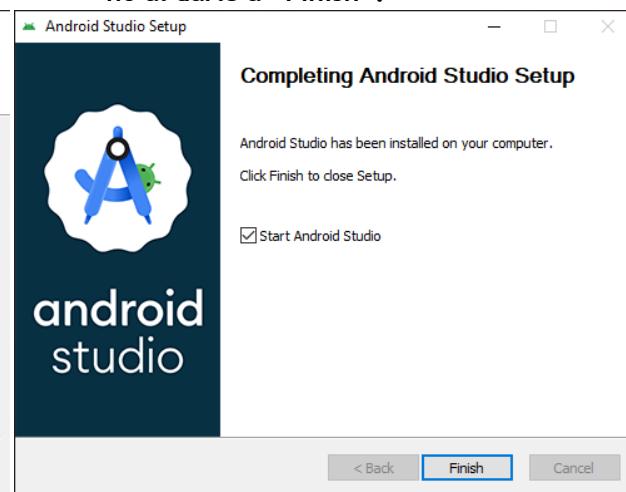
**4. Nos dará la opción de crear accesos directos.**



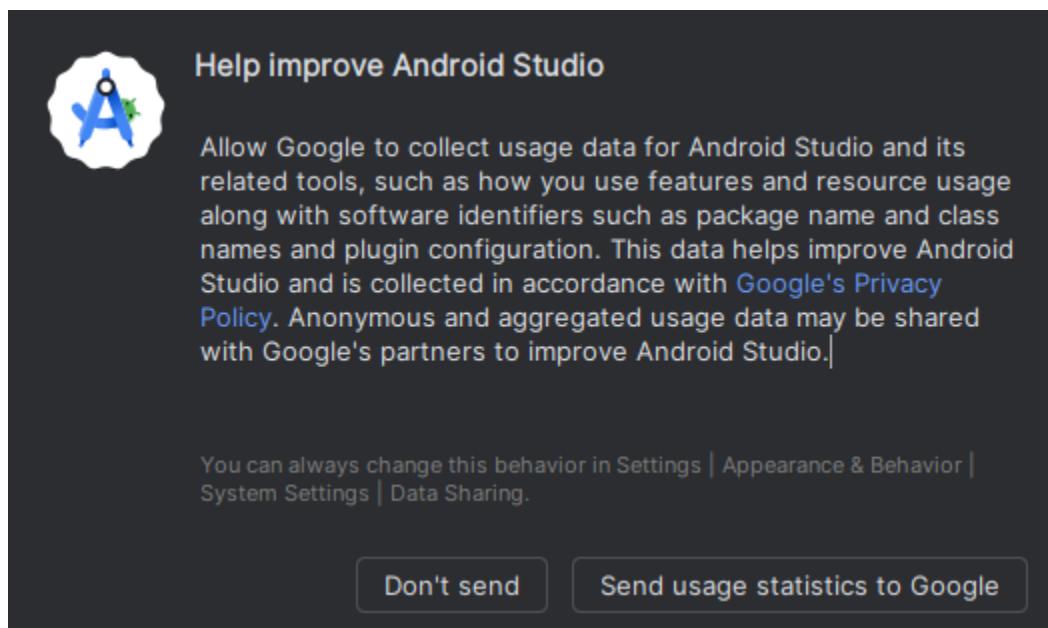
**5. Comenzará a instalarse y daremos a “Next” al finalizar.**



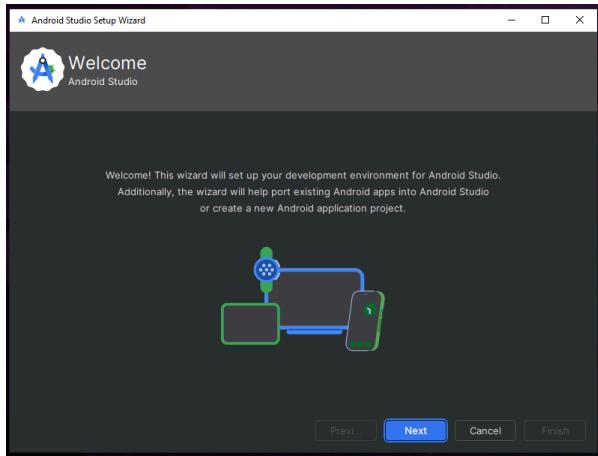
**6. Esta ventana nos dará la opción de abrir o no al darle a “Finish”.**



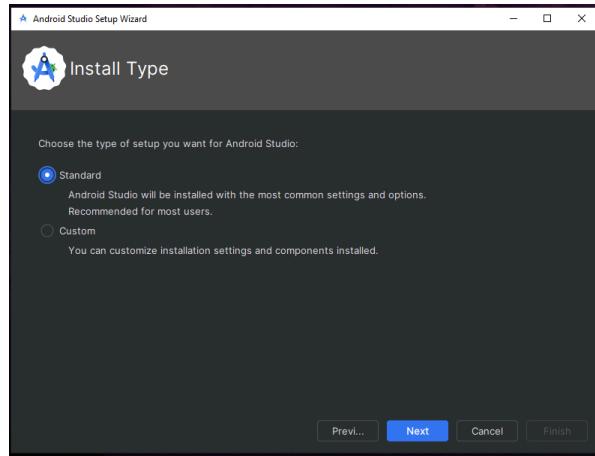
Al abrir Android Studio por primera vez nos preguntará si deseamos permitir a google recolectar datos del uso del IDE.



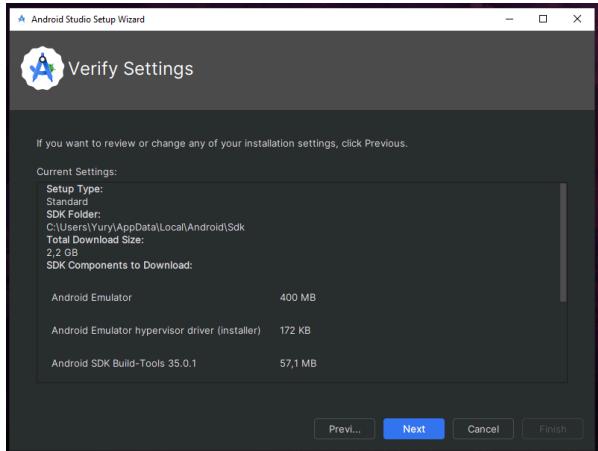
## 1. Nos dará la bienvenida.



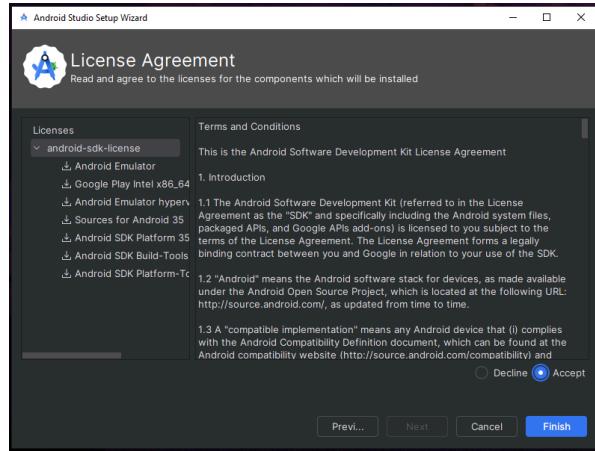
## 2. Nos dará la opción de instalación estándar o personalizada, elegiremos standard, la cual nos instalará todo.



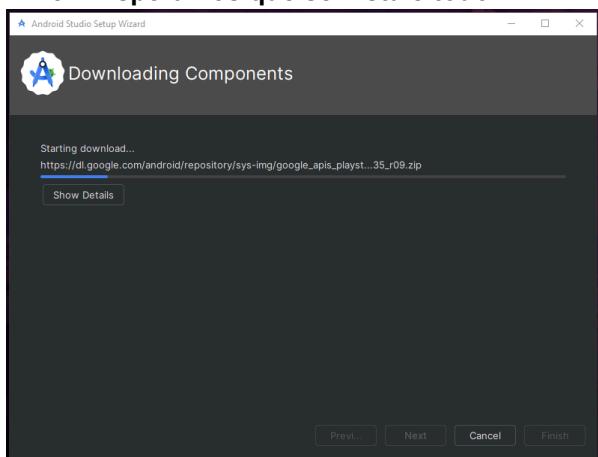
## 3. Previsualización de lo que se instalará



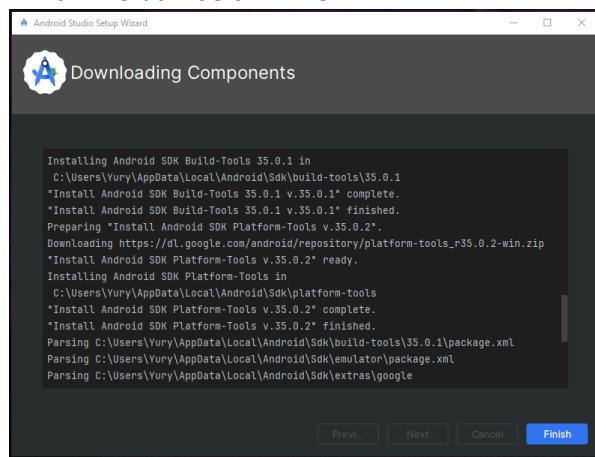
## 4. Aceptación de términos y condiciones de lo que se instalará



## 5. Esperamos que se instale todo



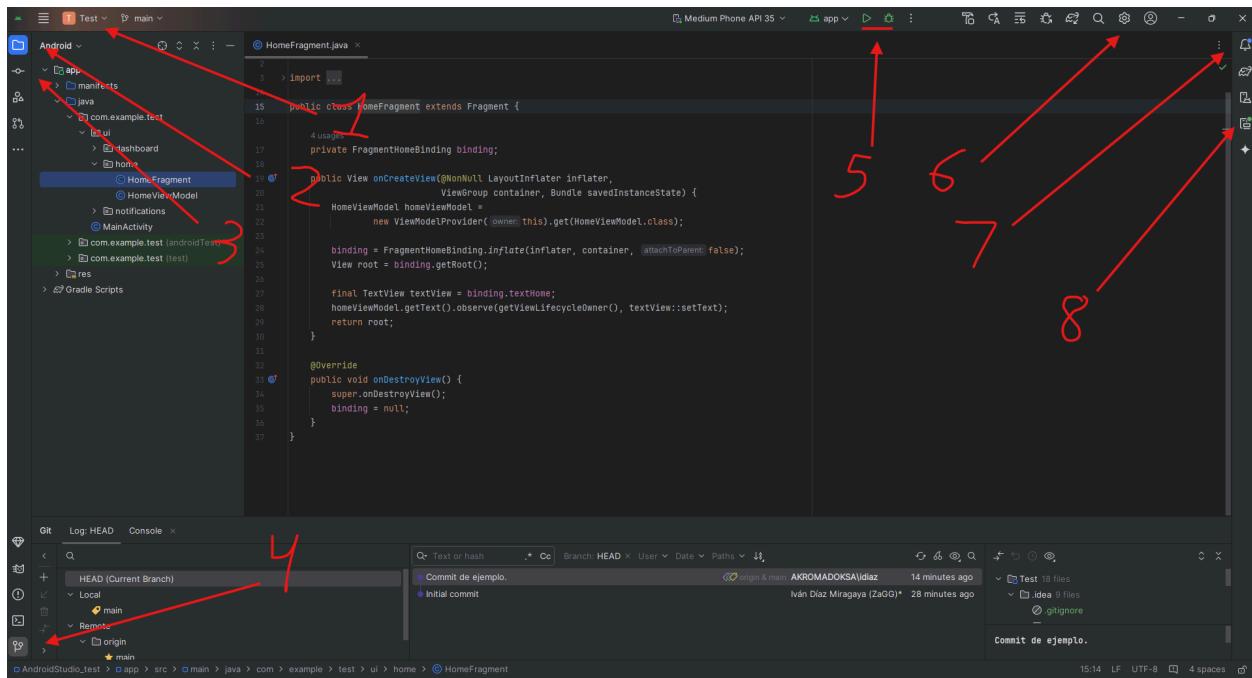
## 6. Le damos a "Finish"



# Interfaz de Android Studio

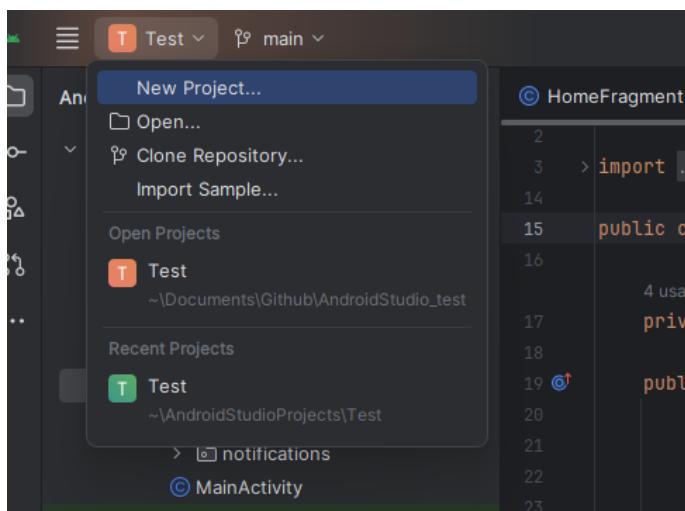
## 1 - Componentes mas usados/importantes.

Podemos resumir las partes importantes de la interfaz principal de Android en 8 partes:

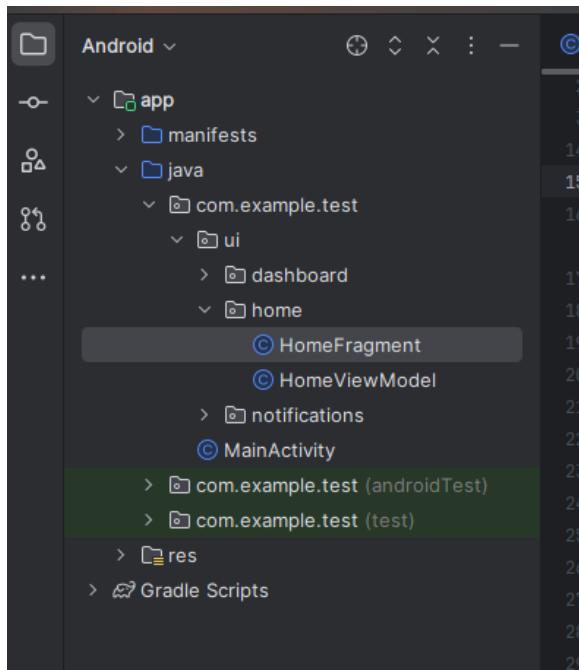


### 1) Acceso directo a proyectos recientes y repositorios.

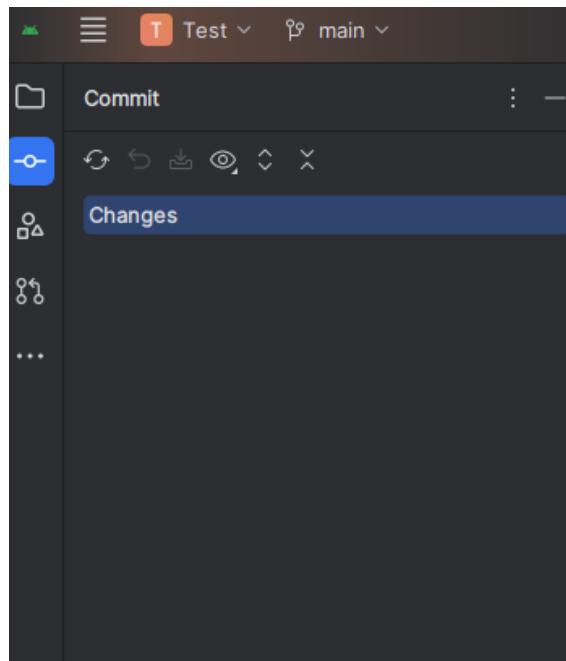
En esta sección podemos acceder a proyectos anteriores con un solo click, clonar repositorios etc.



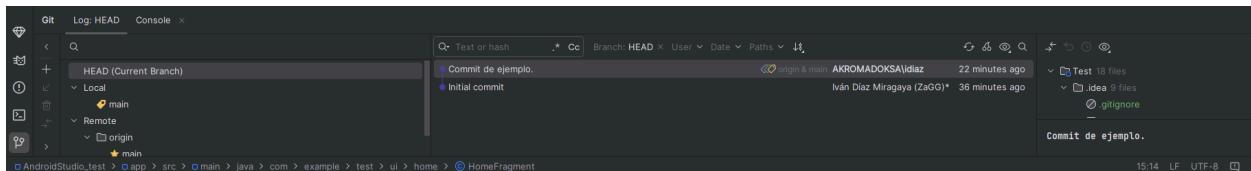
- 2) Sección en donde podemos visualizar, editar, crear, eliminar etc. los archivos y carpetas que componen nuestro proyecto:



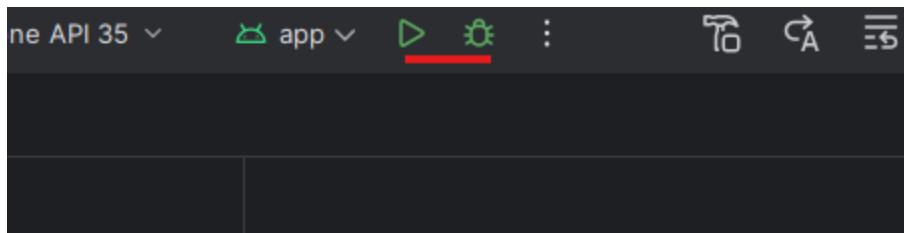
- 3) Acceso directo para poder realizar cambios en un repositorio:



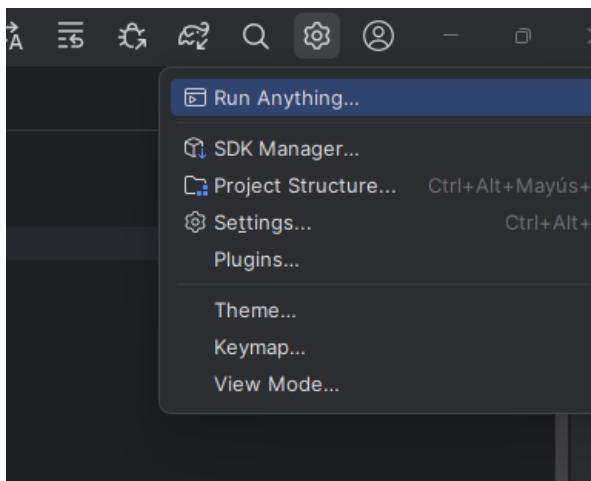
4) Acceso directo a las ramas del repositorio y numerosas herramientas para interactuar con él empleando git:



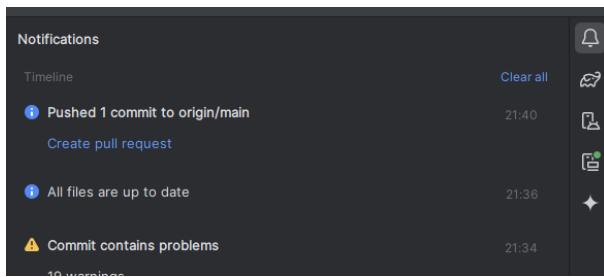
5) En estos botones podemos depurar o ejecutar nuestra app:



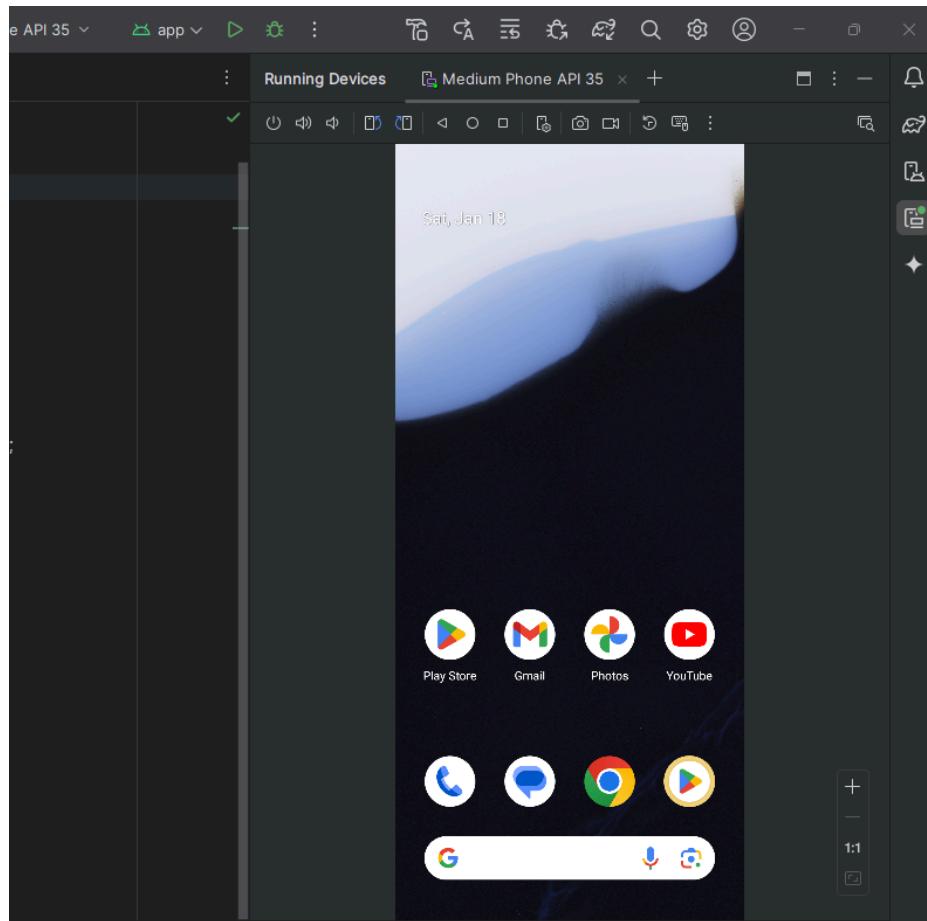
6) Opciones del IDE:



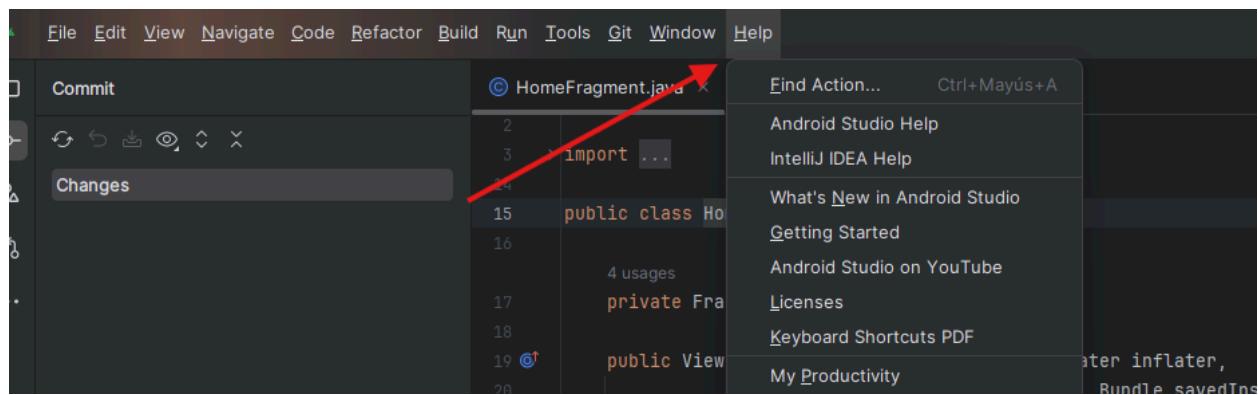
7) Notificaciones:



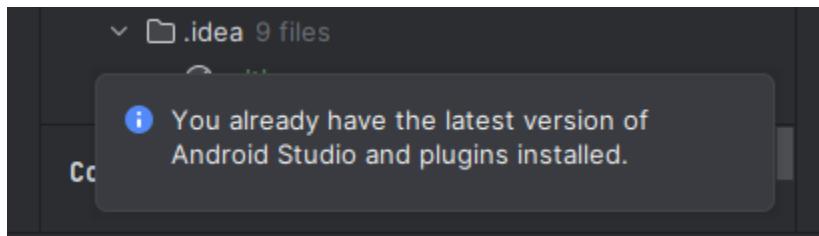
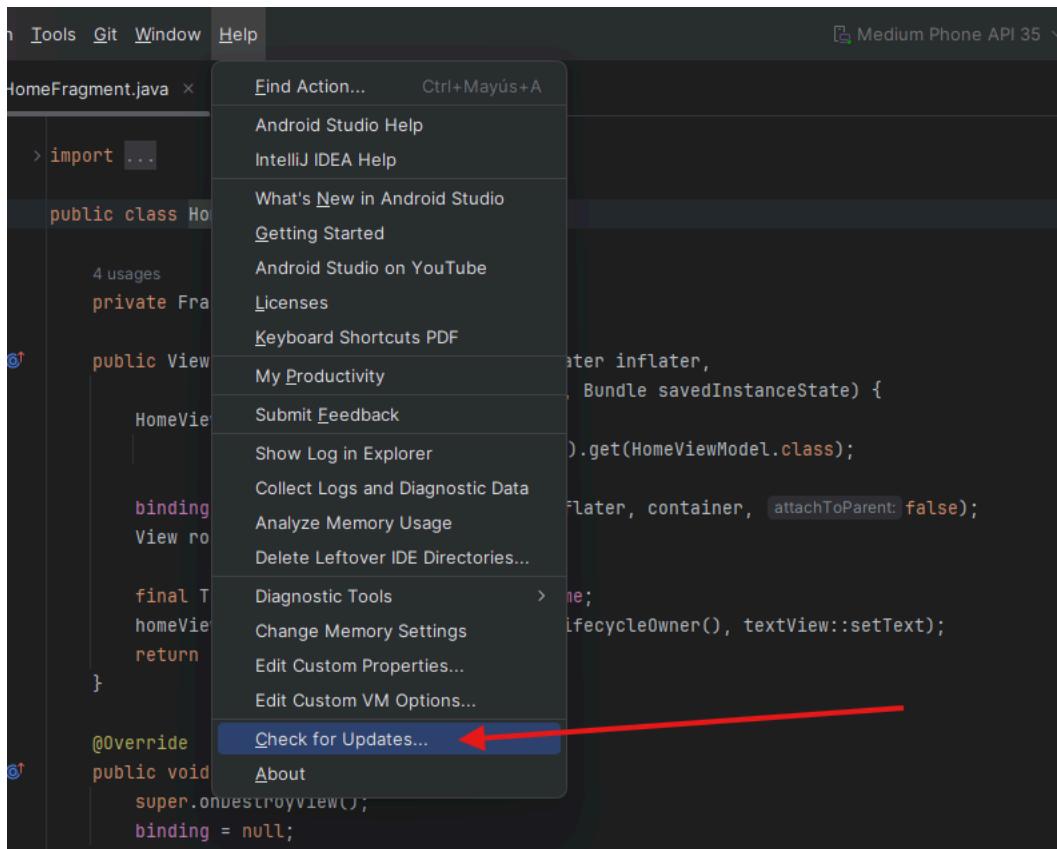
- 8) En esta sección tenemos la opción de virtualizar un móvil Android para probar la app en local sin usar un móvil externo o tener que compilar el apk.



Para actualizar el IDE debemos de ir a “ayuda”:



Y darle a la opción de “buscar actualizaciones”:

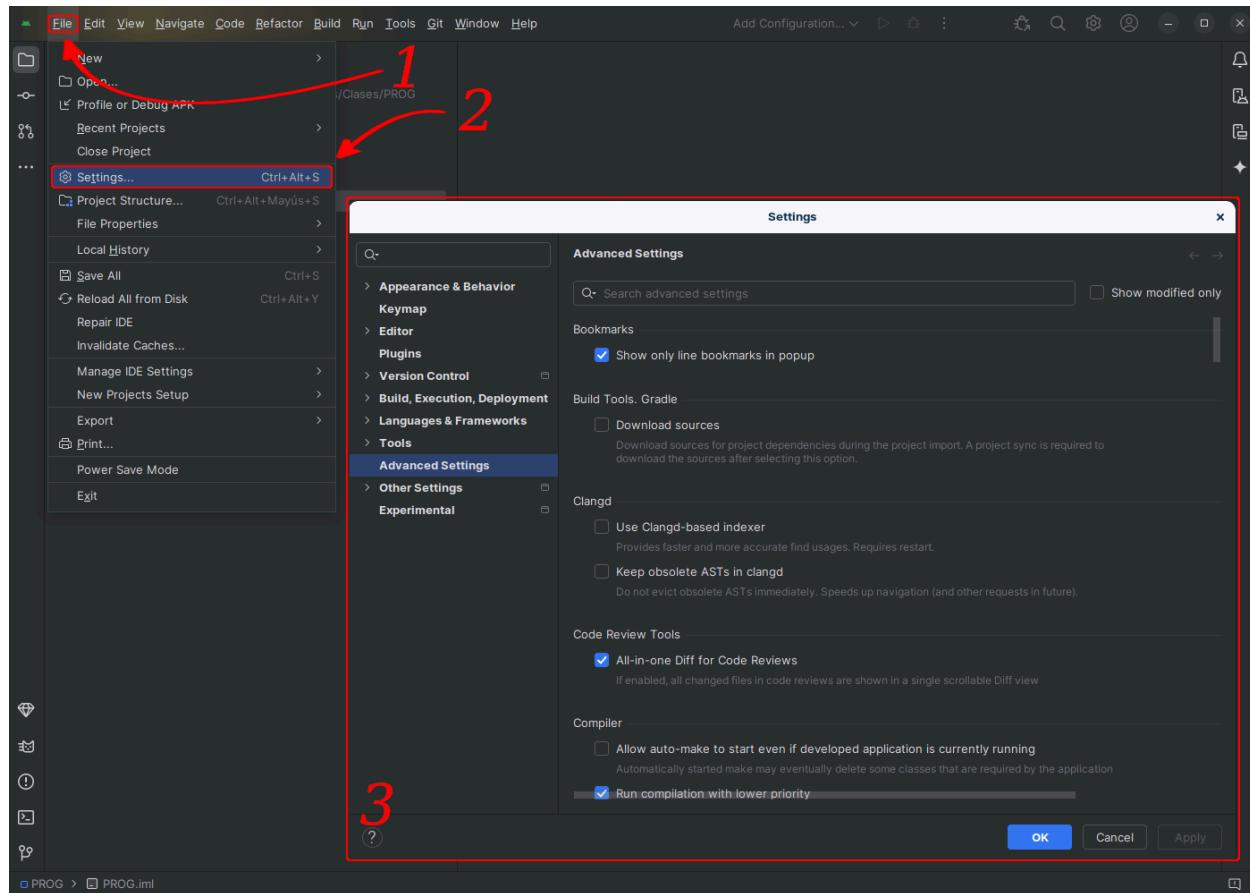


Como se puede ver en la notificación, no solo busca actualizaciones del IDE en sí; también busca actualizaciones para los plugins.

Android Studio cuenta con muchas más funcionalidades, pero estas son de las más importantes y básicas que debemos conocer sobre el IDE.

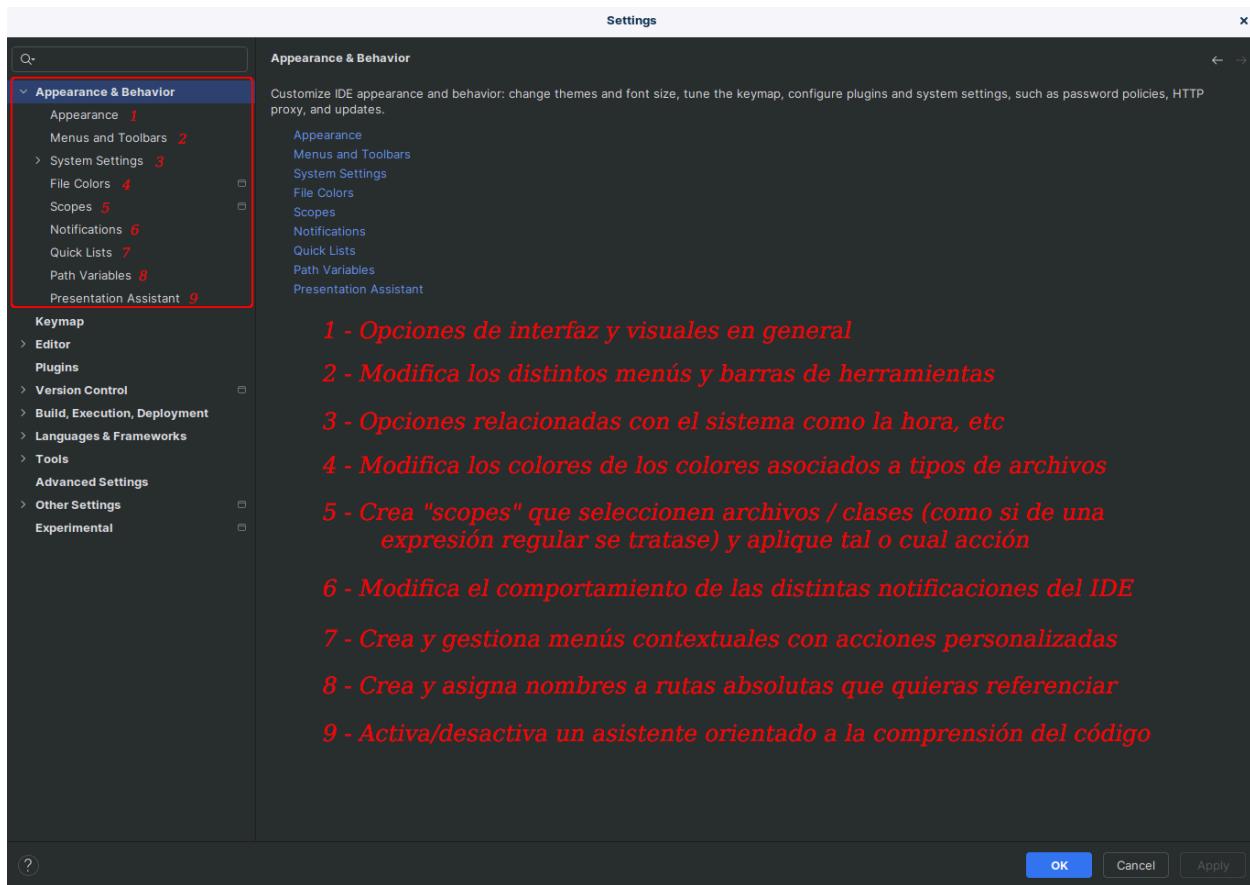
# Opciones de configuración del IDE

Para acceder a la ventana desde la que podremos tocar las distintas configuraciones tenemos que seleccionar ‘File’ > ‘Settings’

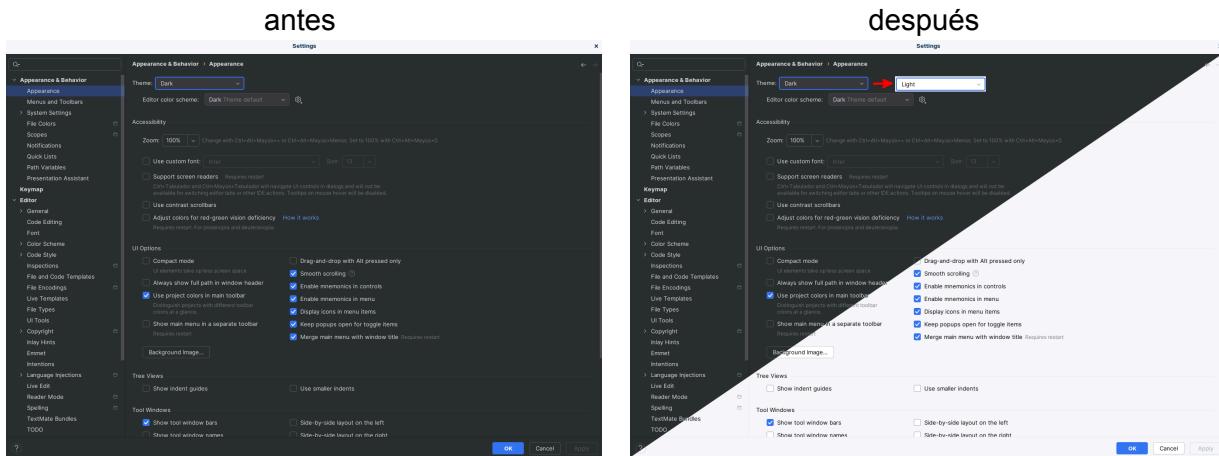


Como se puede apreciar, existen multitud de cosas que podemos modificar. Muchas más de las que yo sepa/pueda explicar, de hecho. Así que, *grosso modo*, haré un breve resumen de aquellas que sí entiendo.

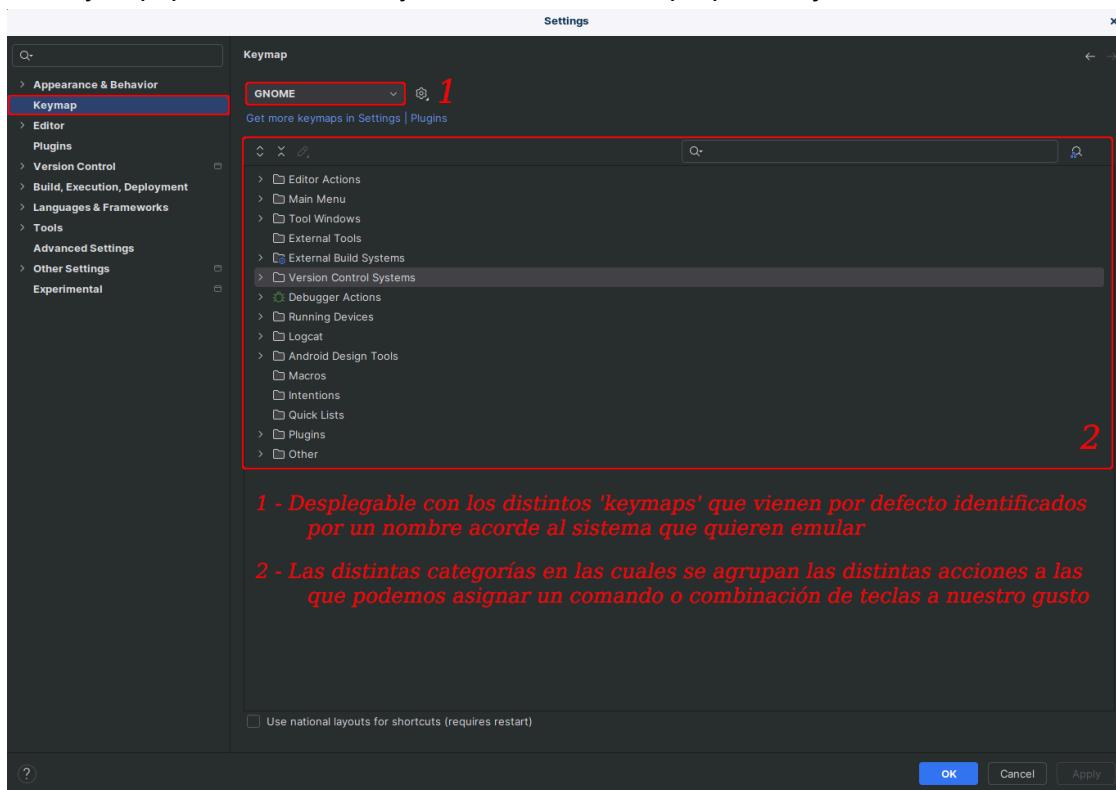
Vemos que hay distintas categorías. La primera toca la apariencia y el comportamiento de algunos aspectos del IDE



## Cambio de modo oscuro a claro (ejemplo)



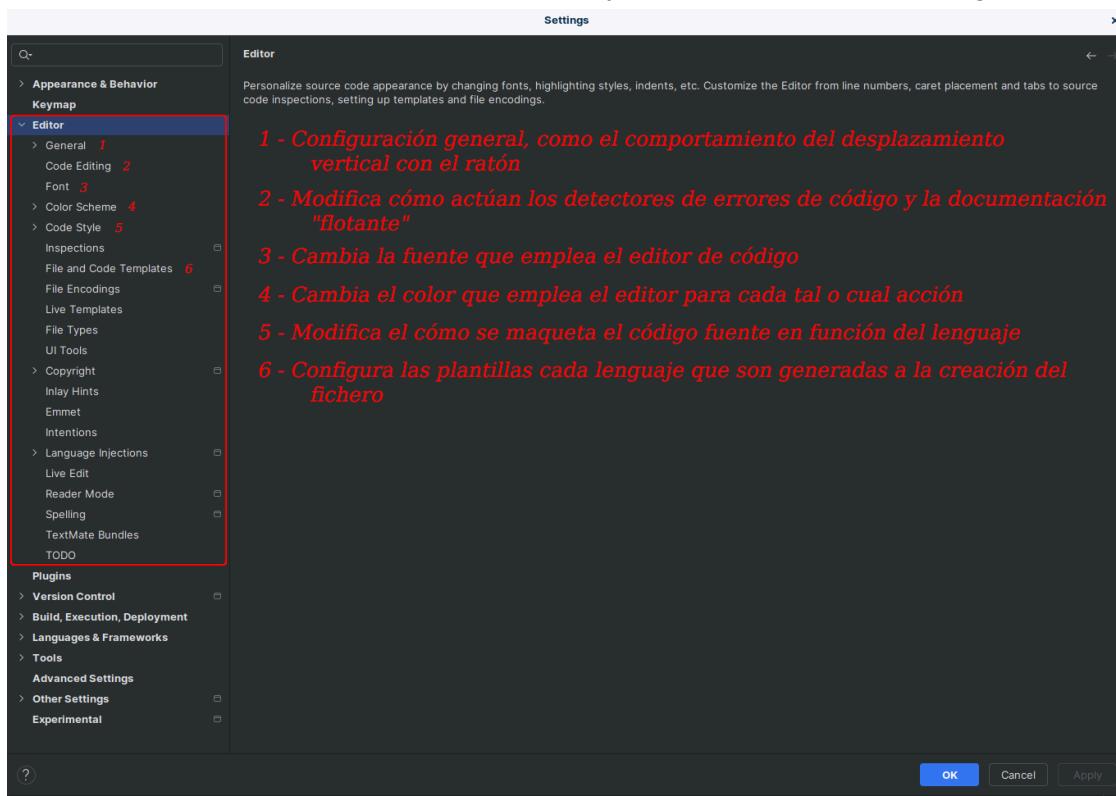
En 'keymap' podremos añadir y modificar nuestros propios atajos de teclado



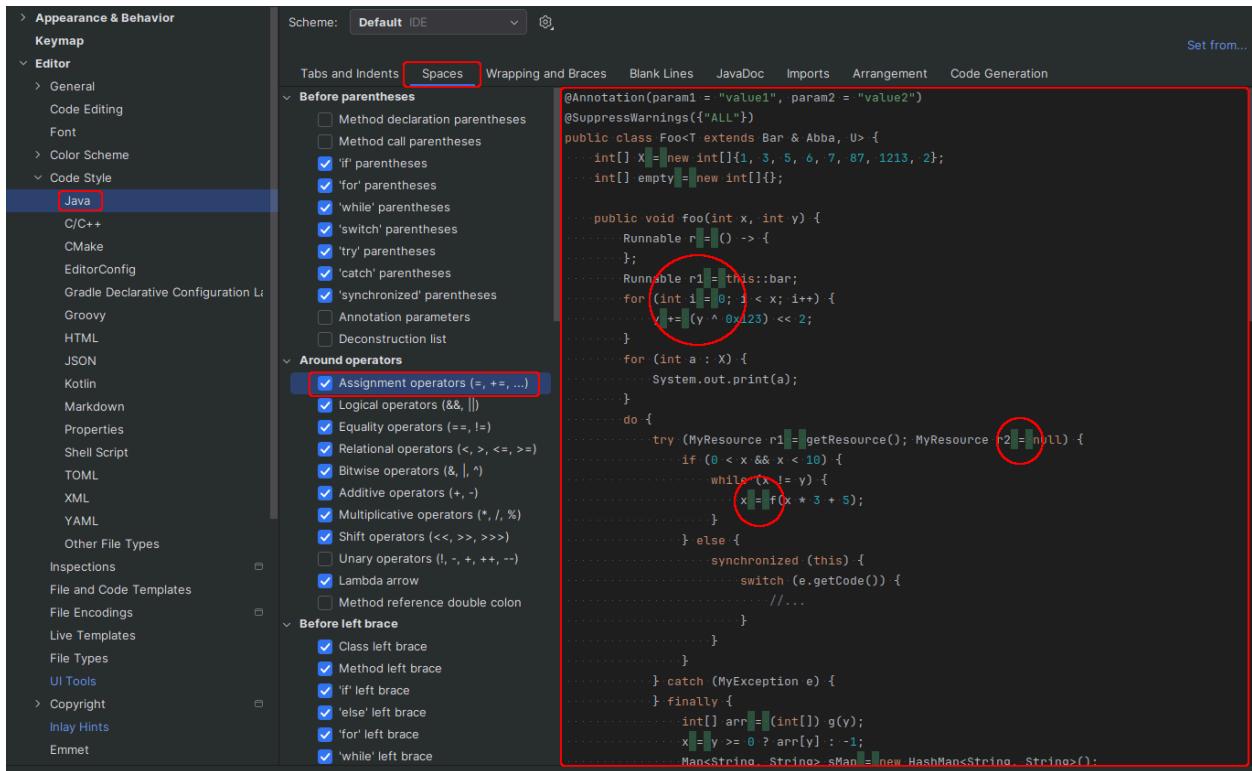
1 - Desplegable con los distintos 'keymaps' que vienen por defecto identificados por un nombre acorde al sistema que quieren emular

2 - Las distintas categorías en las cuales se agrupan las distintas acciones a las que podemos asignar un comando o combinación de teclas a nuestro gusto

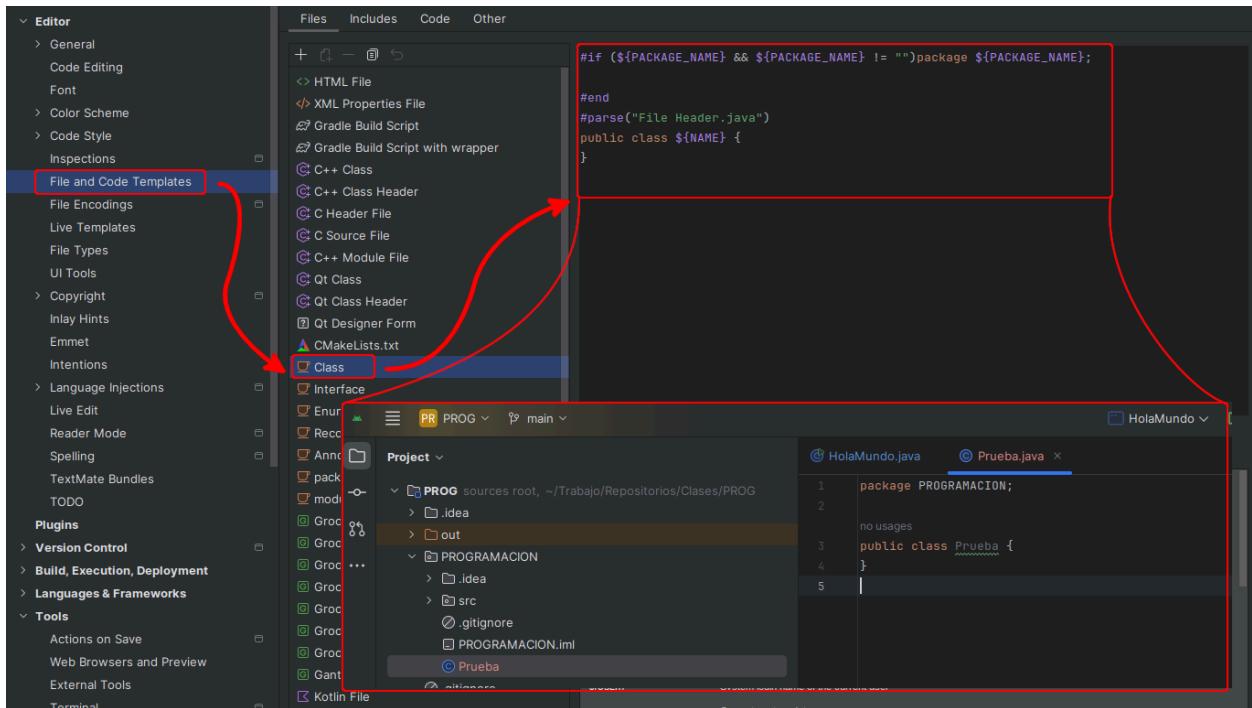
Aquí podremos cambiar la fuente, color, estilo y apariencia de nuestro código



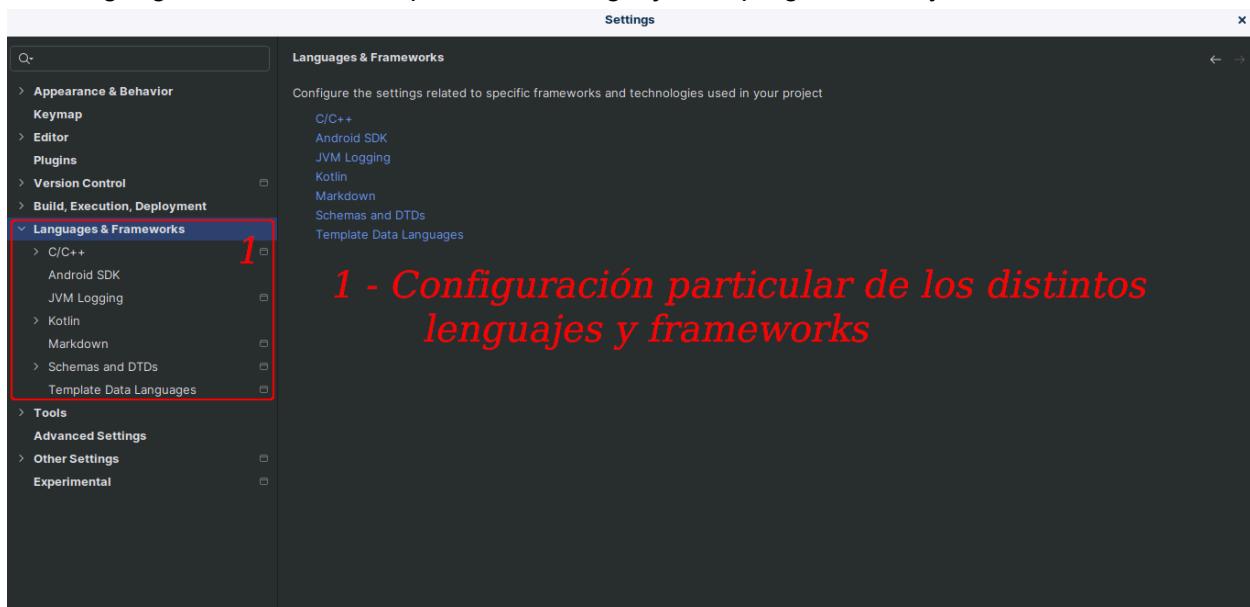
Vemos cómo podemos cambiar cómo se muestra nuestro código (cribado por lenguajes). Sirvan como por ejemplo los espacios, si queremos que se muestren antes de los operadores de asignación. Podemos ver el resultado de las opciones en el código de prueba de la izquierda



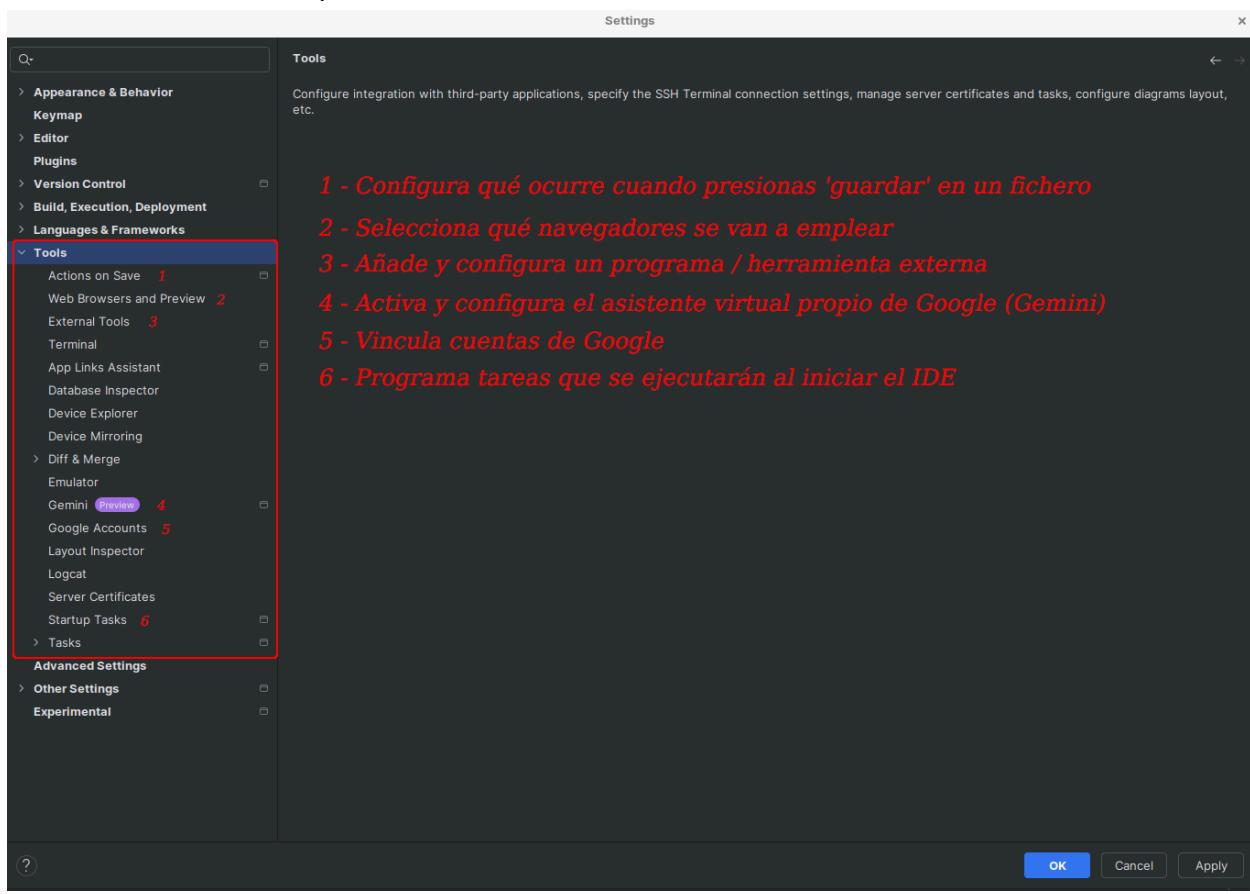
También podemos definir las plantillas al crear un nuevo documento (véase cómo el código superior se ve reflejado cuando creamos una nueva clase de Java)



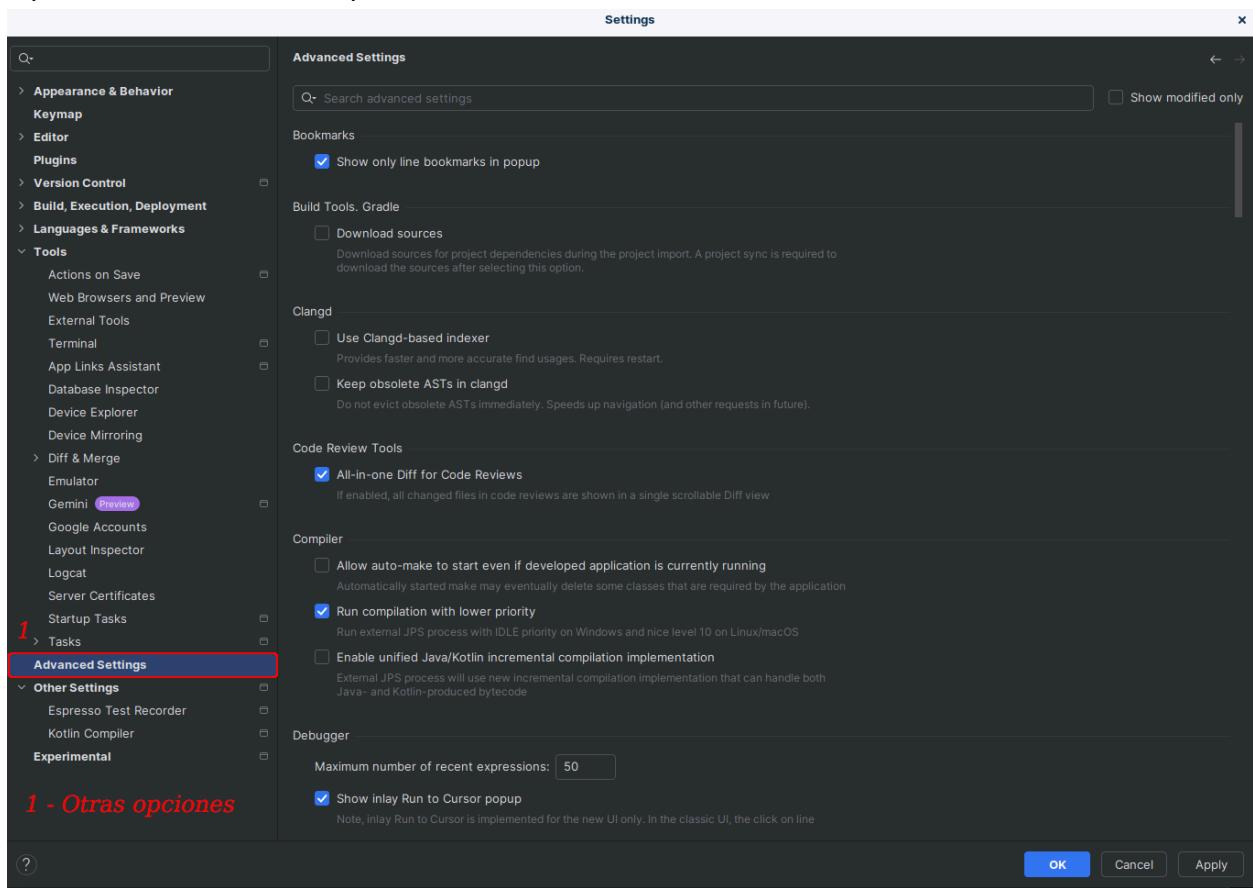
En 'Languages & Frameworks' pues... eso lenguajes de programación y frameworks



En 'Tools' tenemos las opciones relacionadas con herramientas externas.



Y por último tenemos las opciones avanzadas

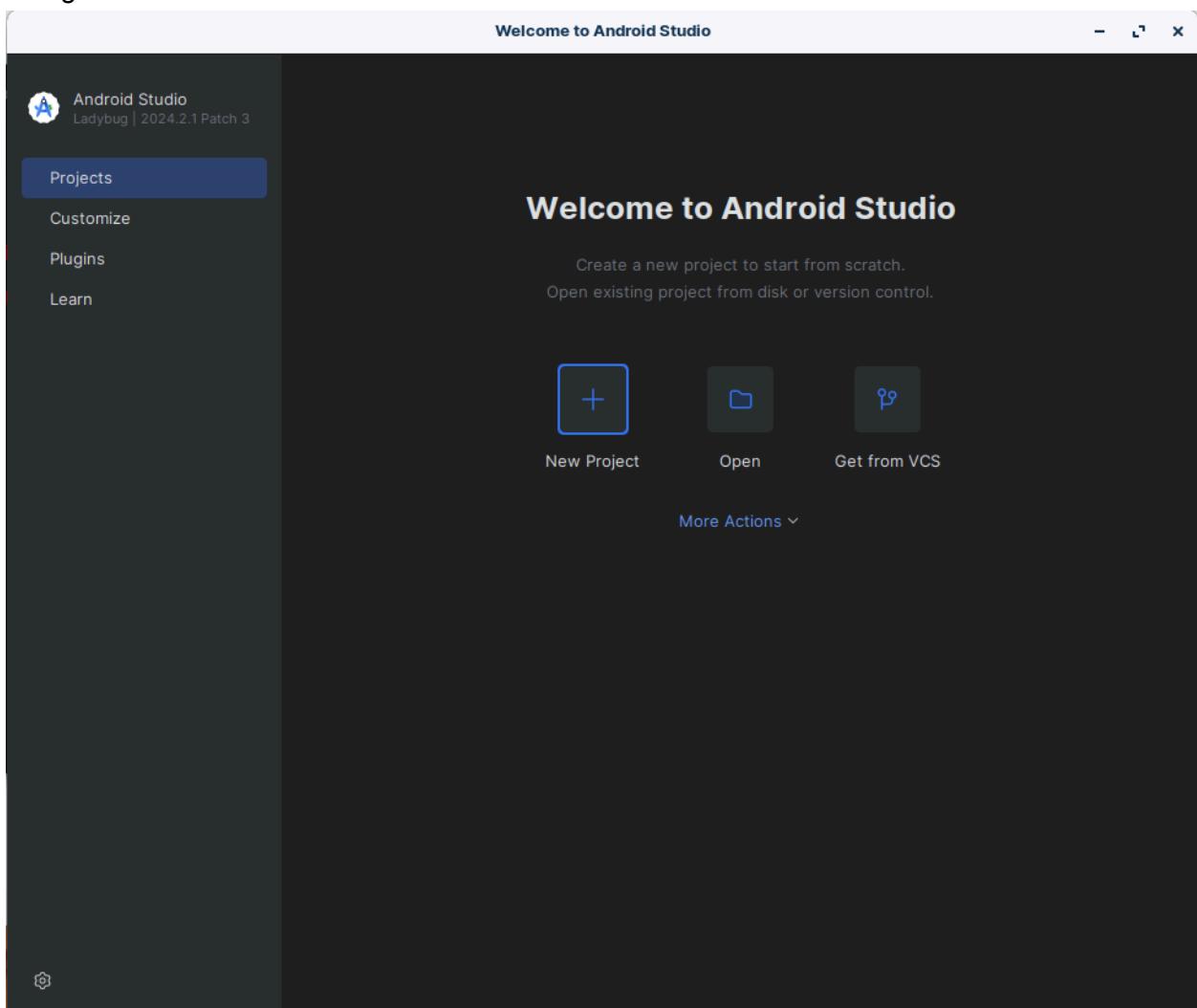


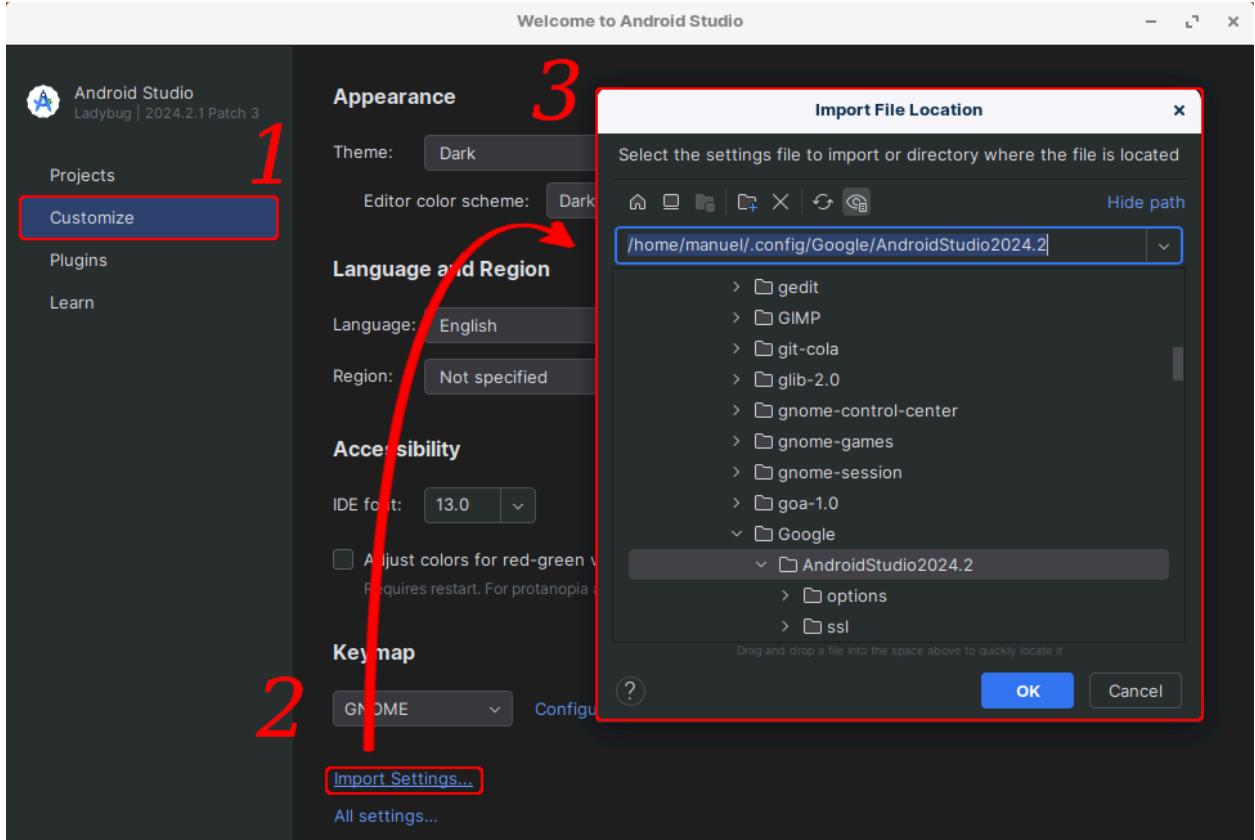
## Importar y exportar configuración

Tras abrir la aplicación nos saldrá el siguiente cuadro

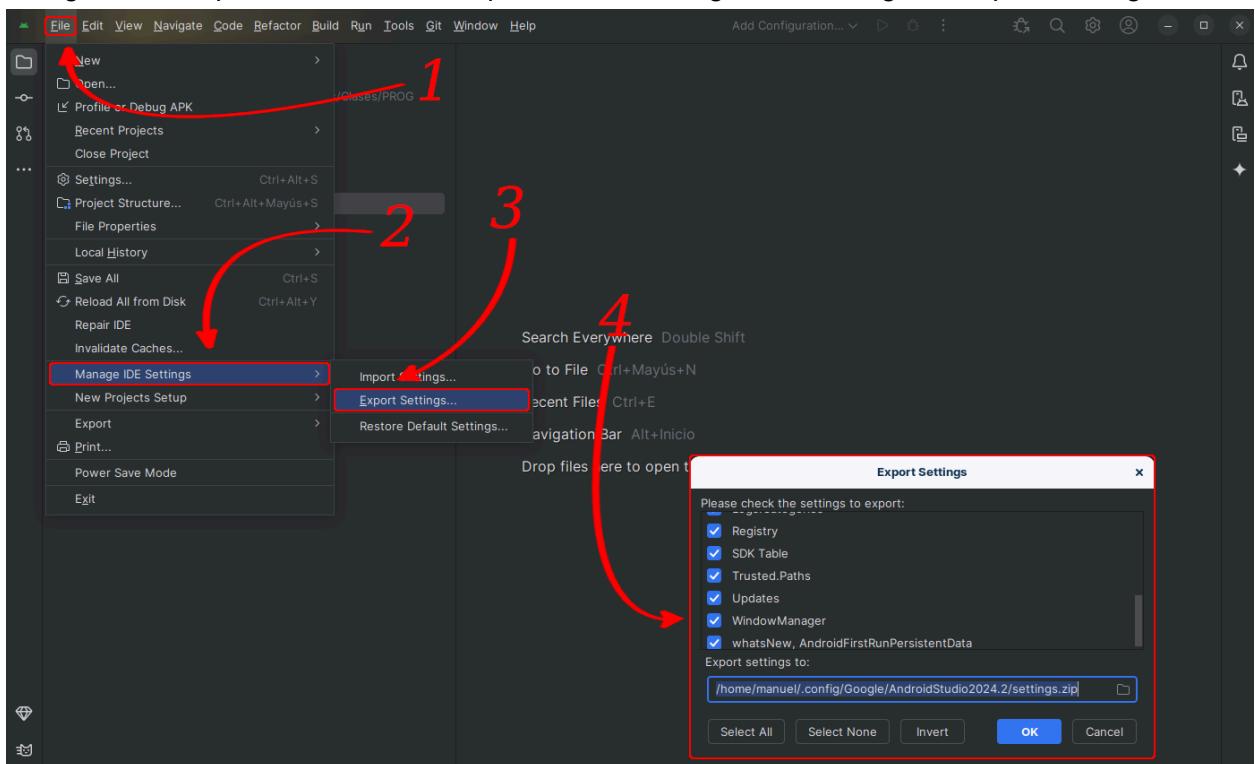
Si acabamos de instalar el IDE en una máquina nueva y queremos mantener la configuración que teníamos podemos hacerlo de varias formas. La primera y más sencilla sería, tras iniciar la aplicación, hacer click en ‘Customize’ > ‘Import settings’ y seleccionar nuestros archivos de

configuración.

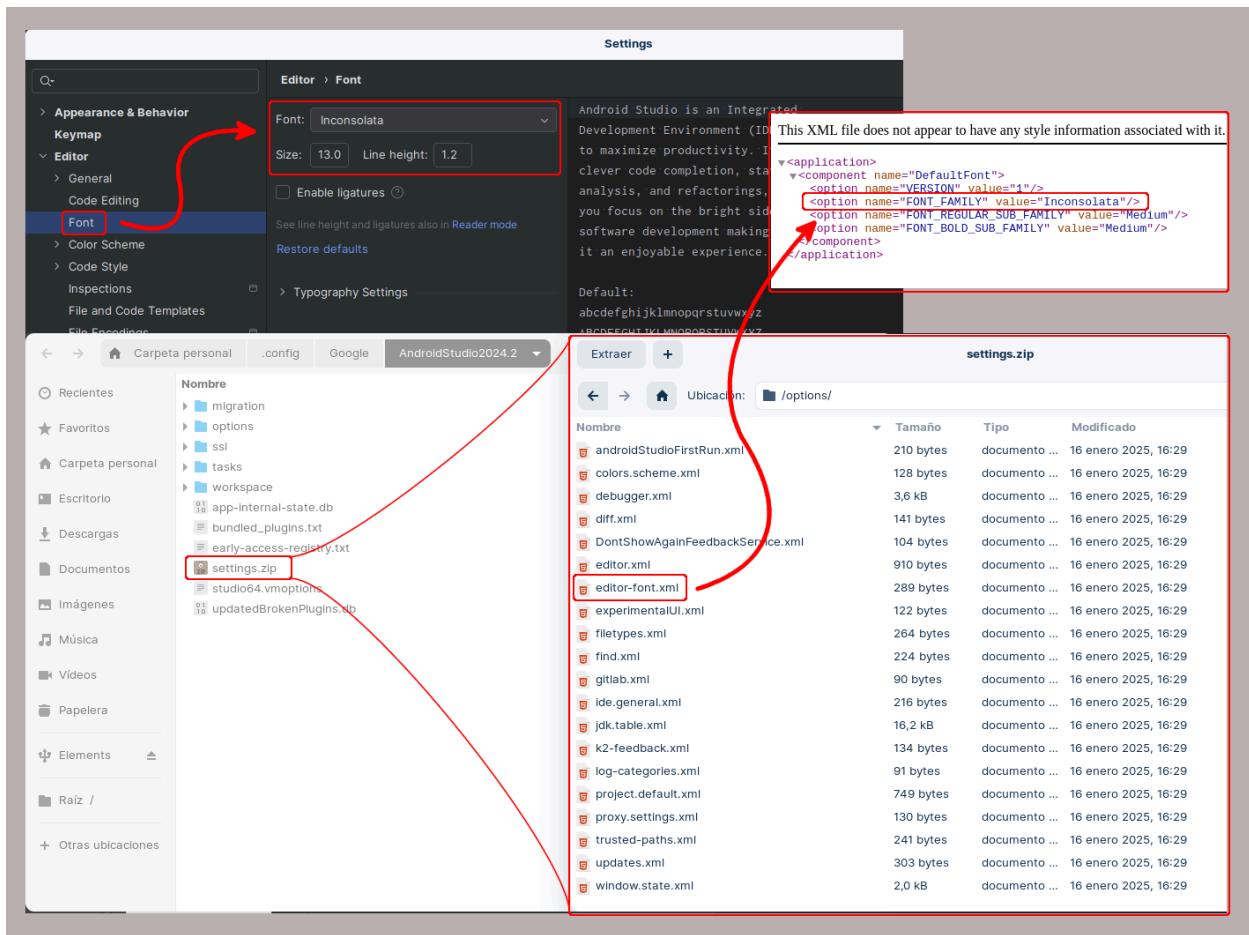




En caso de que quisieramos hacer un cambio de máquina y aún no tuviésemos nuestra configuración exportada tendríamos que 'File' > 'Manage IDE Settings' > 'Export Settings'



Estaríamos exportando un archivo comprimido con una serie de ficheros XML. Algo tal que así: Cambiaré la fuente para que quede reflejada en el XML y podamos ver el resultado.

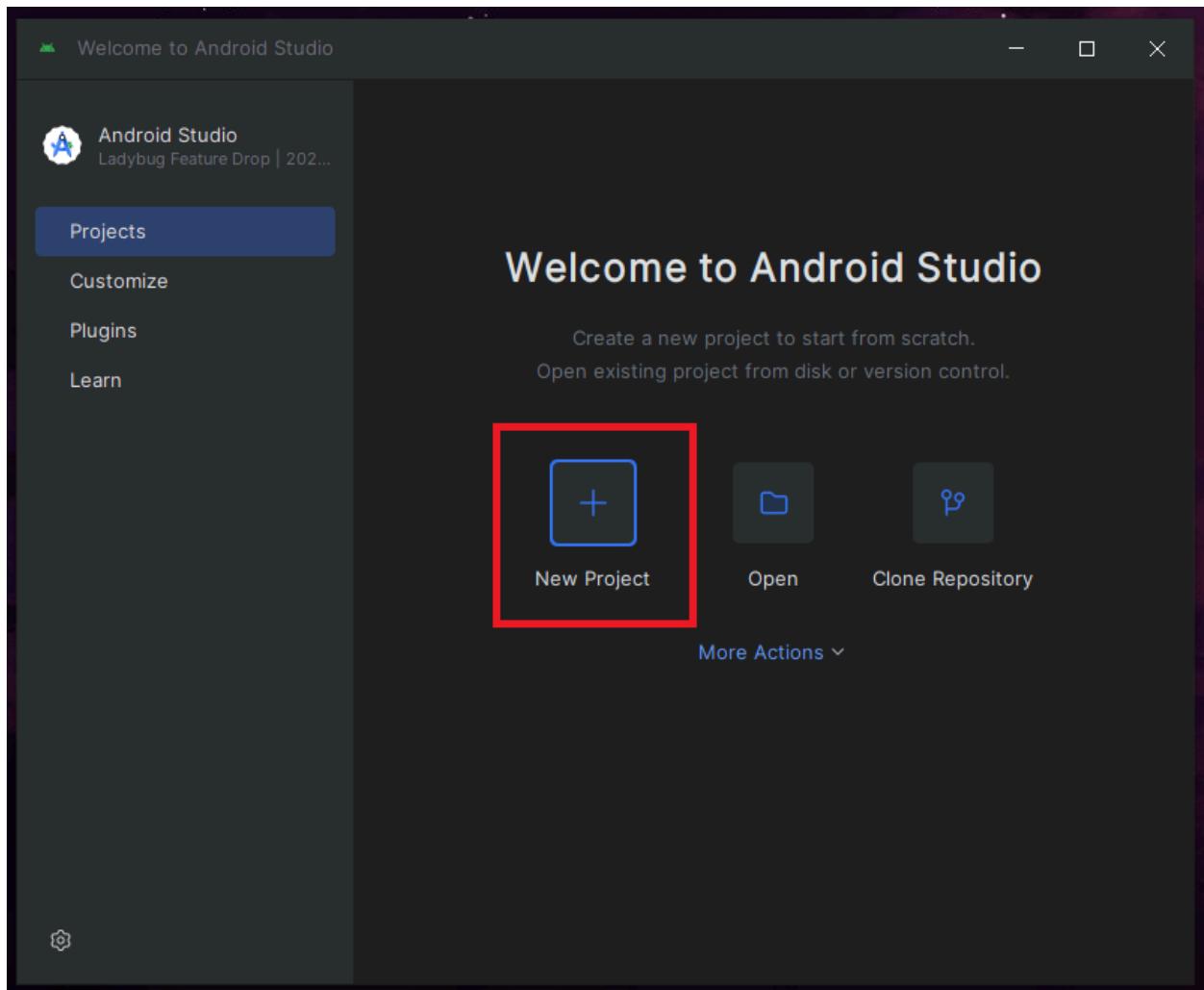


Una vez exportados los ficheros de configuración podríamos importarlos como se describe al principio de la sección o seguir una ruta parecida a la de la exportación pero seleccionando la opción de 'Import Settings'. Sólo nos quedaría buscar y seleccionar nuestros ficheros exportados en el paso anterior con el navegador que nos saltaría.

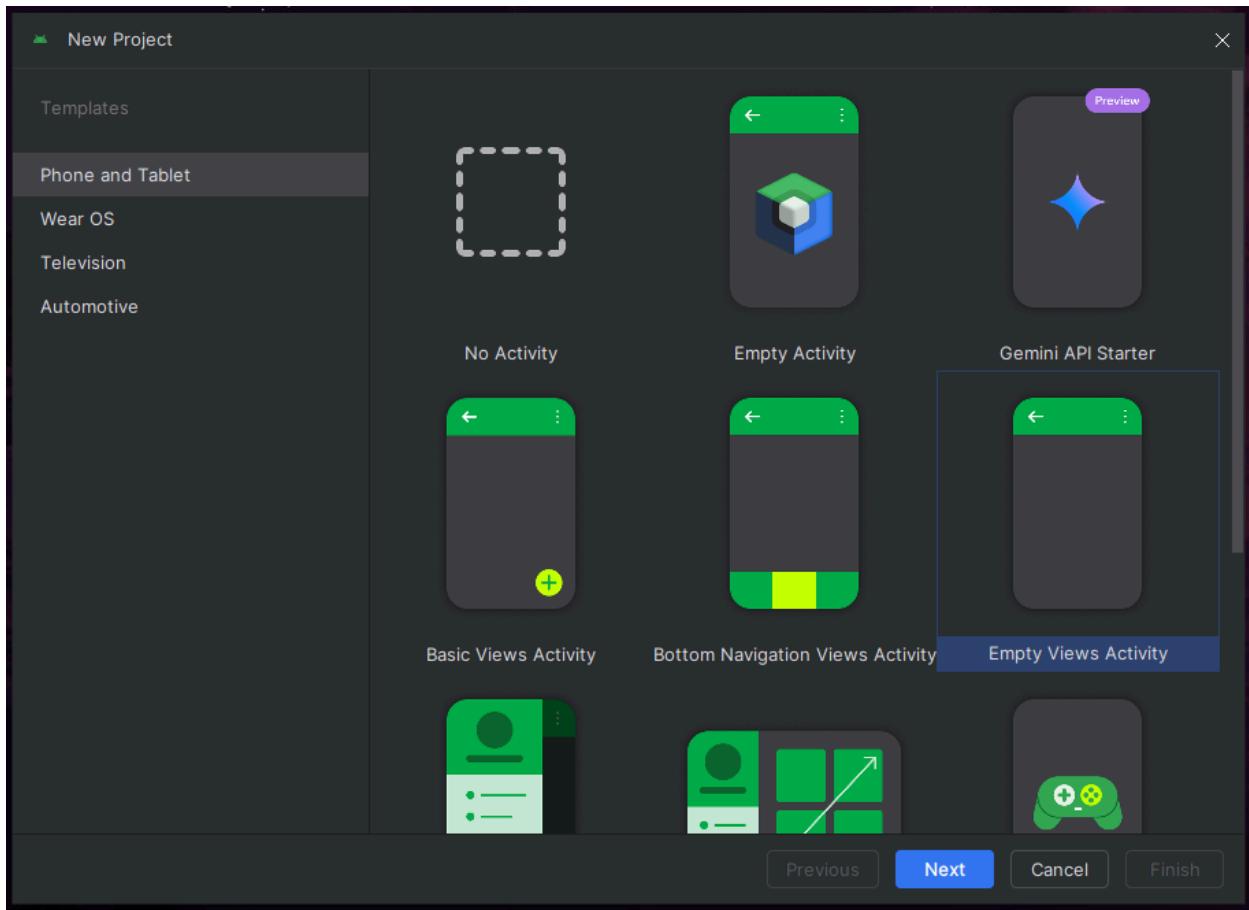


# Crear un proyecto en Java

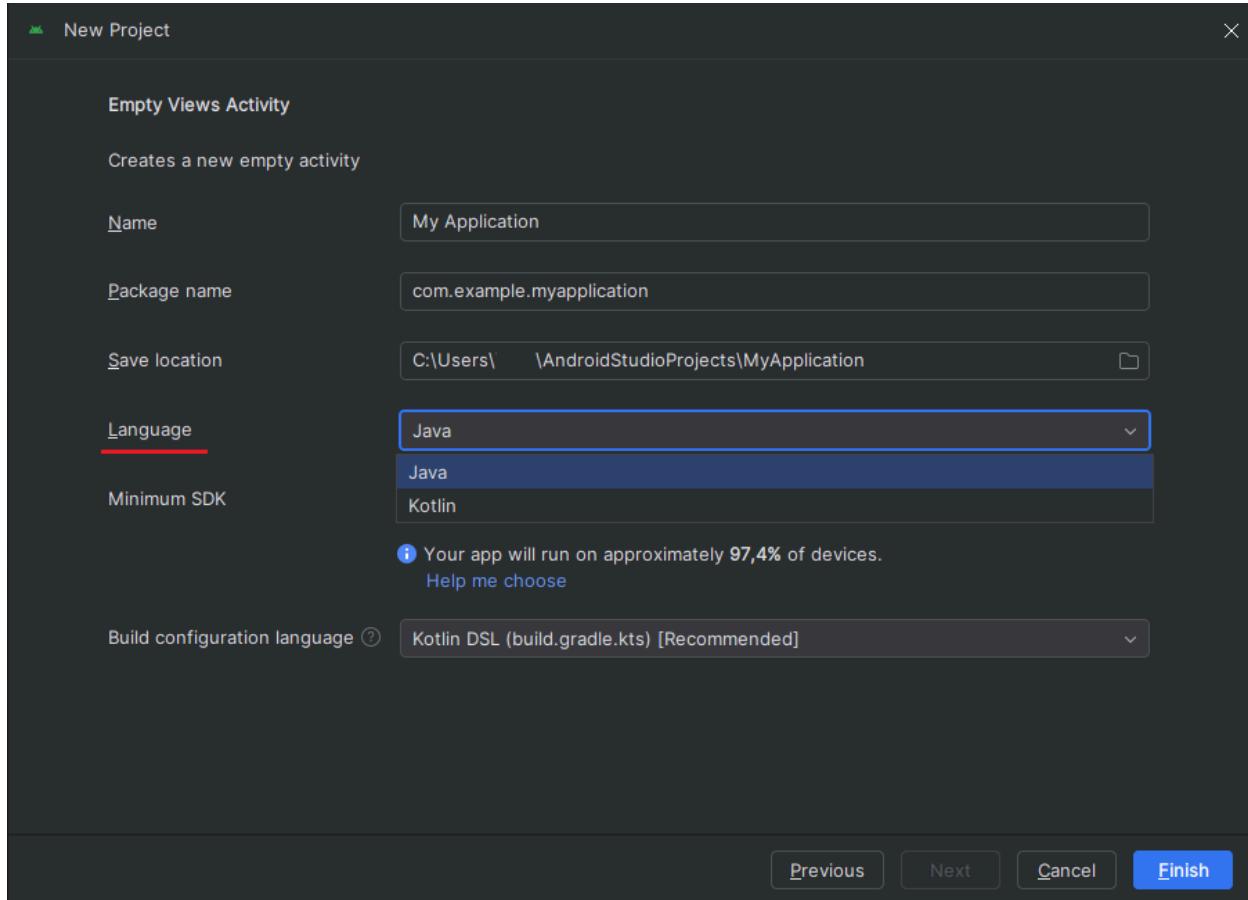
Cuando ya tenemos todo instalado y funcionando, procederemos a abrir android studio y seleccionar “New Project”



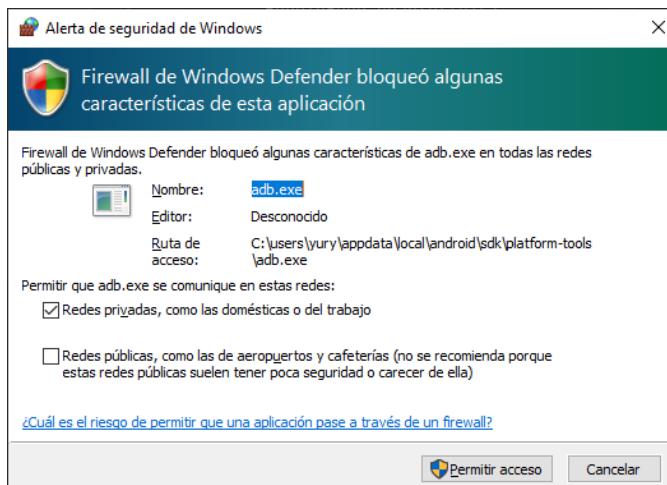
Para comenzar rápidamente un proyecto en java lo mejor es seleccionar la plantilla de proyecto “Empty Views Activity” y le haremos a “Next”



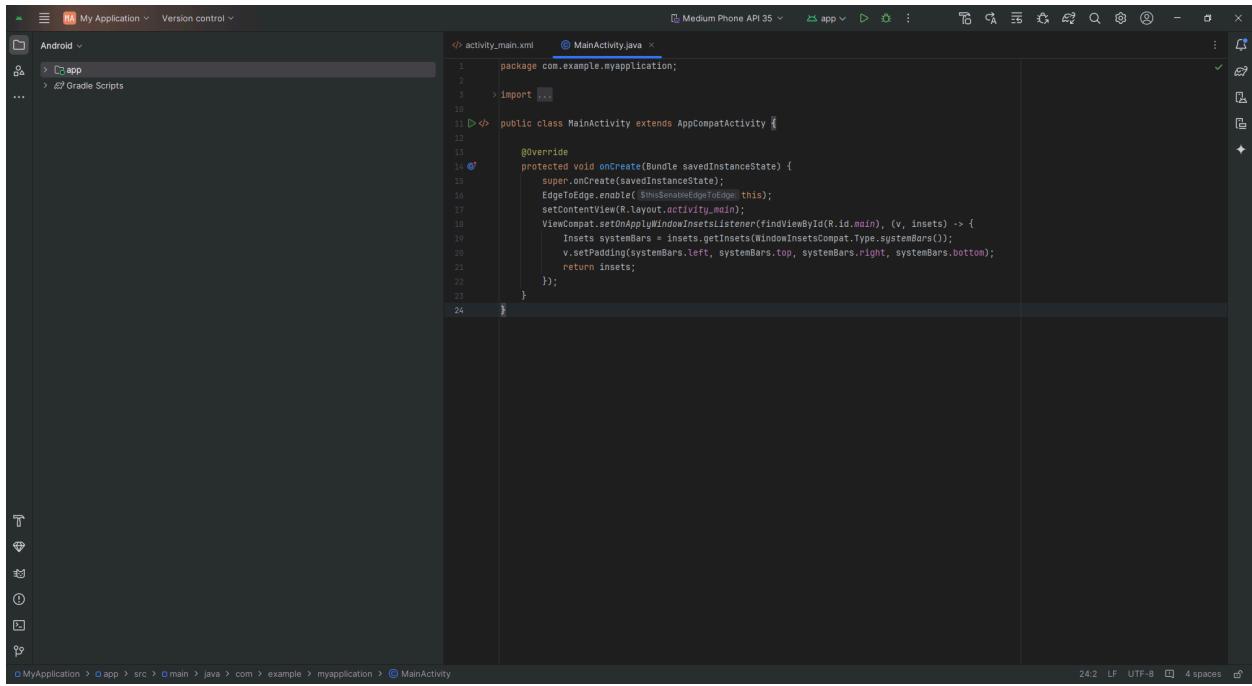
Aquí podremos cambiar en el apartado “Language” para que use Java en vez de Kotlin, darle un nombre a nuestro proyecto y al paquete, elegir la versión del SDK y la configuración de la compilación, la cual recomendamos dejar tal cual



Daremos a “Finish” y se nos abrirá esta ventana donde permitiremos el acceso a través del Firewall



Finalmente nos abrirá el IDE y cargará un proyecto plantilla de java



The screenshot shows the Android Studio interface with the following details:

- Title Bar:** My Application > Version control
- Side Navigation:** Android > app > Gradle Scripts
- Main Area:** Two tabs are open: activity\_main.xml and MainActivity.java. The MainActivity.java tab is active, displaying the following Java code:

```
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        EdgeToEdge.enable((WindowInsetsEdgeEffectManager) this);
11        setContentView(R.layout.activity_main);
12        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
13            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
14            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
15        });
16    }
17}
```

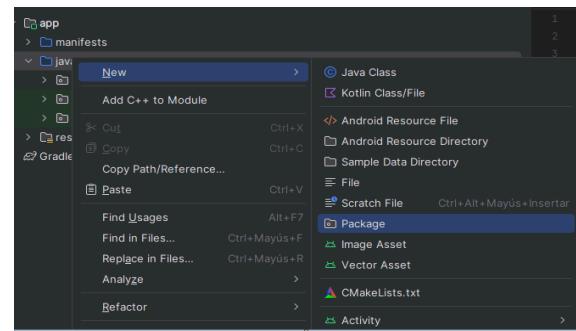
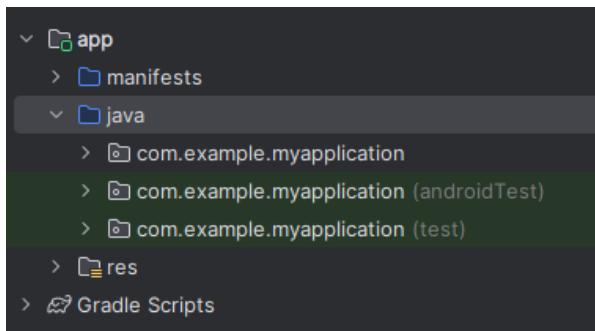
The code implements a custom window insets listener to handle system bars padding.

**Bottom Status Bar:** Shows the file path: MyApplication > app > src > main > java > com > example > myapplication > MainActivity, and the encoding: 24:2 LF UTF-8 4 spaces.

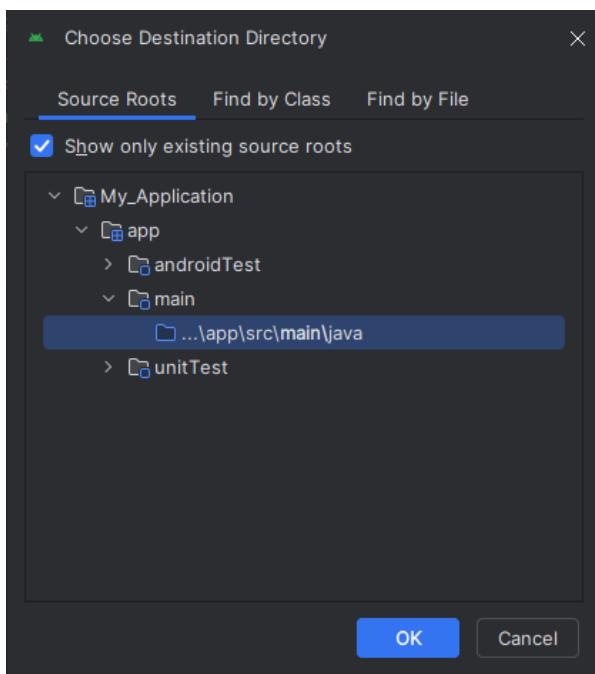
# Creación de Paquetes y Clases

## Creación de paquetes

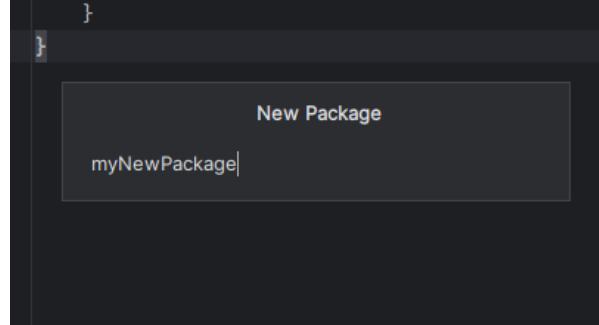
1. En el “File tree” desplegamos app > java
2. Luego damos click derecho donde esta el icono de carpeta que dice Java, nos desplazamos hasta “New” y luego “Package”



3. Aqui solo damos a”Ok”, ya que ya seleccionamos el destino en pasos anteriores

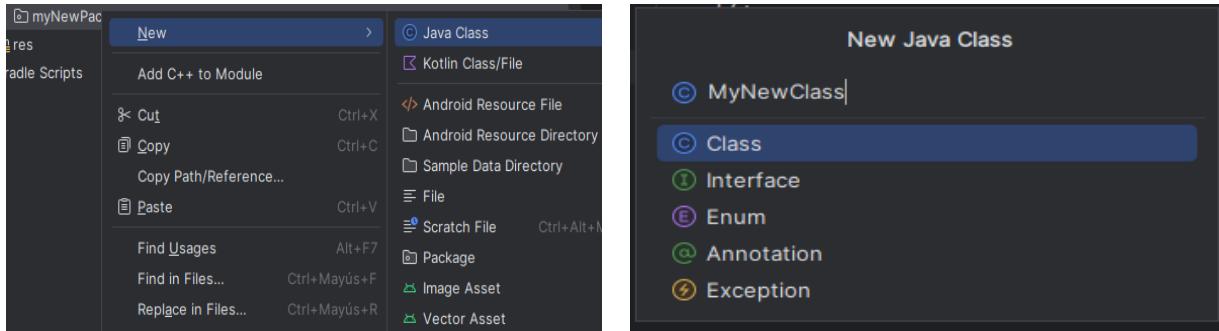


4. Damos nombre a nuestro paquete y presionamos la tecla Enter



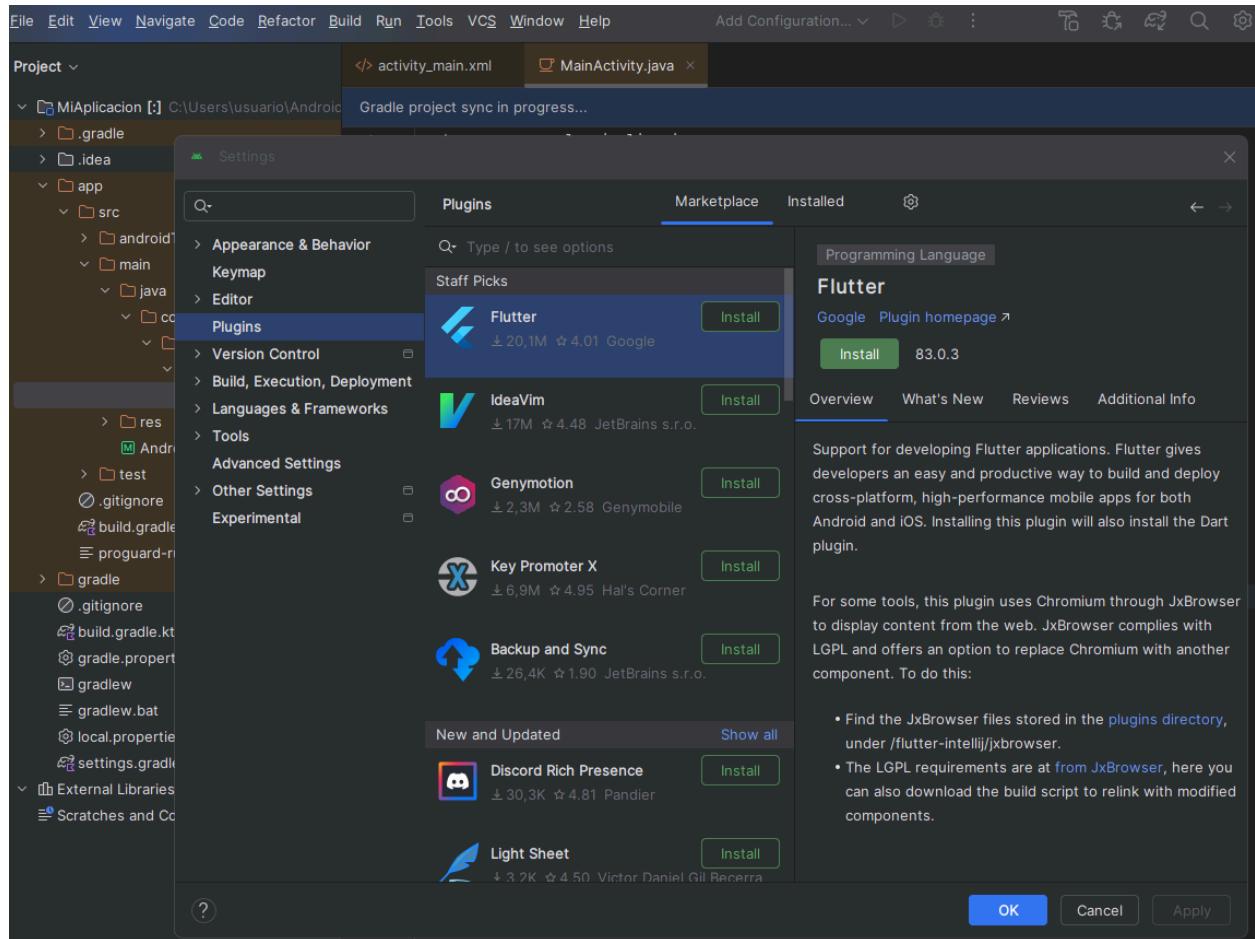
# Creación de Clases

Para crear una clase usaremos nuestro recién creado paquete al que daremos click derecho >Nuevo > Java Class, le daremos un nombre y presionaremos Enter

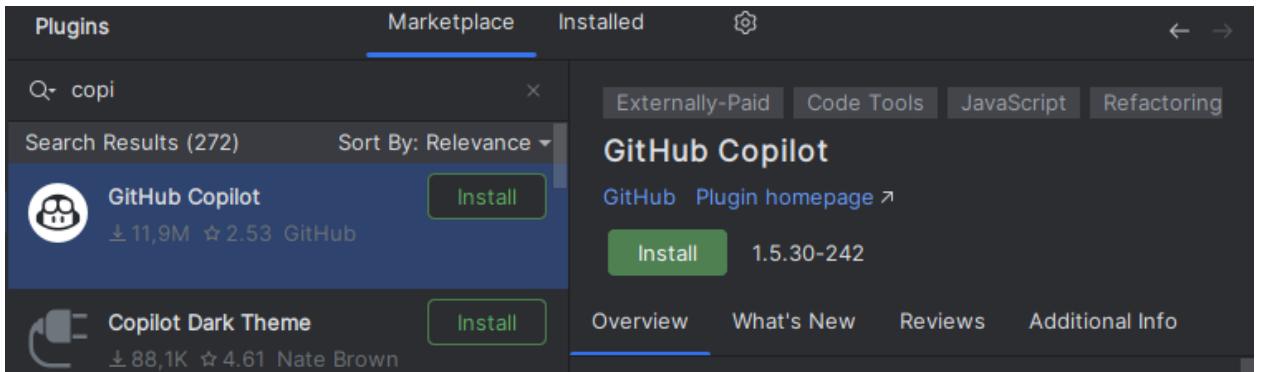


# Instalación de plugins

Una vez dentro para acceder al marketplace de plugins nos dirigimos a File > Settings > Plugins, ahora nos muestra una serie de plugins recomendados y a continuación la lista completa de plugins disponibles.



Integracion de copilot, en el buscador escribimos el nombre del plugin y presionamos en install y una vez instalado deberemos de reiniciar el IDE para que se complete la instalación.



Ahora deberemos de iniciar sesión en nuestra cuenta de github y autorizar el uso para poder usar copilot, al reiniciarse ya es la primera ventana que nos muestra el IDE.

