



DIRECCIÓN GENERAL DE EDUCACIÓN
TECNOLÓGICA INDUSTRIAL Y DE SERVICIOS

CBTIS 44

Manual de usuario de Android Studio.

***CBTis no.44.**

***Programación 6ºIº.**

***Desarrolla aplicaciones móviles para
Android.**

Desarrolla aplicaciones móviles para IOS.

***Peralta Tapia Manuel.**

***Marisela Hernández Morales.**

***Manual de usuario de Android Studio.**

Introducción

Android Studio es un entorno de desarrollo integrado (IDE) creado para que nosotros los programadores podamos desarrollar aplicaciones móviles en el SO Android, siendo esta una herramienta muy útil e intuitiva para desarrollar.

En esta plataforma ha sido muy intuitiva para programar, personalizar, emular, editar código, herramientas, depurar, siendo también muy importante su gran comunidad, para apoyarse, subir tutoriales, cursos, ya que esta al ser una plataforma bastante grande también mantenida por Google, el contenido y las actualizaciones siempre están a la orden del día.

Desarrollo

Instalación por pasos en Windows 10:

1:ingresa a <https://developer.android.com/studio>

The image shows the Android Studio website and the IDE interface. The website header includes the Android logo, 'Developers', and navigation links like 'Imprescindibles', 'Desarrollo', 'Google Play', and 'Más'. A search bar and language selector (Español - A...) are also present. The main content area features a 'Descargar' button and links to guides. A 'Switch to English' button is visible. The IDE interface shows a code editor with Kotlin code for a MultiPreview app, a preview window, and a chatbot named 'Hello, Android Developer'.

Android Studio

El IDE oficial para el desarrollo de apps para Android ahora acelera tu productividad con Gemini en Android Studio, tu asistente de programación potenciado por IA.

[Descargar Android Studio Meerkat](#)

[Leer las notas de la versión](#)

```
MultiPreview
@Composable
fun MultiPreviewPreview(
    @PreviewParameter(UserNewsResourceProvider::class)
    userNewsResource: List<UserNewsResource>,
) {
    MultiTheme {
        Scaffold(
            topBar = {
                MultiAppBar(
                    title = "R.string.app_name",
                    navigationIcon = MultiIcons.Search,
                    navigationContentDescription = "Navigation icon",
                    actionIcon = MultiIcons.Hamburger,
                    actionContentDescription = "Action icon",
                )
            },
            contentWindowInsets = WindowInsets(0, 0, 0, 0)
        ) {
            // Content
        }
    }
}

@Preview
@Composable
fun MultiPreviewPreviewPreview() {
    MultiPreviewPreview(
        userNewsResource = listOf(
            UserNewsResource(1, "News 1", "Description 1"),
            UserNewsResource(2, "News 2", "Description 2"),
            UserNewsResource(3, "News 3", "Description 3"),
            UserNewsResource(4, "News 4", "Description 4"),
            UserNewsResource(5, "News 5", "Description 5"),
            UserNewsResource(6, "News 6", "Description 6"),
            UserNewsResource(7, "News 7", "Description 7"),
            UserNewsResource(8, "News 8", "Description 8"),
            UserNewsResource(9, "News 9", "Description 9"),
            UserNewsResource(10, "News 10", "Description 10"),
        )
    )
}
```

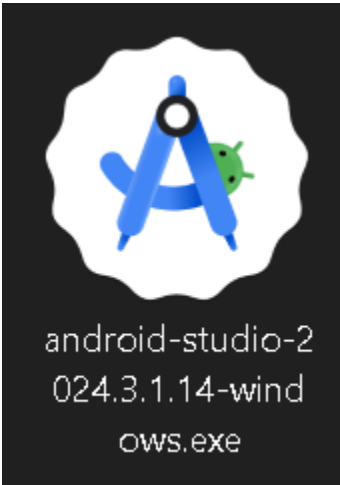
App Quality insights

Issue	Severity	Count	First seen	Last seen	Impact
android.os.MessageQueue: nativeMessageQueue	High	1	10/10/2023	10/10/2023	100%
The ACTION_MOVE event was not cleared	Medium	1	10/10/2023	10/10/2023	100%
The ACTION_MOVE event was not cleared	Medium	1	10/10/2023	10/10/2023	100%

2: Baja hasta encontrar los instaladores y descárgalo dependiendo de tu SO en este caso Windows.

Plataforma	Paquete de Android Studio	Tamaño	Suma de comprobación SHA-256
Windows (64 bits)	android-studio-2024.3.1.14-windows.exe Recomendado	1.3 GB	0291303c541d0313d2d31b6bf7ad8de3df34a51604bdb273c1ae75f6fb793fe0
Windows (64 bits)	android-studio-2024.3.1.14-windows.zip Sin instalador .exe	1.3 GB	78e1cc774a9771dd632f10caaf22109baae43fcd571d3a48feb5cb865c2db05
Mac (64 bits)	android-studio-2024.3.1.14-mac.dmg	1.4 GB	a1fc20c46a3d7dc3692ff48827d9dcabfde1b59ac14cf686adb51b7c455f9167
Mac (64 bits, ARM)	android-studio-2024.3.1.14-mac_arm.dmg	1.3 GB	89f55dd8b329fa28c61f9635bafc10cfe52055772c8a9721719c54a0e403cfee
Linux (64 bits)	android-studio-2024.3.1.14-linux.tar.gz	1.4 GB	54d5c4adf6f83e1963709c06abceb795d87fde2f1f31d262ae5c28944224f9f2
ChromeOS	android-studio-2024.3.1.14-cros.deb	1.1 GB	d1379c25b597a2d84130ab6fdb5f82921d5fc353e92a21c5fd5dcb6a195f5891

3: Ejecuta el instalador.

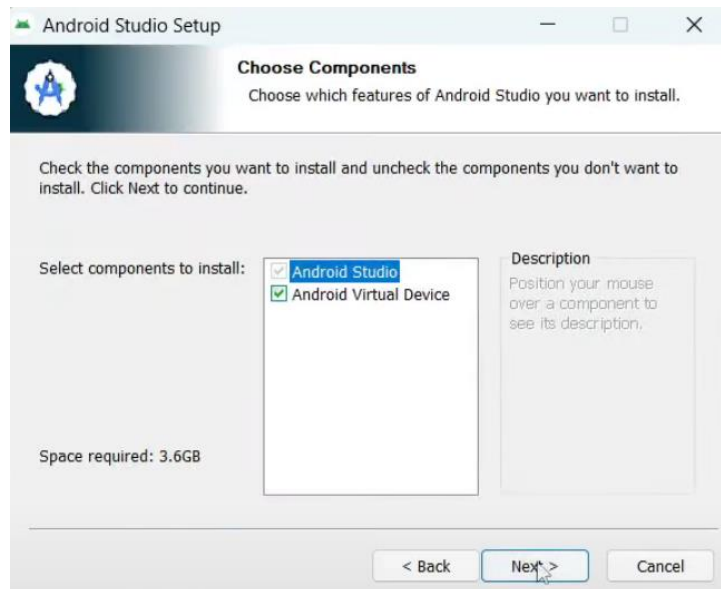


4: Puede aparecer una pantalla emergente donde te pregunte si quieres que Android Studio realice cambios en tu dispositivo, donde presionaras si.

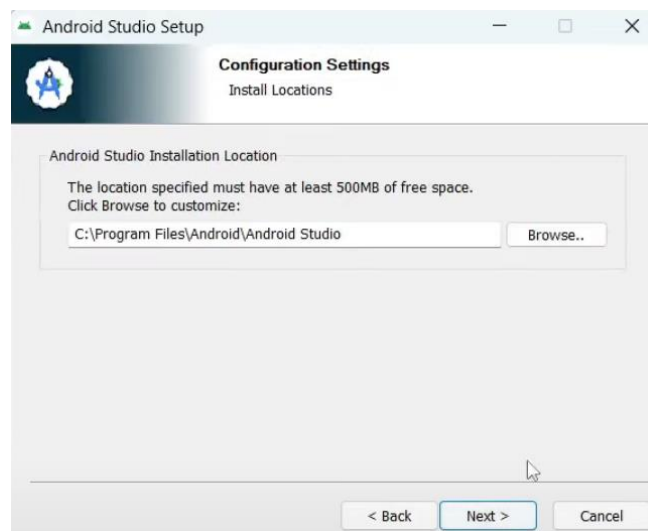
5: En la segunda pantalla presionaras next.



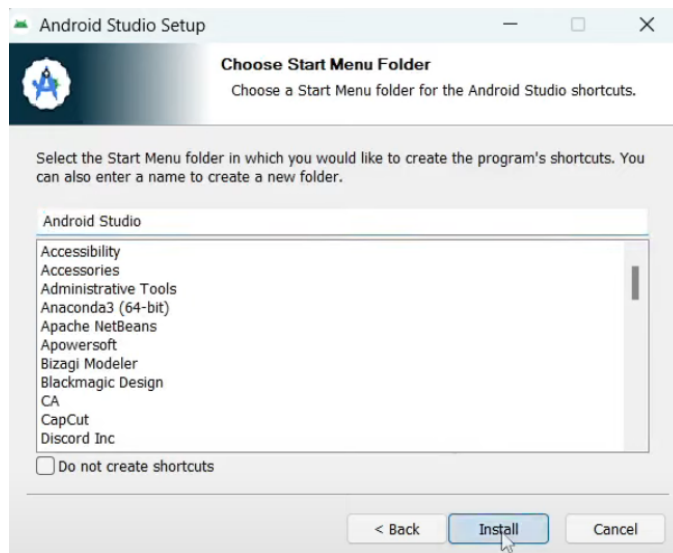
6: También en la siguiente pantalla.



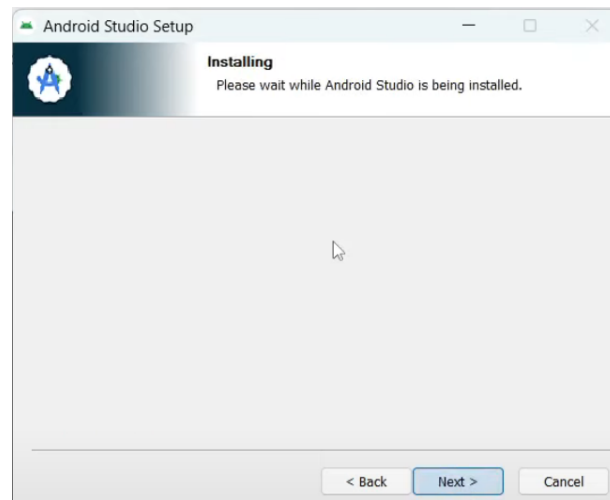
7: También en la siguiente pantalla (si es que quieres instalar el Android Studio en tu disco principal o en una carpeta en específico).



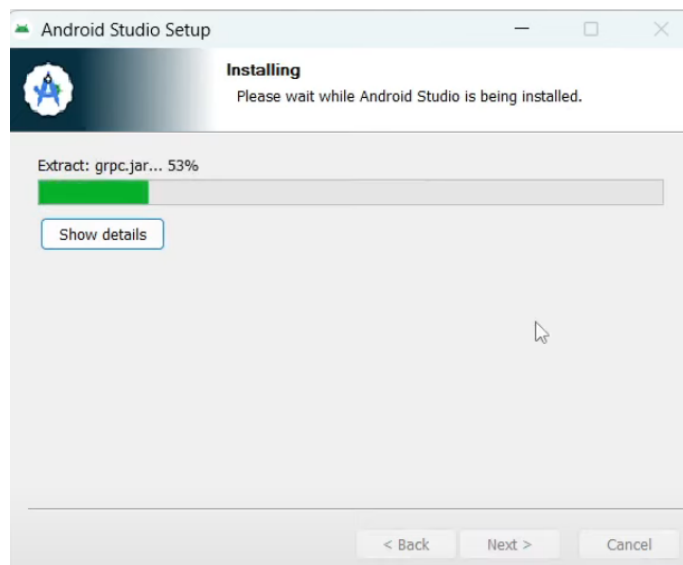
8: Presionaras Install.



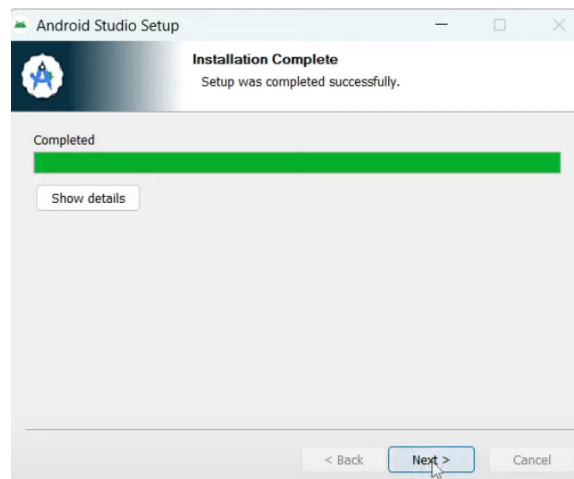
9: Presionaras next para que empiece a instalar.



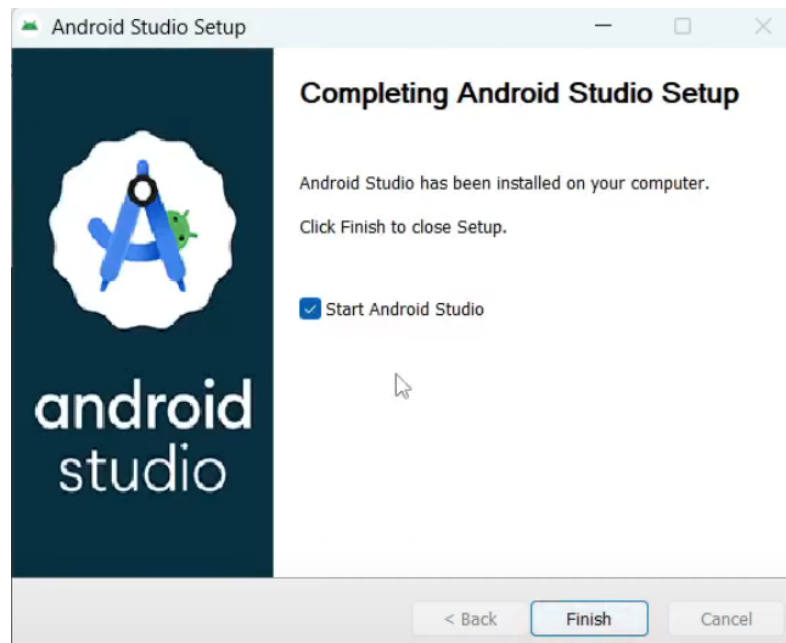
10: Espera a que termine de instalarlo.



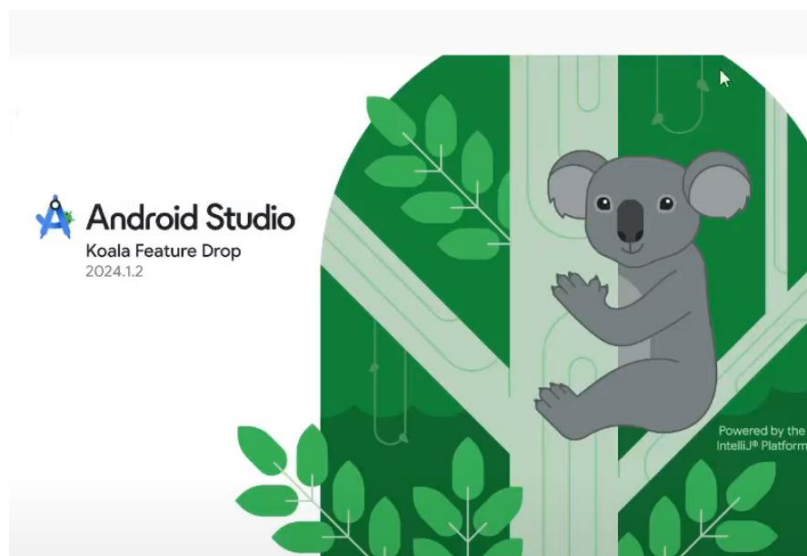
11: Cuando termine la instalación presionar next.



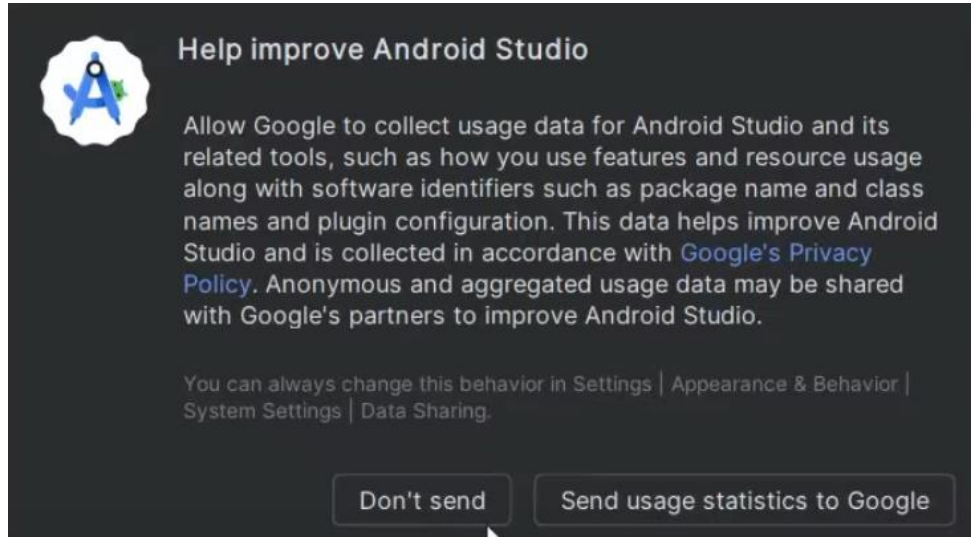
12: presionar finish con la casilla Start Android Studio para ejecutarlo.



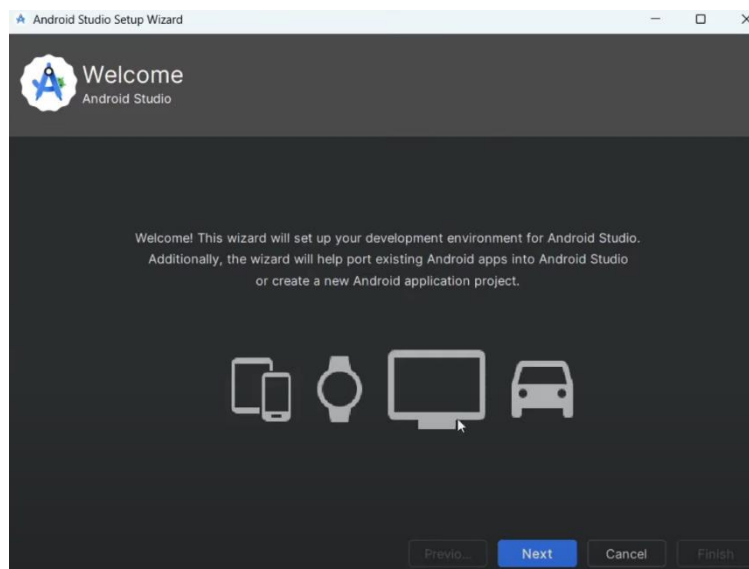
13: Al ejecutarlo aparecerá esta pantalla (o de algún otro animal).



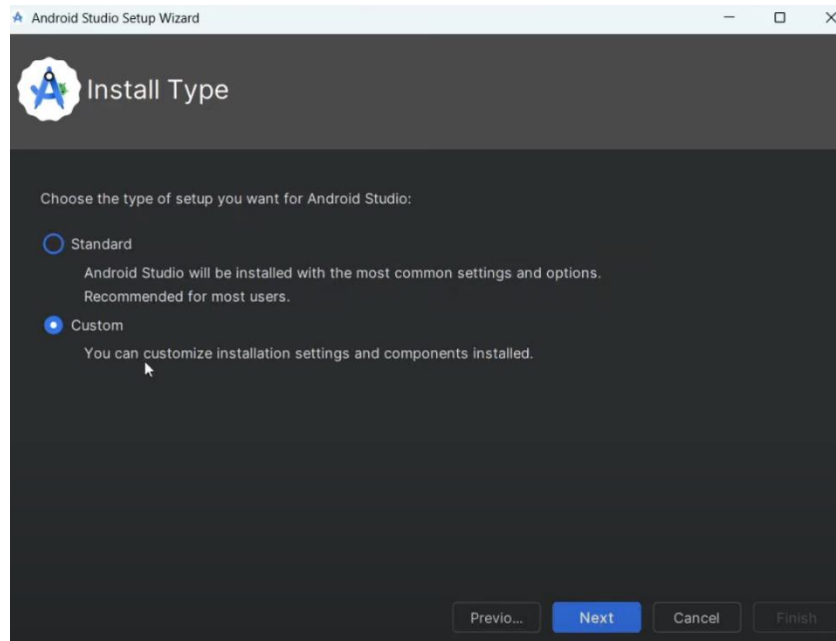
14: Aparecerá una pantalla oscura que dice Help improve Android Studio, es para enviarle estadísticas a Google, esto dependerá de cada uno, pero personalmente presiono “Don’t send”.



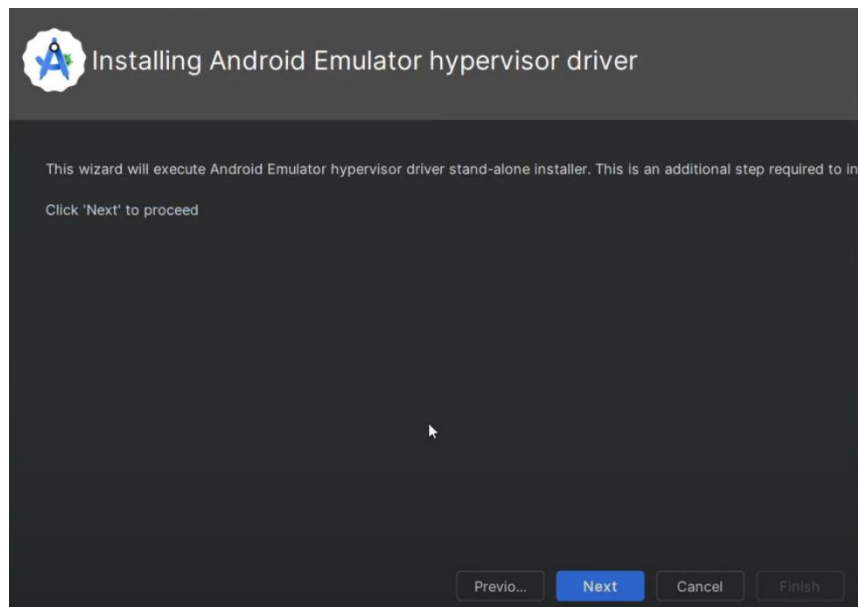
15: Presionaras next.



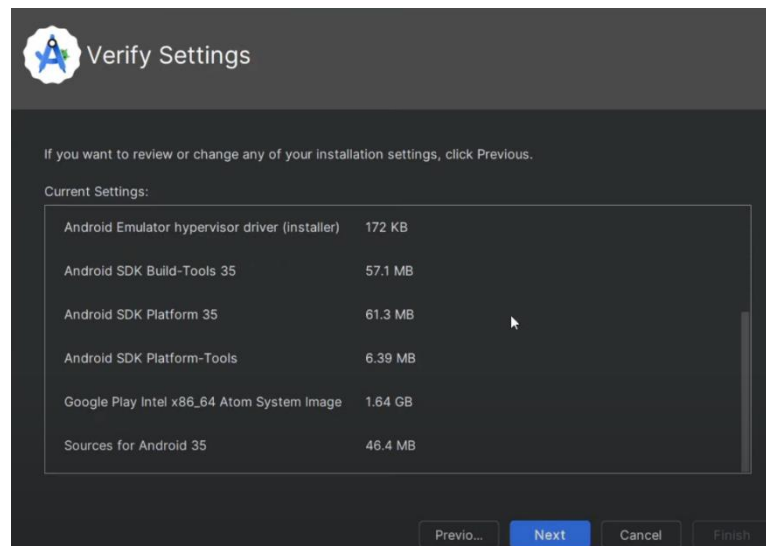
16: En esta pantalla hay dos opciones, si instalar Android Studio de manera estándar o custom, yo utilizo la forma estándar ya que prefiero instalar todo el contenido de esta app, y en la opción custom tu eliges que instalar de la aplicación.



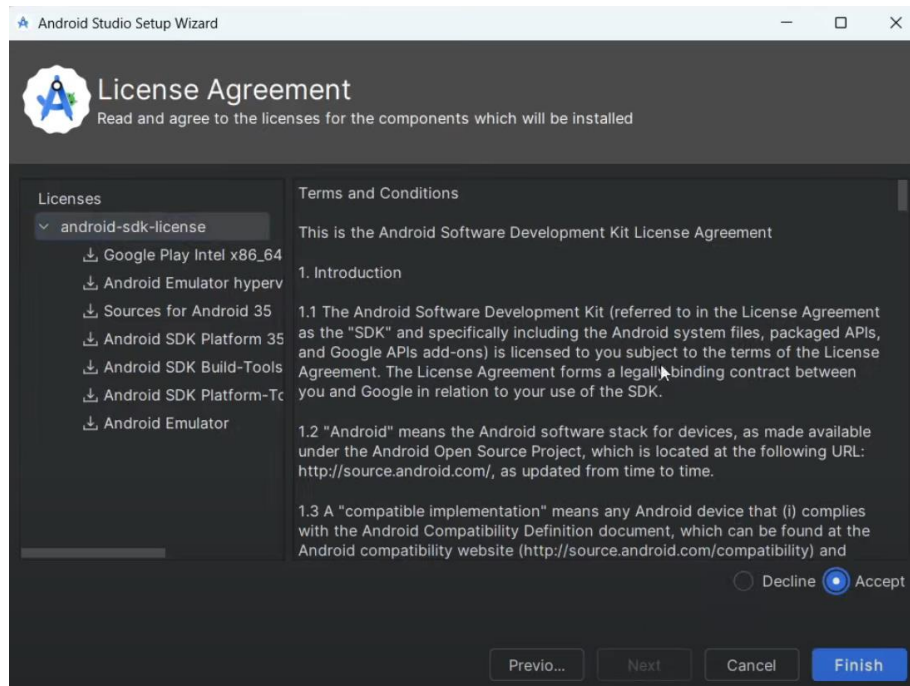
17: presionar next.



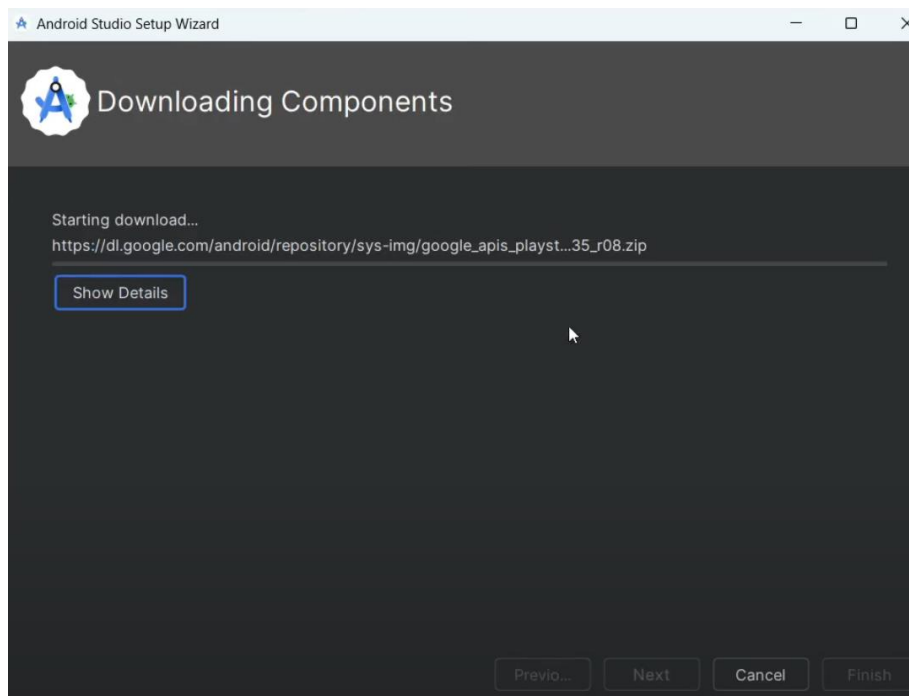
18: presionar next.



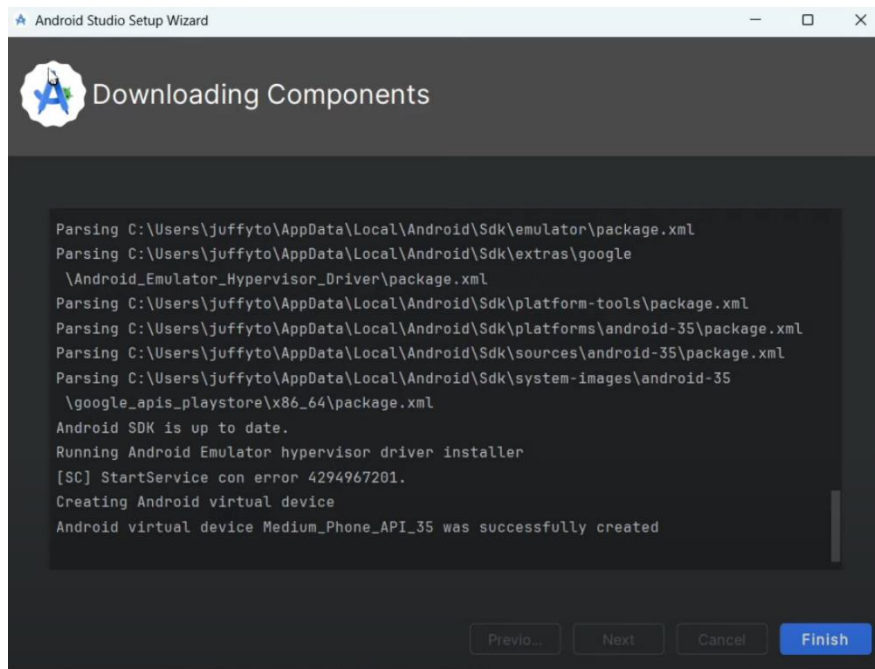
19: Presionar la casilla accept para aceptar los términos y condiciones y luego finish.



20: Descargara todos los componentes faltantes.

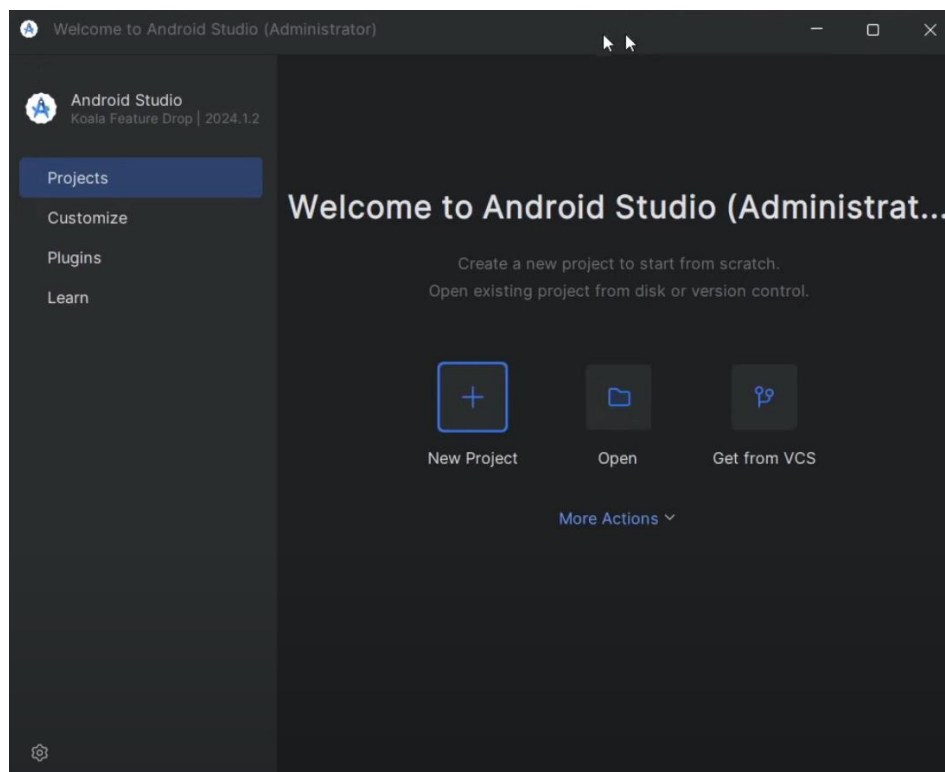


21: Presionar finish para completar la instalación y ejecutar Android Studio.

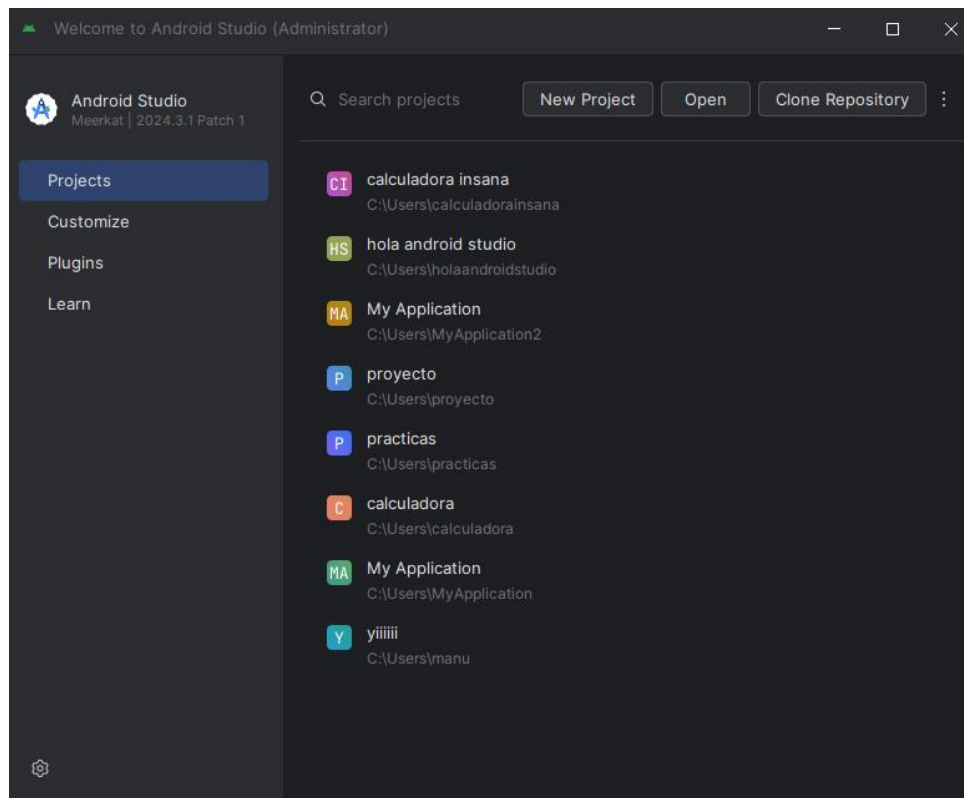


Primeras vistas en Android Studio

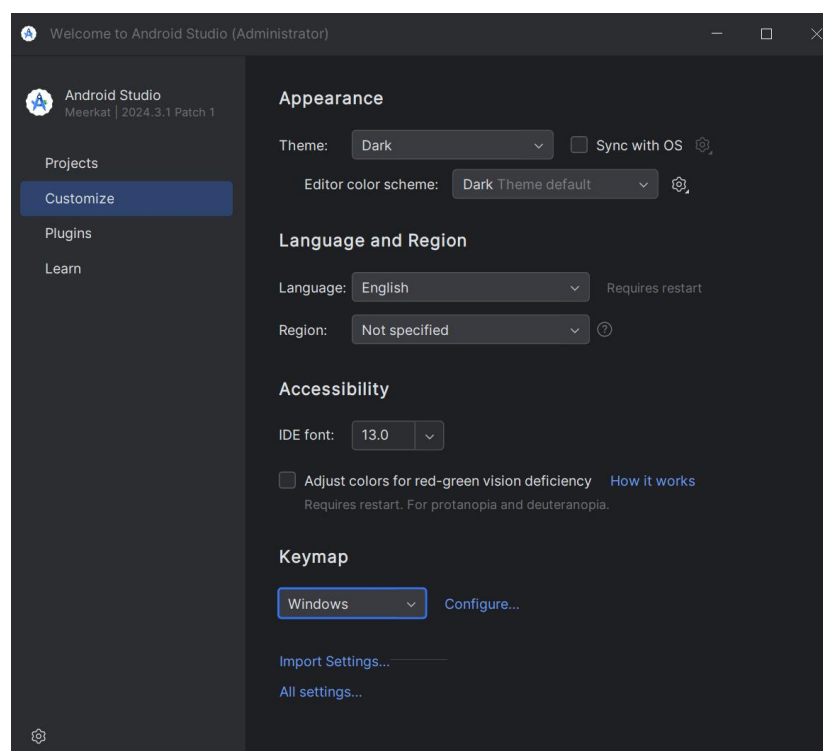
Al ingresar por primera vez a Android Studio aparecerá una ventana donde estarás en el apartado de proyectos, donde podrás crear y abrir proyectos.



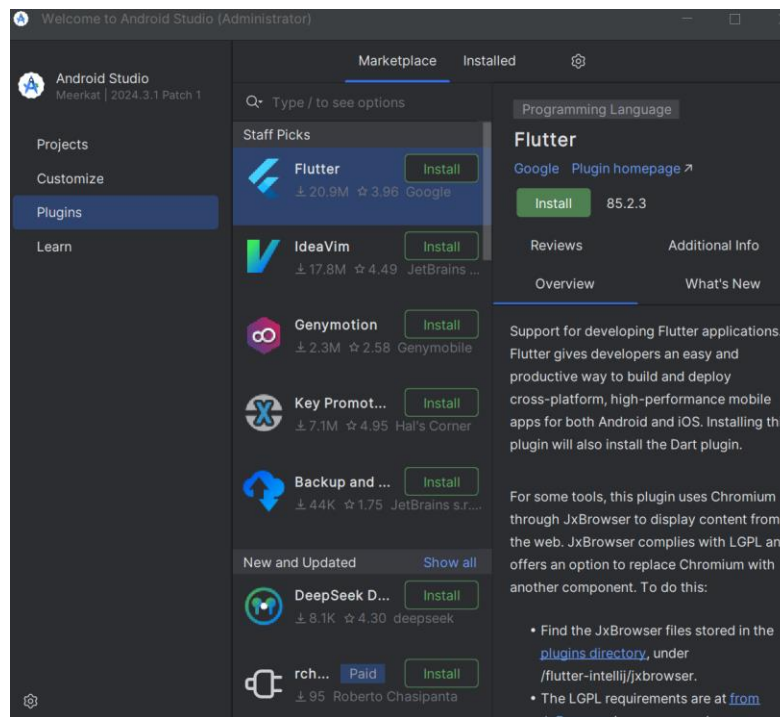
Si ya tienes proyectos creados, o abiertos recientemente cambiara la perspectiva.



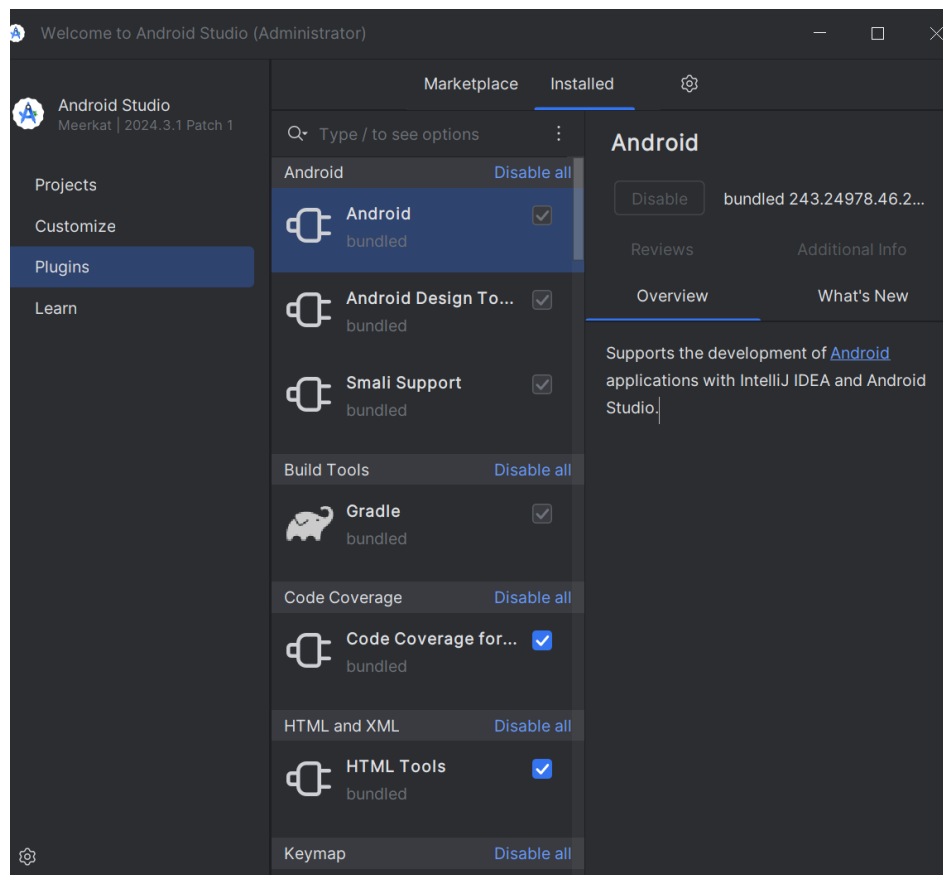
En el apartado de Customize se puede configurar el aspecto visual de Android Studio como Apperance que es como queremos que se vea Android Studio, sus colores o tonos si obscuro o blanco, en Language and Region, en que idioma queremos que este y su región o país, Accessibility puedes configurar el tamaño de la fuente del texto, y Keymap sirve para ver, modificar, asignar, e importar atajos de teclado.



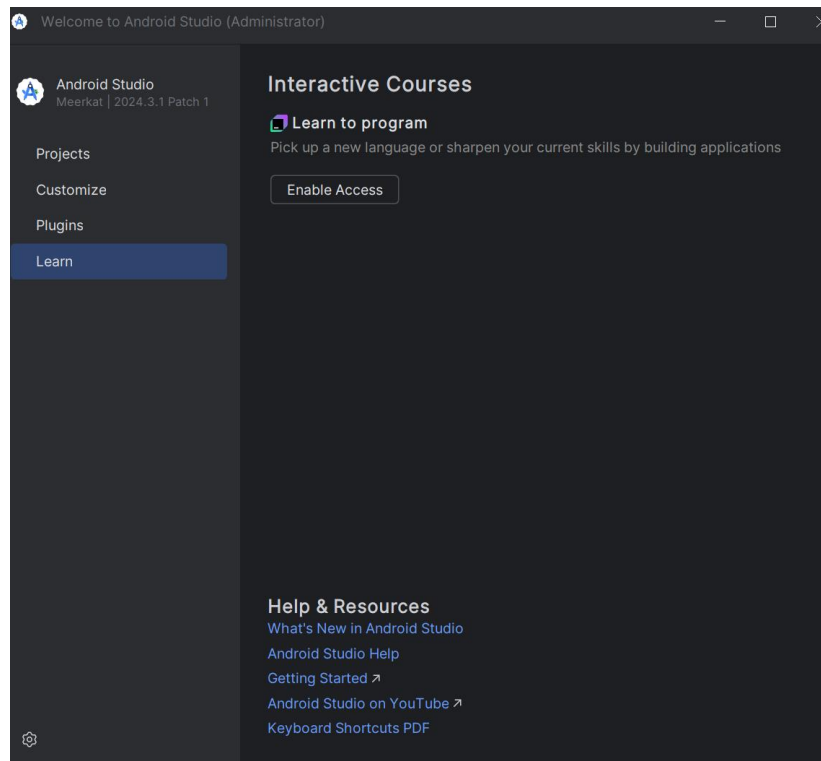
En el apartado de pluggins se abrirán por así decirlo dos “pestañas” la primera es por así decirlo una Play Store de extensiones de Android Studio como decoraciones, herramientas, lenguajes de programación y mejoras para estos, así como una descripción del pluggins, opiniones, que mejoraron, etc.



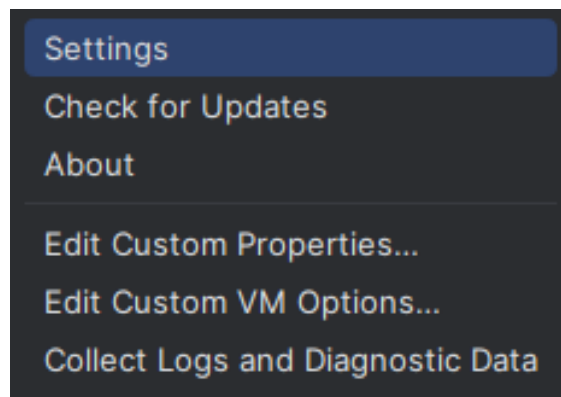
Y en la otra pestaña se encuentran los pluggins que ya tienes instalados.



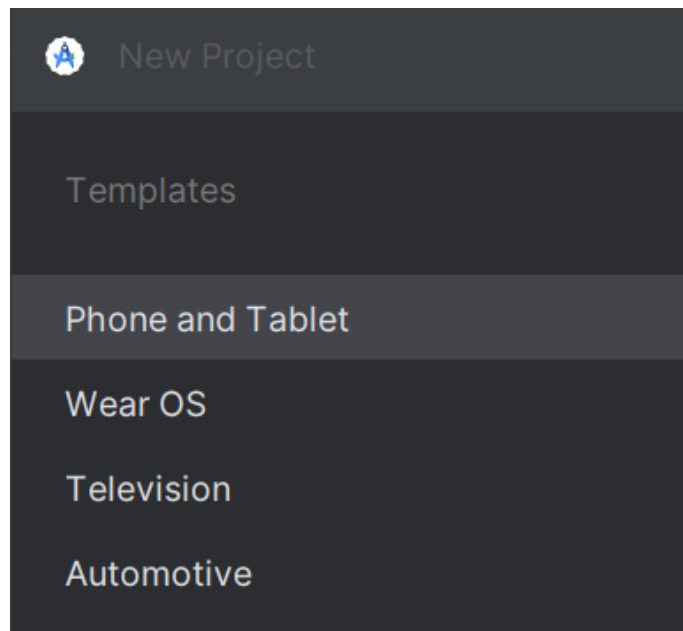
En el apartado Learn, hay un botón para descargar un curso interactivo, y en la parte de abajo, links para ver que hay de nuevo en Android Studio, ayuda, el canal de YouTube, si apenas estoy iniciando, etc.



En la esquina inferior izquierda, hay otro apartado el cual es un engranaje que al poner el cursor sobre este despliega opciones para configurar, **la primera** es la configuración general de todo, **la segunda** realiza un chequeo de las actualizaciones para comprobar que tengas la versión mas reciente, **la tercera** muestra información general como su versión, información del sistema, **la cuarta** editar el archivo “idea.properties” que puedes configurar el ide, como el tamaño del historial de archivos, cache, **la quinta** permite editar las opciones de la maquina virtual de java, como asignar más o menos memoria, **la sexta** recopila logs del sistema, configuración del IDE en un archivo comprimido.

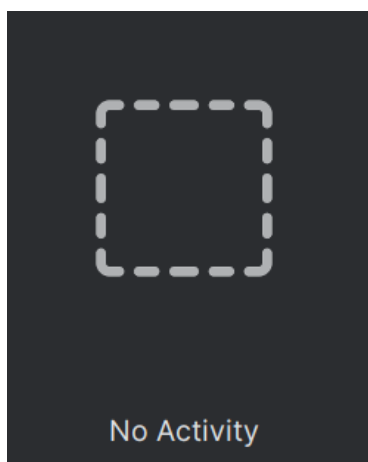


Regresando al apartado para iniciar a hacer proyectos, al momento de presionar New Project, se abrirá una pestaña donde, el lado izquierdo se abrirá por así decirlo los perfiles o entornos en los que se podrá ejecutar la app, puede ser en teléfono y tableta, reloj inteligente, televisión, o pantalla de un automóvil.

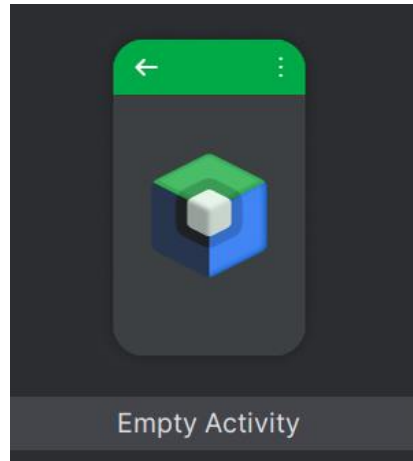


Pero regresando en el apartado de teléfono y Tablet, que es el mas completo en realizar apps, tenemos varias opciones o perfiles, donde cada uno te da diferentes “Herramientas precargadas”.

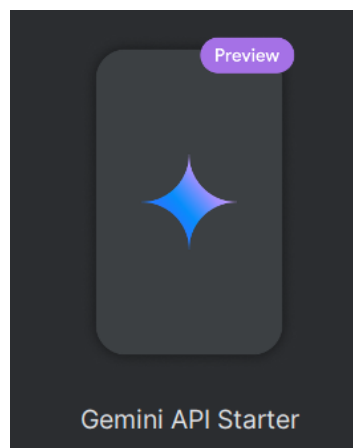
Crea un proyecto **vacío sin ninguna actividad (activity)** inicial, para algo totalmente personalizado.



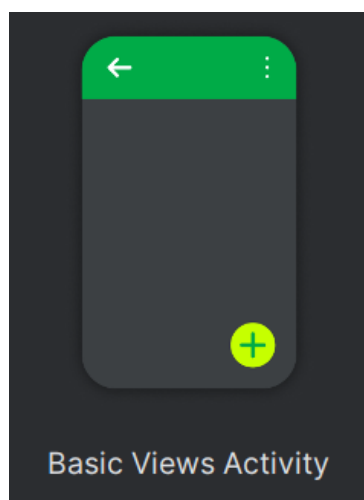
Crea un proyecto **vacío sin ninguna actividad (activity)** inicial, si quieres una estructura simple pero lista para correr.



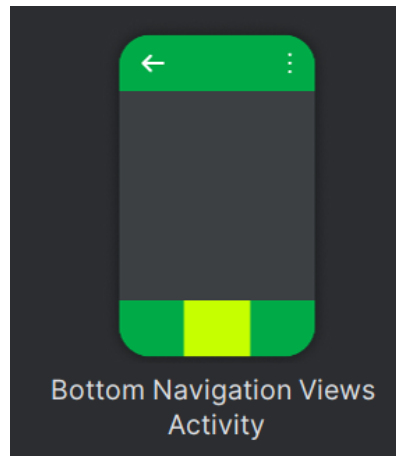
Plantilla para usar la API de Gemini (IA de Google), Viene con código inicial para conectar con Gemini y hacer peticiones, **Ideal para:** apps que integran IA generativa o procesamiento de lenguaje natural.



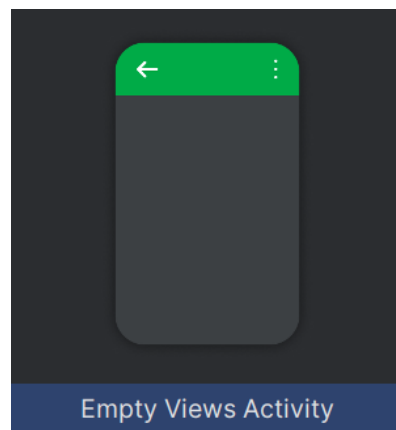
Actividad básica con **Componentes Views (XML tradicionales)** incluye AppBar y botón flotante (FAB), **ideal para:** apps con interfaz tradicional (no Jetpack Compose), con elementos comunes listos.



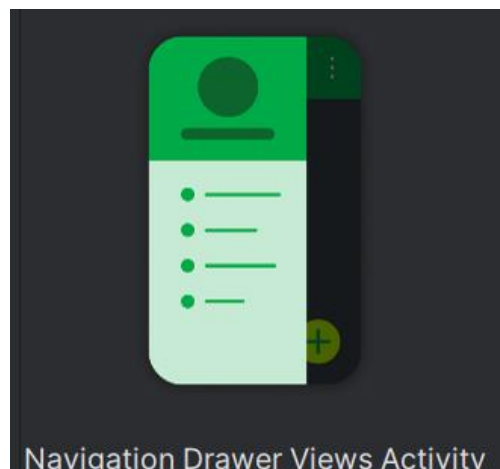
Actividad con **barra de navegación inferior** (BottomNavigationView) usa fragmentos y arquitectura de navegación con **Views (XML)** **ideal para:** apps que requieren navegación entre secciones/pestañas (ej. perfil, inicio, configuración).



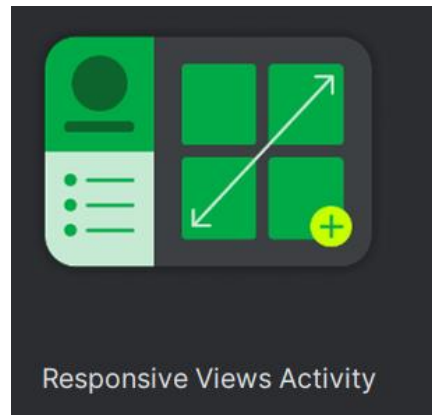
Igual a Empty Activity pero usando **Views en XML** (no Jetpack Compose) incluye archivos base MainActivity y activity_main.xml **ideal para:** desarrolladores que prefieren XML sobre Compose.



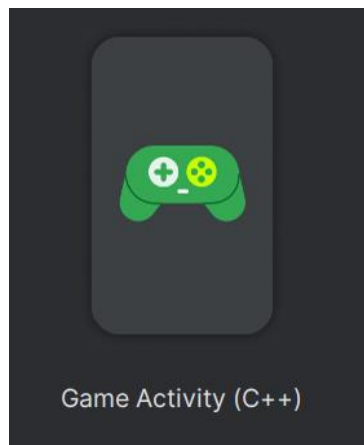
Actividad con **menú lateral deslizable (drawer)** configurado con NavigationView, DrawerLayout y fragmentos. **ideal para:** apps con muchas secciones organizadas desde el menú (ej. Gmail, apps de empresa).



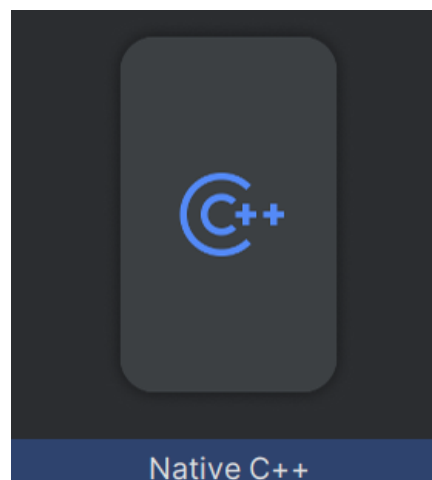
Diseñada para **adaptarse a distintos tamaños de pantalla** (celular, tablet, etc.) usa vistas y fragmentos dispuestos según el espacio disponible **ideal para:** apps que deben funcionar bien en tablets o pantallas grandes.



Proyecto inicial para juegos con **soporte C++ y OpenGL/NDK** viene con bindings y archivos .cpp MainActivity + **ideal para:** desarrolladores de videojuegos nativos o motores como SDL, OpenGL, etc.

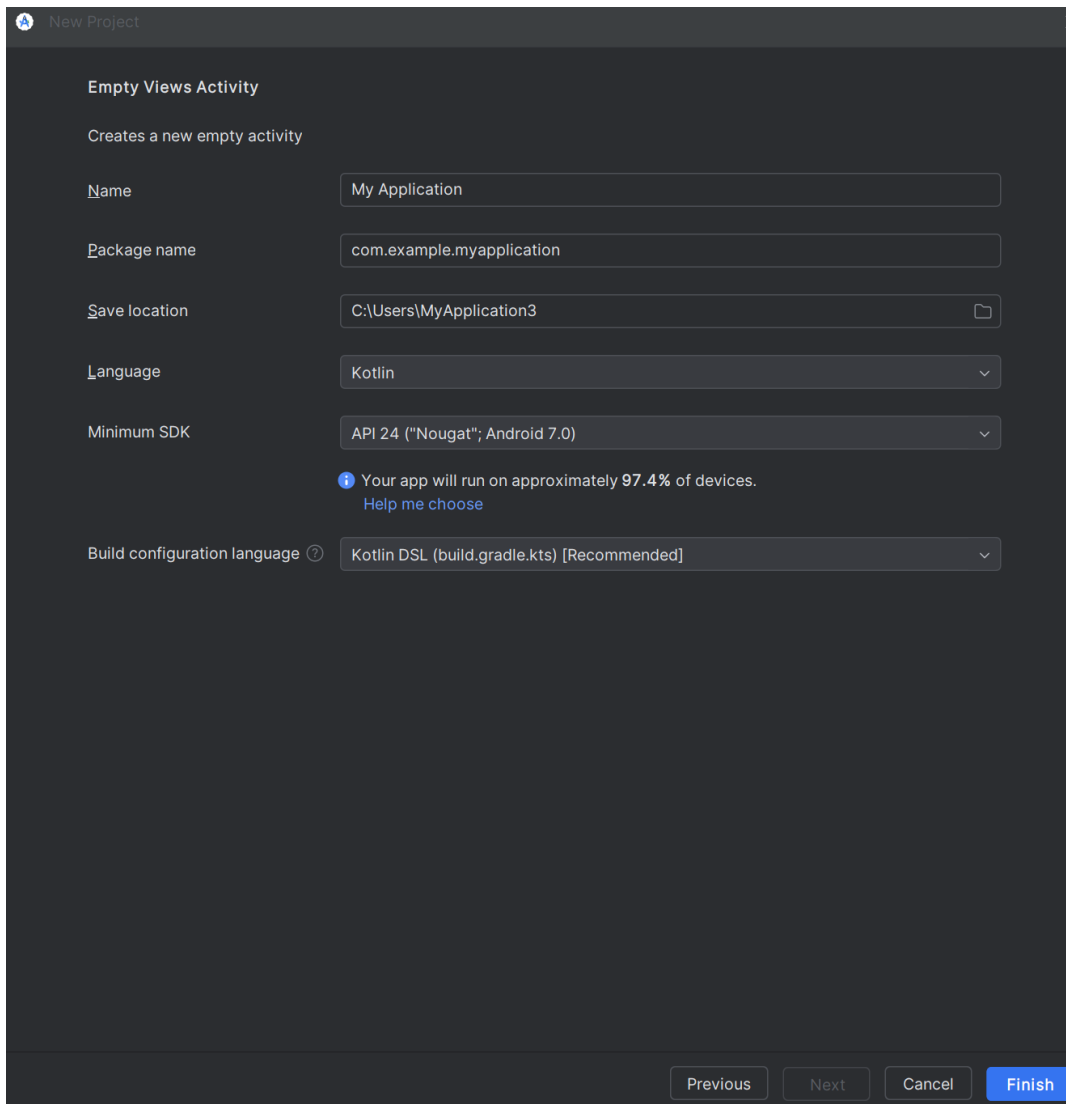


Proyecto mínimo con C++ (NDK) pero sin plantillas visuales ni lógica preconstruida integra CMake y JNI para llamadas entre Java y C++ **ideal para:** apps con procesamiento nativo, rendimiento alto o integraciones específicas.

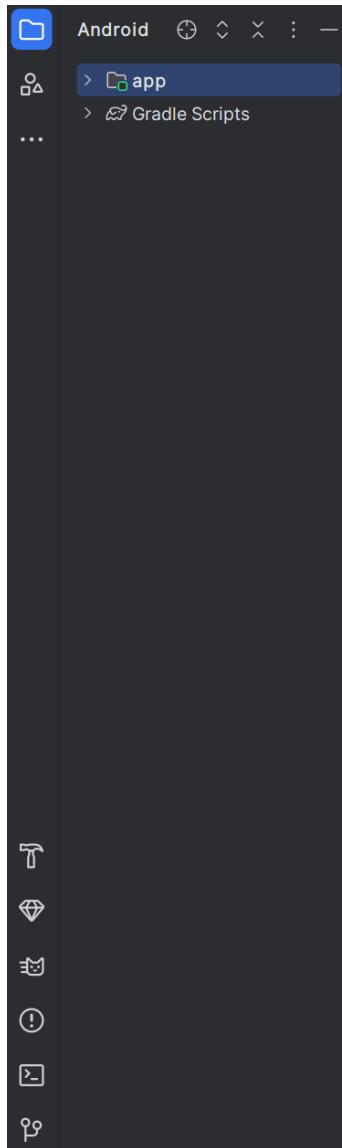


Vista de programador

En este caso elegiremos las opciones Empty Views Activity ya que personalmente siento que es la mas completa y la que me recomendaron, para tener una vista para programar, se abrirá una ventana donde elegiremos el nombre del proyecto, su ruta de guardado, el lenguaje de programación, la mínima versión de Android, y la configuración del lenguaje.



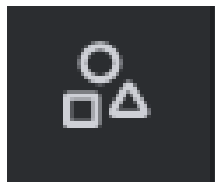
Se abre el código y muchas opciones que las dividiré por partes, primero toda la parte izquierda.



Muestra la estructura de carpetas de tu proyecto puedes cambiar entre diferentes vistas como: Android (vista simplificada), Project (estructura real de archivos)



Muestra todos los recursos de tu app: imágenes (drawables), colores, strings, temas, layouts, etc. te permite importar y organizar recursos visualmente.



Muestra el estado de la compilación (build) del proyecto aquí puedes ver errores, advertencias o logs después de compilar.



Ayudas del propio Google como una consola.



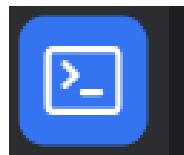
Muestra la salida de la ejecución de tu app o los mensajes del sistema (Logcat) también puedes colocar puntos de interrupción (breakpoints) y analizar paso a paso el código mientras se ejecuta.



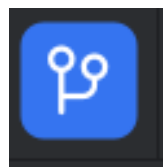
Lista errores y advertencias detectados en el código (incluso antes de compilar) también sugiere correcciones rápidas.



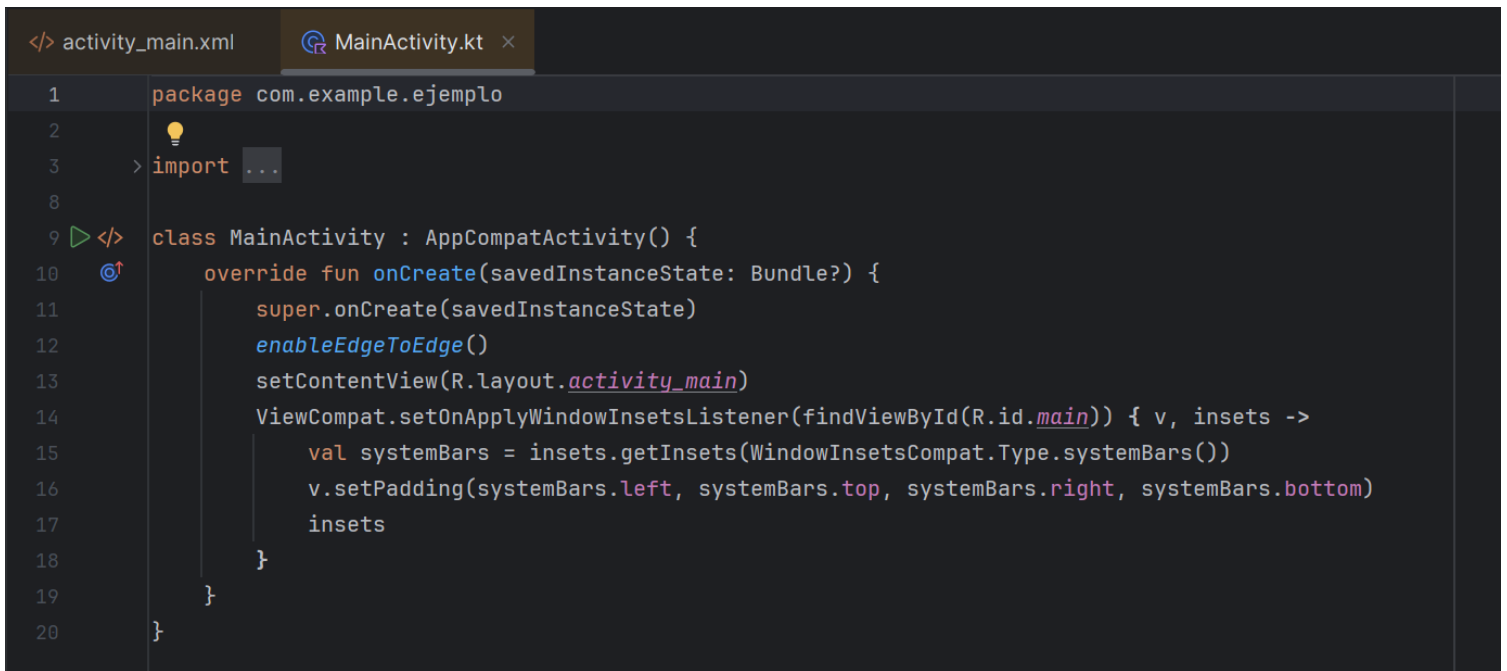
Consola para ejecutar comandos directamente (como git, gradlew, etc.) es el equivalente a una terminal de tu sistema operativo, pero integrada.



Se conecta con sistemas de control de versiones como **Git** puedes ver cambios, hacer commits, comparar ramas, hacer push/pull, etc.

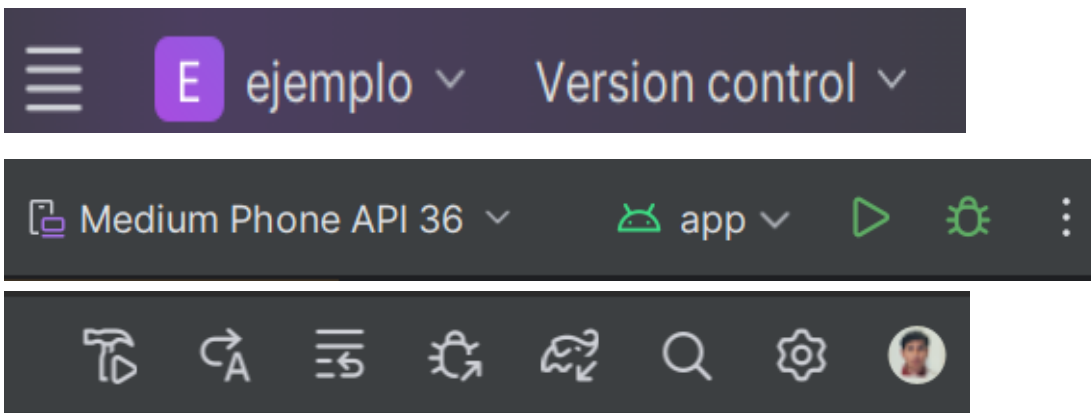


En el centro se encuentran los archivos abiertos y el código de estos, en este no hay mucho que decir solo que, puedes cerrar el archivo si es necesario.

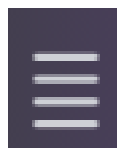


```
</> activity_main.xml  MainActivity.kt x
1  package com.example.ejemplo
2
3  > import ...
4
5
6
7
8
9  </> class MainActivity : AppCompatActivity() {
10  @<@> override fun onCreate(savedInstanceState: Bundle?) {
11      super.onCreate(savedInstanceState)
12      enableEdgeToEdge()
13      setContentView(R.layout.activity_main)
14      ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
15          val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
16          v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
17          insets
18      }
19  }
20  }
```

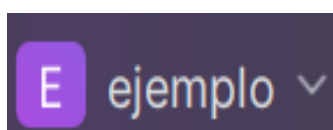
Ahora la parte de arriba.



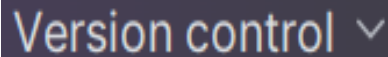
Abre o cierra el menú lateral izquierdo (Project, Structure, etc.) útil para ganar espacio si estás en una pantalla pequeña.



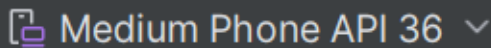
Muestra el nombre actual del proyecto abierto puedes cambiar entre proyectos o abrir otro desde aquí.



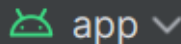
Muestra la rama actual de Git, el historial y otras herramientas de control de versiones muy útil si estás usando Git, GitHub o cualquier VCS.

A dark grey rectangular button with the text "Version control" in a light blue font, followed by a small downward-pointing chevron icon.

Aquí seleccionas el dispositivo donde correrás tu app puede ser un **emulador Android** o un **dispositivo físico conectado** también puedes crear uno nuevo desde aquí.

A dark grey rectangular button with a small purple icon of a phone on the left, followed by the text "Medium Phone API 36" in a light blue font, and a small downward-pointing chevron icon on the right.

Indica qué módulo del proyecto se ejecutará (normalmente "app") en proyectos más grandes puedes tener varios módulos.

A dark grey rectangular button with a small green Android robot icon on the left, followed by the text "app" in a light blue font, and a small downward-pointing chevron icon on the right.

Compila y lanza tu app en el dispositivo/emulador seleccionado, atajo: Shift + F10.



Solo compila el proyecto sin abrir la app, útil para verificar si hay errores de compilación, atajo: Ctrl + F9.



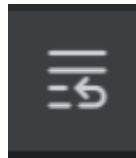
Cambia entre configuraciones como debug, release, o pruebas específicas, también puedes acceder a las configuraciones de ejecución (Edit Configurations).



Aplica cambios en todo el código y vuelve a iniciar la app.



Aplica los cambios en el código.



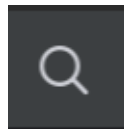
Conectar el depurador de Android Studio a un proceso de una app que ya está corriendo en un dispositivo (emulador o físico), **tu app ya está corriendo** (porque la ejecutaste manualmente o desde otro IDE/dispositivo), pero no se lanzó desde Android Studio.



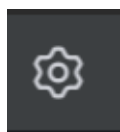
Este botón **sincroniza tu proyecto con los archivos de configuración de Gradle** (build.gradle, settings.gradle, etc.), actualizando Android Studio para que reconozca cualquier cambio.



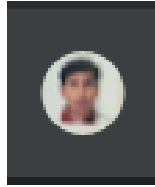
Busca archivos, clases, métodos, acciones o configuraciones en todo el proyecto, atajo: Shift dos veces.



Abre la configuración completa de Android Studio, puedes cambiar temas, atajos, apariencia, SDK, plugins, etc.



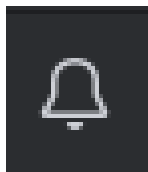
Muestra la cuenta de Google o JetBrains conectada, puedes cambiar de cuenta o cerrar sesión.



Ahora la parte de la derecha.



Muestra las notificaciones del propio IDE.



El botón **Gradle** abre una **ventana lateral** donde puedes ver y ejecutar manualmente todas las **tareas (tasks)** que Gradle tiene disponibles para tu proyecto.



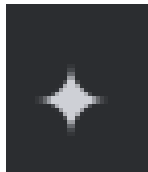
Son los emuladores disponibles que puedes utilizar en tu proyecto tiene información como su nombre, su API, su tipo, si quieres ejecutarlo, y opciones, incluso podrías crear uno o modificar los ya existentes.



Es la ventana del emulador, donde si no se esta ejecutando nada, te el “link” para poder ir a la anterior interfaz de emuladores disponibles.

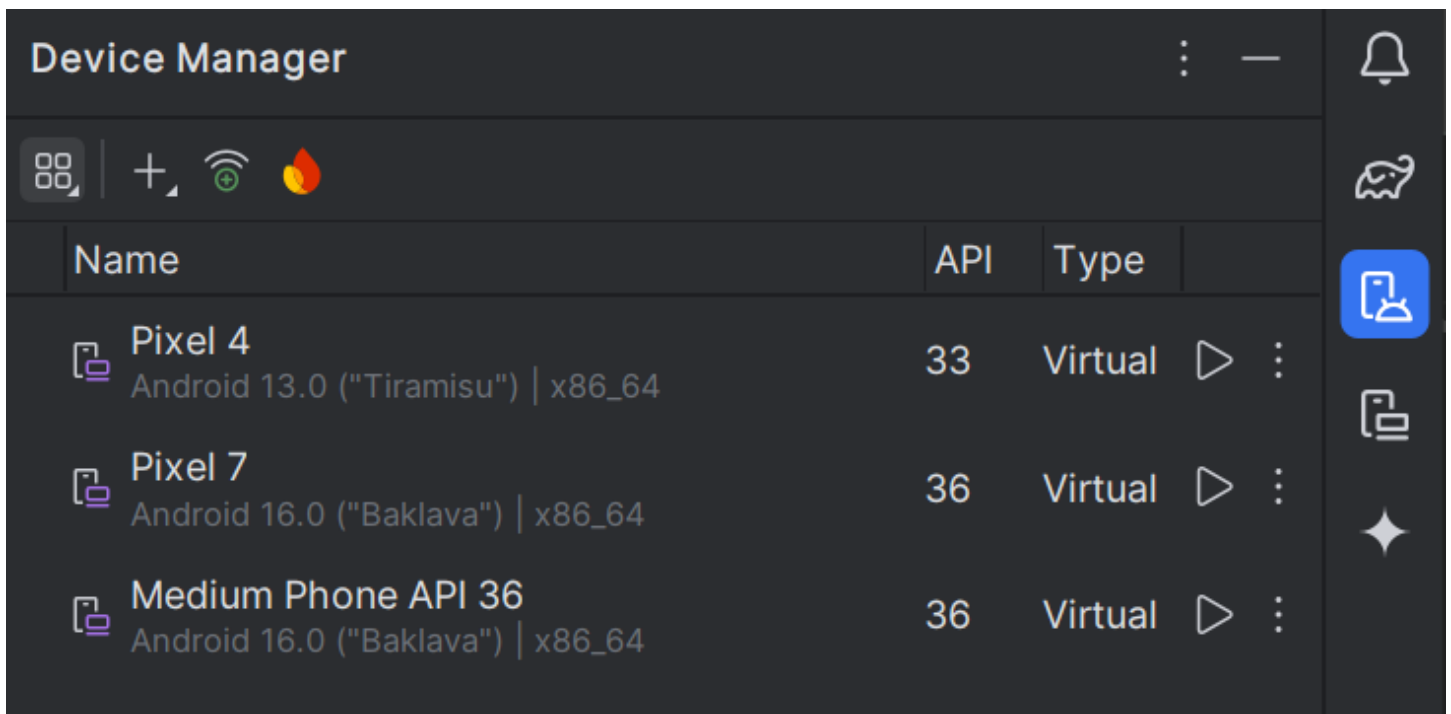


Es la IA de Google Gemini para poder ayudarte en tus códigos, analizándolos desde a la propia app, y no necesariamente escribir el código.

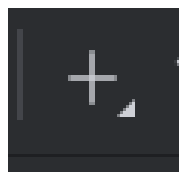


Emuladores

Regresando al tema de emuladores ya que son muy importantes, explicare como crear, o modificar algunos al presionar el botón device manager tendremos esta interfaz.



Donde se presentaran los emuladores disponibles, de cajón tiene instalado el que se llama “Medium Phone API 36” pero si no te gusta ese emulador puedes crear tu propio emulador, presionando el mas.



Te aparecerán dos opciones al presionarlo, si queremos que sea virtual ósea que nuestro propio equipo ejecute el emulador, o si queremos que sea remoto, ósea que con un celular ejecute nuestra propia app.

Create Virtual Device
Select Remote Devices

Utilizaremos el create virtual device, para que sea más rápido y sencillo, donde nos mostrara una interfaz, donde nos muestra en la parte izquierda, que sea, si un celular, una tableta, un reloj inteligente, una computadora, una televisión, o para un automóvil con pantalla, en este caso nos centraremos en el teléfono, y en el centro tenemos por así decirlo los perfiles que tienen como celulares ya existentes, donde nos muestran sus ventajas o estadísticas, si cuentan con Play store, su API, las dimensiones de la pantalla, y su densidad.

Add Device

Form Factor

- ☒ Phone
- ☐ Tablet
- ☐ Wear OS
- ☐ Desktop
- ☐ TV
- ☐ Automotive

☐ Show obsolete device profiles

Search for a device by name

Name	Play	API	Width	Height	Density
Small Phone		24+	720	1280	320 dpi
Medium Phone		24+	1080	2400	420 dpi
yiii		1+	1080	1920	480 dpi
New Device		1+	1080	1920	480 dpi
Pixel 9 Pro XL		35+	1344	2992	480 dpi
Pixel 9 Pro Fold		35+	2076	2152	390 dpi
Pixel 9 Pro		35+	1280	2856	480 dpi
Pixel 9		35+	1080	2424	420 dpi
Pixel 8a		34+	1080	2400	420 dpi
Pixel 8 Pro		34+	1344	2992	480 dpi
Pixel 8		34+	1080	2400	420 dpi
Pixel Fold		34+	2208	1840	420 dpi
Pixel 7a		34+	1080	2400	420 dpi
Pixel 7 Pro		33+	1440	3120	560 dpi
Pixel 7		33+	1080	2400	420 dpi
Pixel 6a		33+	1080	2400	420 dpi
Pixel 6 Pro		31+	1440	3120	560 dpi
Pixel 6		31+	1080	2400	420 dpi
Pixel 5		30+	1080	2340	440 dpi
Pixel 4a		30+	1080	2340	440 dpi
Pixel 4 XL		29+	1440	3040	560 dpi
Pixel 4		29+	1080	2280	440 dpi

New hardware profile... Import hardware profile... Cancel Previous Next Finish

También en la parte de abajo, nosotros podemos crear nuestro propio “perfil” o importarlo desde otro lado, donde modificaras los aspectos ya mencionados.

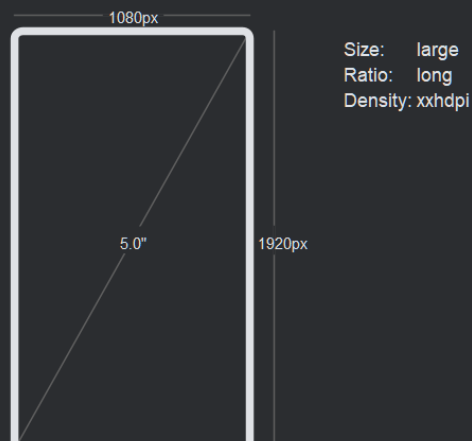


Configure Hardware Profile

Configure this hardware profile

Device Name	New Device 1		
Device Type	Phone/Tablet		
Screen	Screen size:	5	inch
	Resolution:	1080	x 1920 px
	<input type="checkbox"/> Round		
Memory	RAM:	2048	MB
Input	<input type="checkbox"/> Has Hardware Buttons (Back/Home/Menu)		
	<input type="checkbox"/> Has Hardware Keyboard		
	Navigation Style:	None	
Supported device states	<input checked="" type="checkbox"/>	Portrait	
	<input checked="" type="checkbox"/>	Landscape	
Cameras	<input checked="" type="checkbox"/>	Back-facing camera	

New Device 1



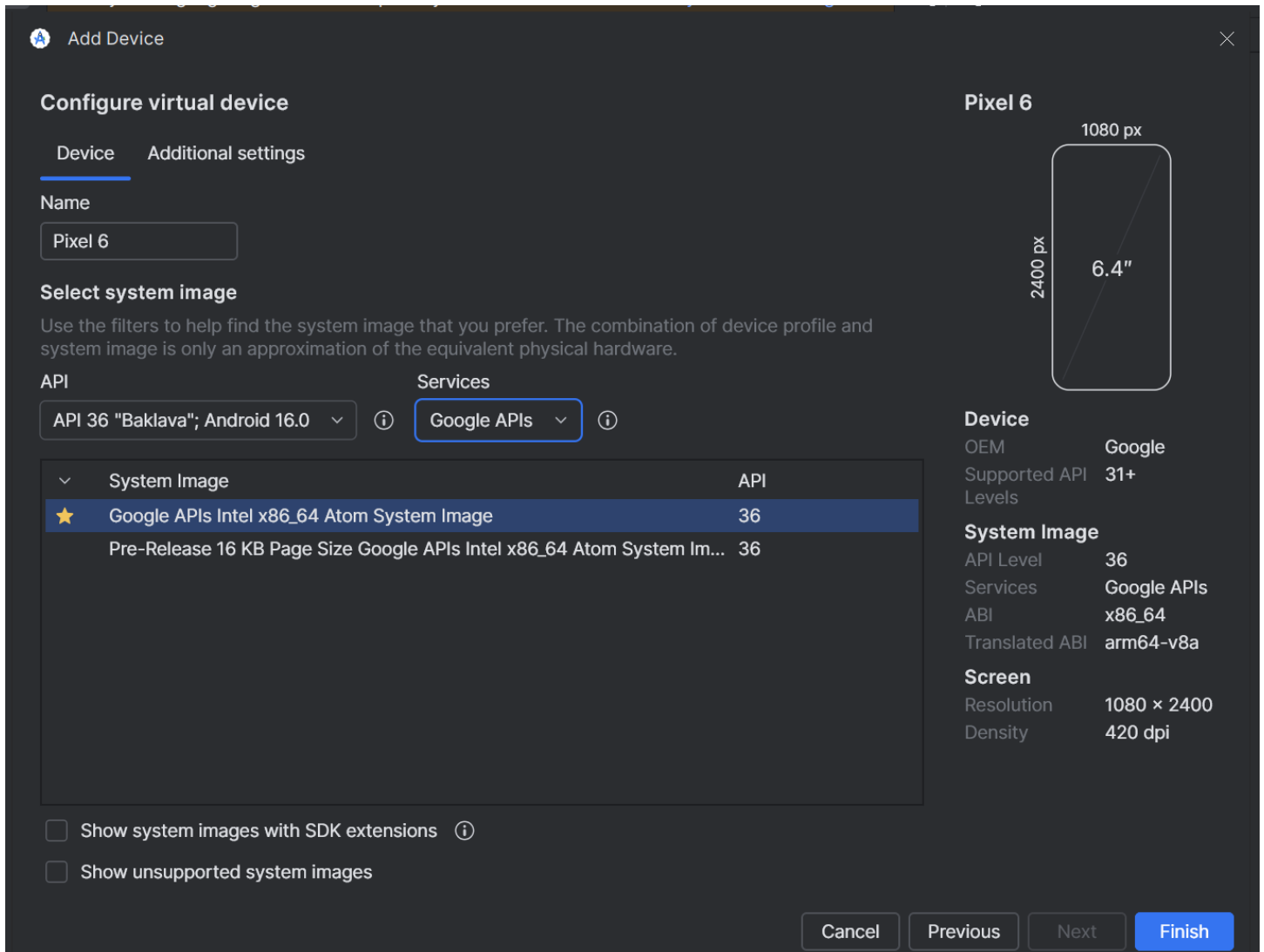
Cancel

Finish

Pero por lo mientras regresemos a los perfiles, donde podemos seleccionar uno, elegiremos el pixel 6 como ejemplo, pero tu puedes elegir el que sea.

 Pixel 6

Aparecerá una interfaz donde podrás configurar pocos detalles, por así decirlo lo “general” de tu emulador, donde podemos modificar su nombre, la versión de Android, sus servicios, si quieres la APIs de Google, la imagen de su sistema, etc.



Pero hasta arriba, esta la pestaña de additional settings, donde podemos ajustar mas el emulador en detalles mas técnicos.

Configure virtual device

Device

Additional settings

Device skin

pixel_6

Camera

Front

Emulated

Rear

VirtualScene

Network

Speed

Full

Latency

None

Startup

Orientation

Portrait

Default boot

Quick

Storage

Internal storage

2

GB

Expanded storage

Custom

512

MB

Existing image

None

Emulated Performance

CPU cores

4

Graphics acceleration

Automatic

RAM

2

GB

Pixel 6

1080 px

2400 px

6.4"

Device

OEM

Google

Supported API Levels

31+

System Image

API Level

36

Services

Google APIs

ABI

x86_64

Translated ABI

arm64-v8a

Screen

Resolution

1080 × 2400

Density

420 dpi

Cancel

Previous

Next

Finish

Aquí podemos configurar el por así decirlo modelo del equipo, si queremos que tenga una cámara, que tenga acceso a internet, su booteo, cuanto espacio le queremos dar, si tendrá una memoria sé, y cuantos recursos de nuestro equipo le daremos al emulador.

Bibliografía

ChatGPT

Autor: OpenAI

Enlace: <https://chatgpt.com/>

Fecha: 18 de abril de 2025

YouTube

Autor: Programación Android by AristiDevs

Enlace: https://www.youtube.com/watch?v=vJapzH_46a8&t=14941s

Fecha: 2 de marzo de 2023

YouTube

Autor: MoureDev by Brais Moure

Enlace: https://www.youtube.com/watch?v=BQaxPwZWboA&list=PLNdFk2_brsRdYF0FXDtSaGvluzBNHRbNe

Fecha: 17 de enero de 2020

YouTube

Autor: MoureDev by Brais Moure

Enlace: https://www.youtube.com/watch?v=8WfQQ0BRDY8&list=PLNdFk2_brsRdYF0FXDtSaGvluzBNHRbNe&index=71

Fecha: 3 de septiembre de 2018

Conclusión

Este IDE es bastante confuso si es tu primera interacción en el mundo de la programación, pero gracias a sus herramientas, como Gemini, los pluggins, se convierte en un poco más sencillo, además de que es el “principal” IDE al querer realizar una app para Android, ósea que muchas personas lo utilizan, donde hay muchos cursos gratuitos y de paga, como recomendación personal, saber inglés, te facilita mucho entender las cosas, ya que si no entiendes nada de inglés, te pierdes muy fácil.

Es un excelente IDE en todo ámbito, y tendrá bastantes años de vida, ya que lo proporciona Google así que es una buena inversión a futuro aprender aquí.

Diferencias entre el Playground y Android Studio

*Introducción y/o presentación: al momento de ingresar al Playground de kotlin no hay muchas cosas ni avisos, solo un pequeño código que al momento de imprimir dice hola mundo, y botones.

Diferente a Android Studio que por así decirlo es un ambiente mas profesional, donde a primeras instancias nos pregunta como queremos el proyecto, o que vista queremos, con diferentes “vistas” predeterminadas, luego en donde se guardara, el nombre del proyecto, la versión mínima de Android en la que se ejecutara, tiene un ámbito o objetivo mas profesional, y el Playground tiene un objetivo mas casual o mas simple para solo entender el funcionamiento de Kotlin.

*Archivos: en Android Studio, tenemos la opción de guardar nuestros proyectos, crear más, modificar, y separar cada uno por proyecto para que no nos perdamos en este ámbito, cada archivo dependiendo de su funcionamiento, si es la lógica del código es .kt y si es la parte visual que vera el usuario es .xml

En el Playground esto no ocurre ya que es un “simulador” en línea, ósea que no ingresa a nuestra computadora para almacenar datos, o archivos, asi que solo sirve para probar códigos de manera rápida.

*Emuladores: En Android Studio tienes un gran repertorio de emuladores, dependiendo de su versión de Android, si quieres que tengan Play store o solo APIs o incluso crear tu tus propios emuladores, eligiendo la versión de Android, cuanta memoria RAM y ROM le quisieras asignar, si tenga cámara, o modificar los ya existentes que te da el propio Android Studio, también existe la manera de conectar tu propio celular para ejecutar las aplicaciones y así no ejecutar un emulador, que gastaría recursos de tu equipo.

En el Playground esto no existe, ya que no esta planeado para ejecutar algo tan complejo como emular una app y tampoco existe la idea de conectar tu celular para emular la app, como dije anteriormente el Playground su función principal es dar una muestra de como es el lenguaje de Kotlin y otros.

*Lenguajes de programación: en Android Studio primeramente tiene unos pocos lenguajes de programación como lo son Kotlin, Java, pero con su herramienta o

apartado de pluggins podemos agregar más, o “actualizar” los que ya tenemos de cajón.

En el Playground esto es un poco mas limitado, y confuso, ya que a lado de la versión del Playground en la esquina superior izquierda, tenemos los lenguajes que están escritos de diferente manera, en el cual, si eres alguien nuevo puede resultar confuso, están Kotlin, JavaScript, y Swift.

Conclusión: Android Studio es mucho mas completo en todo aspecto que el Playground, opciones, visualización, productividad, las principales ventajas que tiene el Playground es que no necesitas instalar nada en tu computadora, solo tienes que escribir código, y en el caso de Android Studio si es un poco pesada la instalación, e instalar los emuladores, así como también, tener un equipo medio, o medio alto para poder emular y ejecutar códigos en Android Studio, y en el Playground cualquier equipo que tenga internet se puede utilizar, en mi opinión el Playground se centra en el público que apenas están iniciando en este ámbito o solo quieren aprender el lenguaje, y el Android Studio se centra en el publico mas profesional.