PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

TEMA 2:

Instalación, Configuración Y Uso Del Entorno De Desarrollo Android Studio. Emuladores.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

ÍNDICE

PMDM

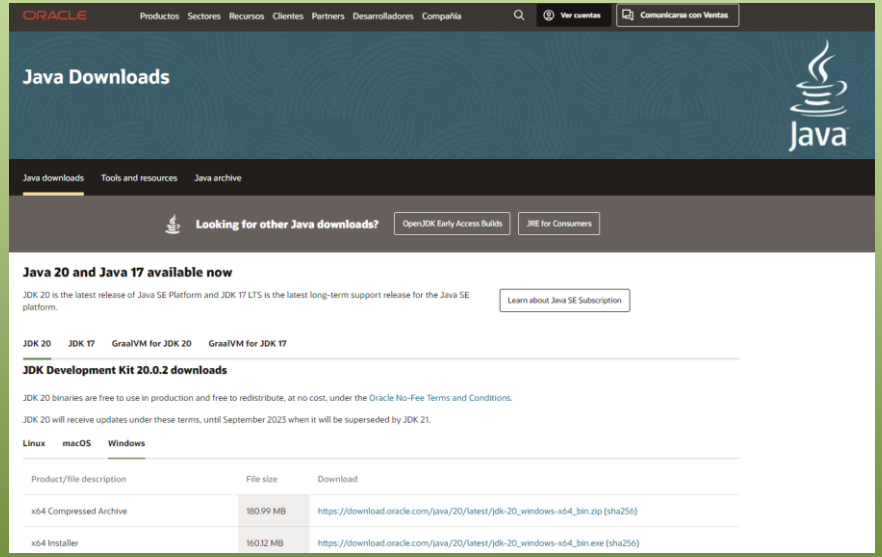
Instalación, Configuración Y Uso Del Entorno De Desarrollo Android Studio. Emuladores.

1. Instalación de Android Studio

1.- Instalación de Android Studio

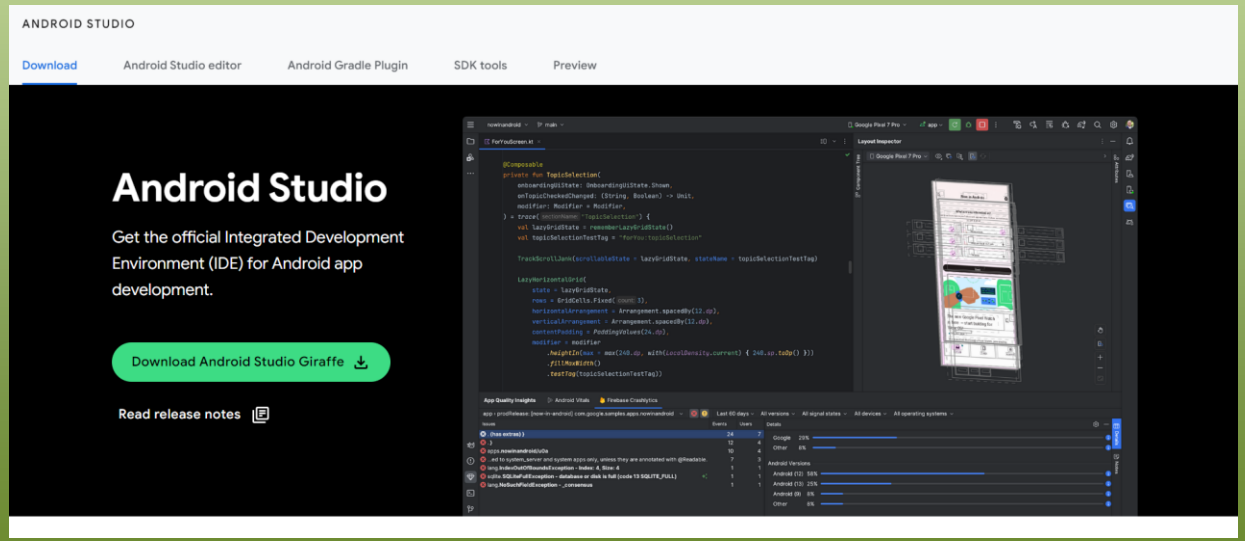
- **<u>Paso 1</u>:** Descarga e instalación de "Java Development Kit (JDK)" de la página de Oracle.com. Elegiremos el que corresponda con nuestro Sistema Operativo.

- Nota: *Si ya tienes instalado en tu equipo el jdk este paso no sería necesario.*



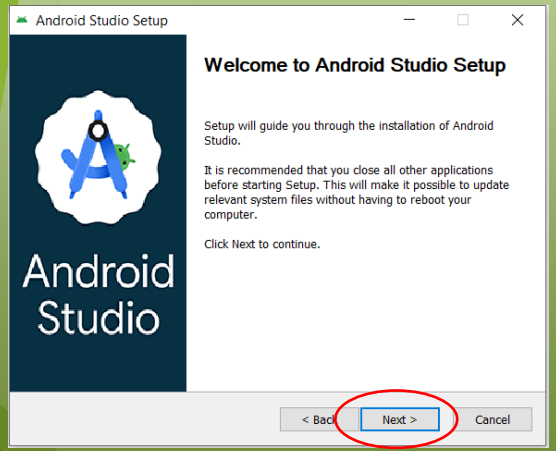
1.- Instalación de Android Studio

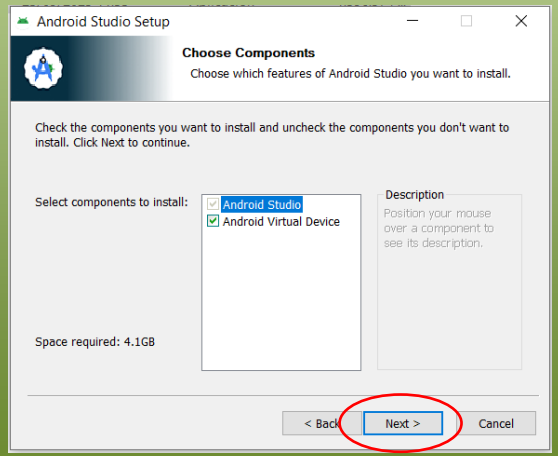
- **Paso 2:** Descarga e instalación de Android Studio: https://developer.android.com/studio

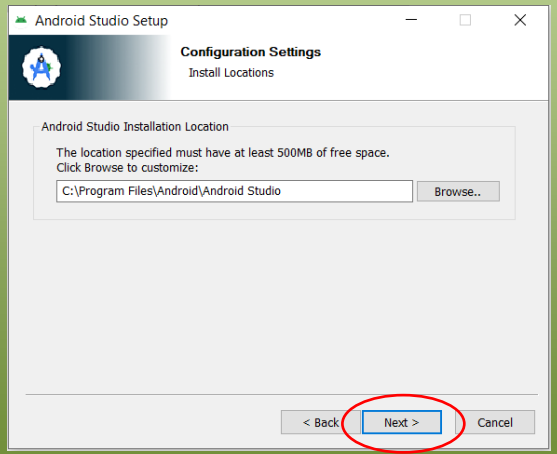


1.- Instalación de Android Studio

- Durante el proceso de instalación, podemos instalar también un emulador marcando la opción de *Android Virtual Device*:



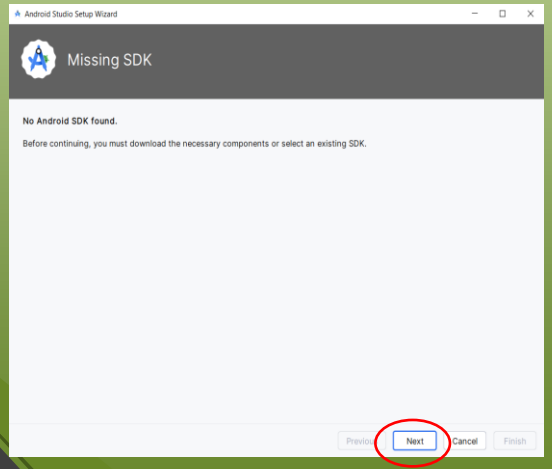


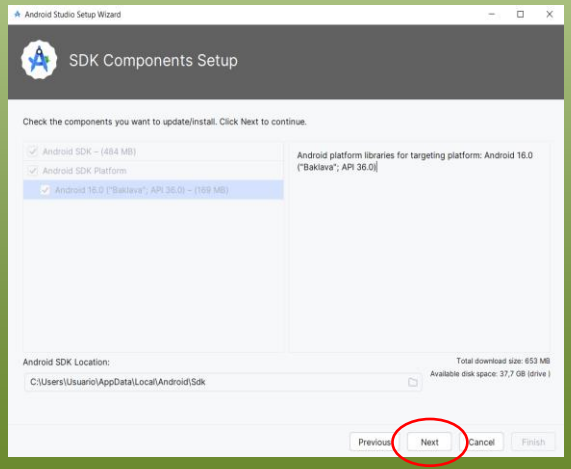


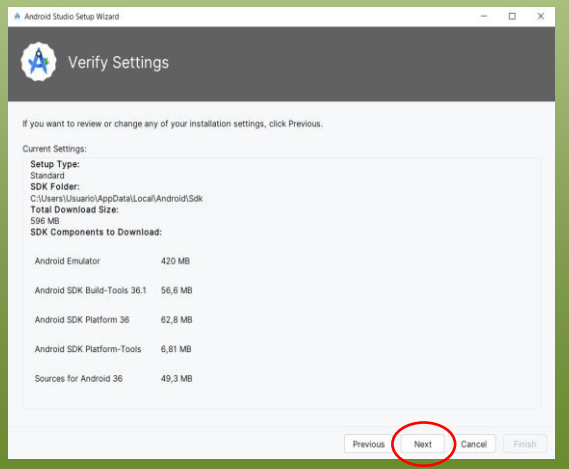
1.- Instalación de Android Studio

- **Paso 3**: Instalación y actualización de los componentes SDK de Android.

Podemos hacerlo la primera vez que abrimos Android Studio una vez finalizado el proceso de instalación. Para ello, abrimos Android Studio y seguimos los siguientes pasos:

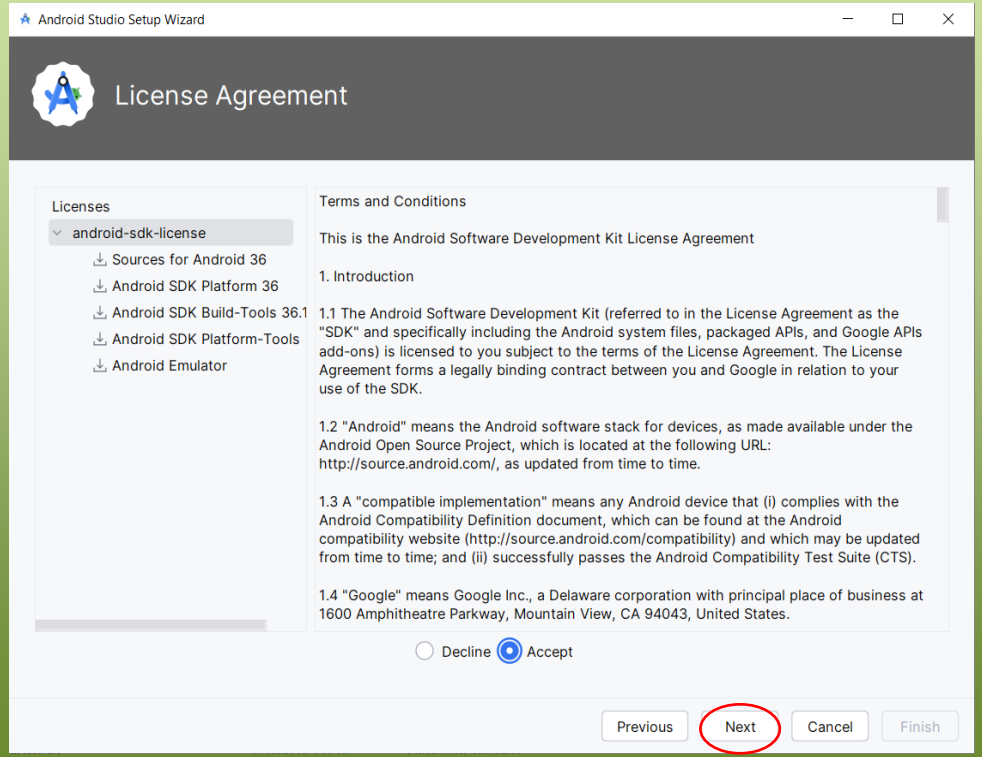






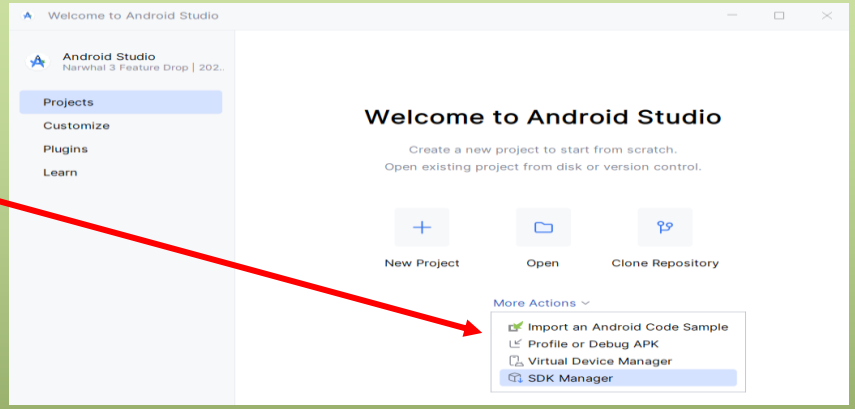
1.- Instalación de Android Studio

- Debemos aceptar las licencias para poder descarcargar todas las herramientas del SDK de Android:

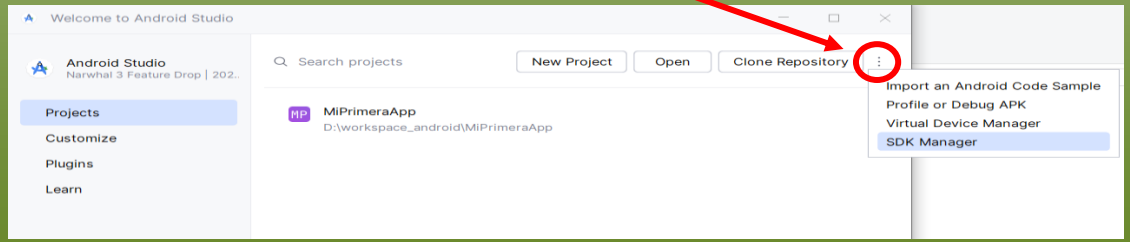


1.- Instalación de Android Studio

- Otra opción es hacerlo desde la ventana de bienvenida de Android Studio. Para ello, accederemos al menú "More actions / SDK Manager":

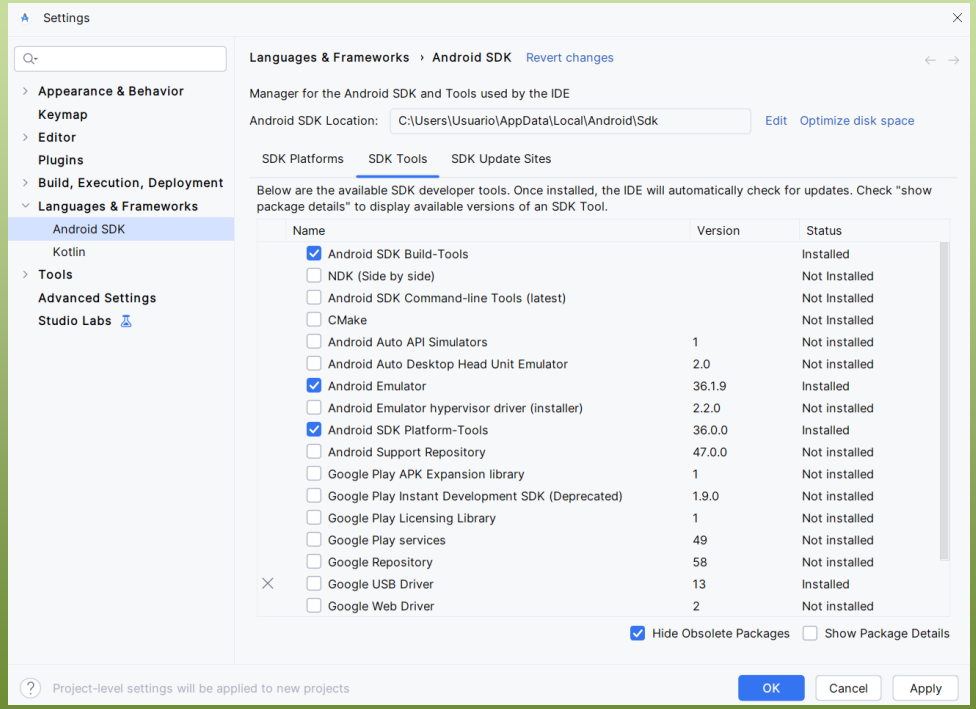


- Si ya hemos creado un proyecto anteriormente, también podemos acceder así (desde el menú de la ventana de proyectos recientes):



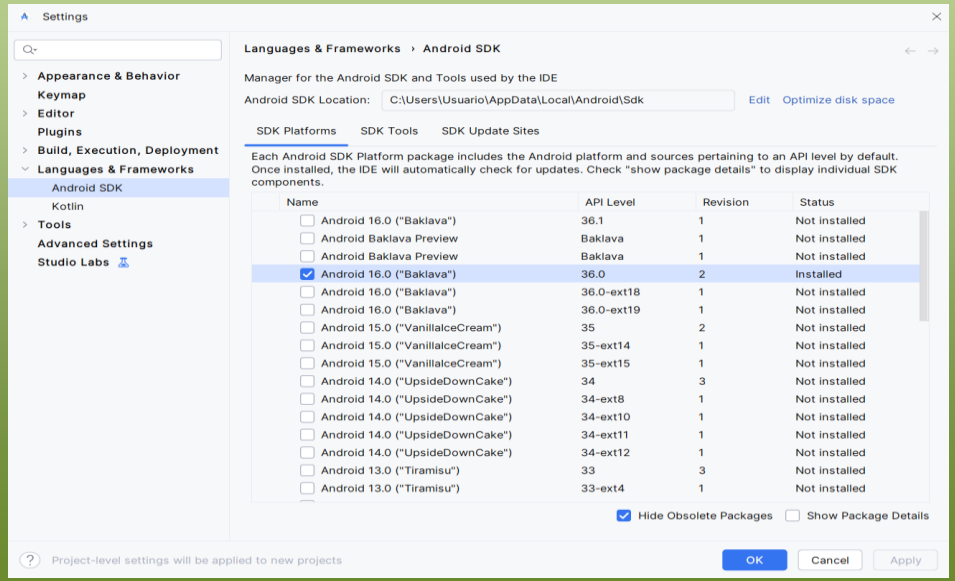
1.- Instalación de Android Studio

- Seleccionamos las opciones que aparecen marcadas en la pestaña *SDK Tools*:



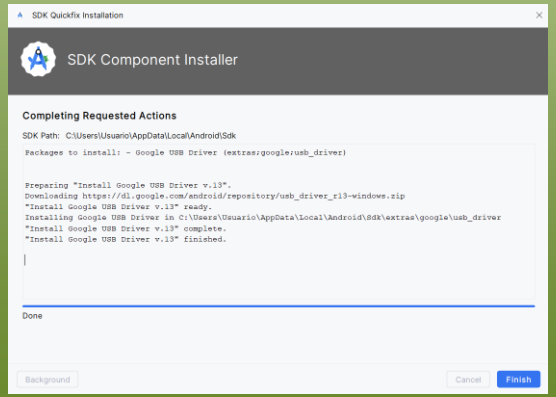
1.- Instalación de Android Studio

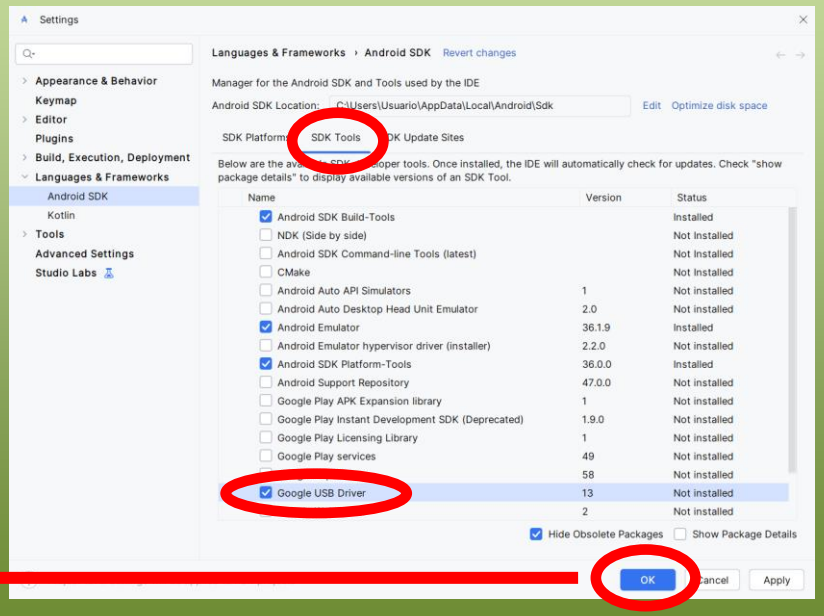
- Si queremos que la aplicación que desarrollemos sea compatible con versiones anteriores de Android, deberemos seleccionar las API´s deseadas. En la ventana de un proyecto, esta configuración se encuentra en **"File -> Settings -> Android SDK"**.



1.- Instalación de Android Studio

- Muy importante marcar en la pestaña SDK Tools la instalación del Google USB Driver, que nos permitirá ejecutar las aplicaciones que desarrollemos en nuestro teléfono conectado por usb al ordenador:





1.- Instalación de Android Studio

- Para más información sobre la instalación y configuración del entorno visita:
  - Descargar e instalar Android Studio (Android Developers): https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#0
  - Cómo instalar Android Studio (Android Developers): https://developer.android.com/studio/install?hl=es-419
  - Instalación Android Studio Giraffe (2023): https://www.youtube.com/watch?v=tvOBTwamsSo
  - Instalación Android Studio Koala (2024): https://www.youtube.com/watch?v=ZDS6_u6Pmp4

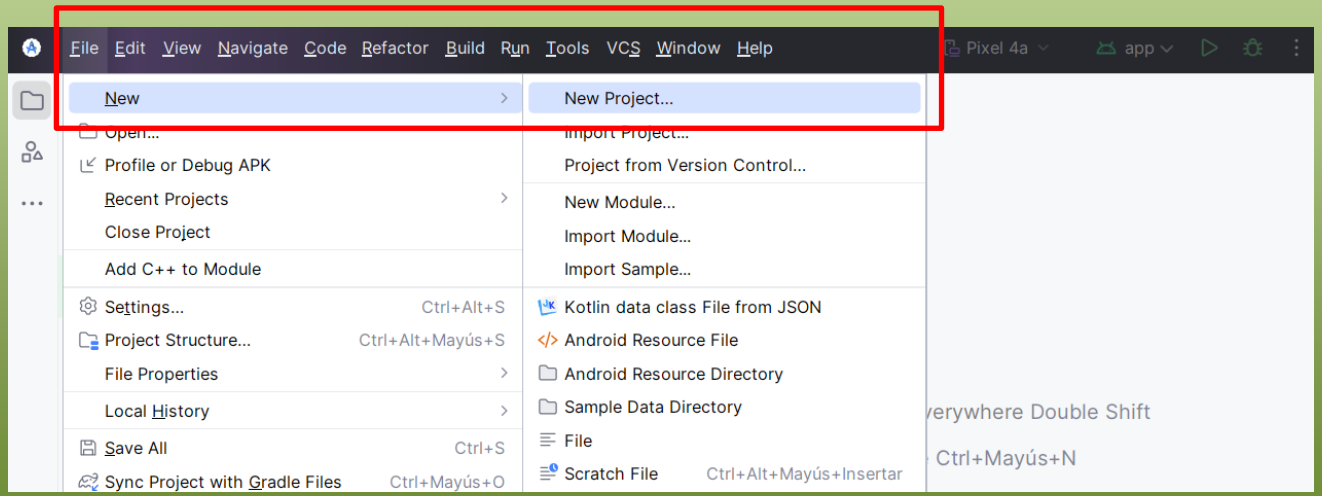
PMDM

Instalación, Configuración Y Uso Del Entorno De Desarrollo Android Studio. Emuladores.

2. Estructura de un proyecto

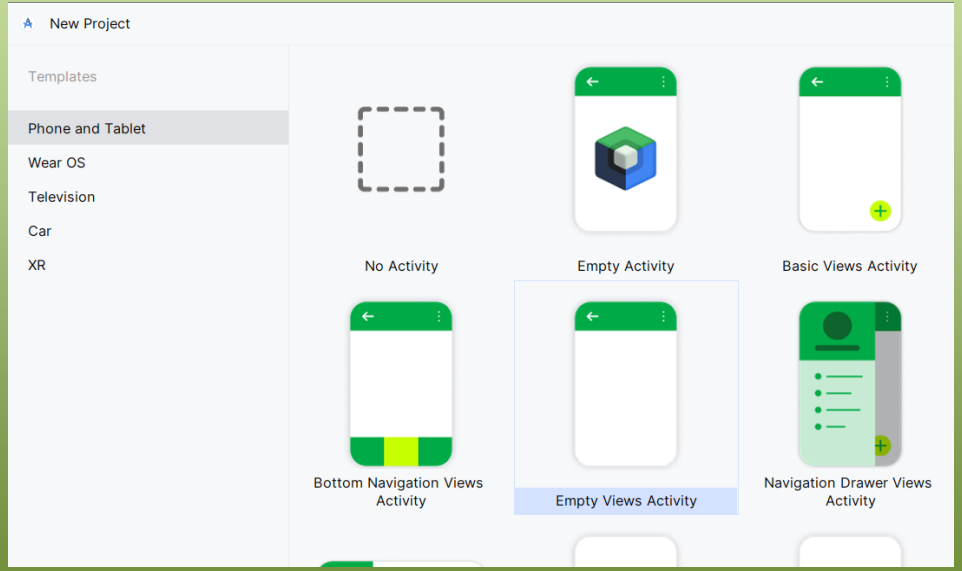
2.- Estructura de un proyecto

- Para crear una nueva app, ejecutaremos Android Studio y haremos click en File -> New -> New Project:



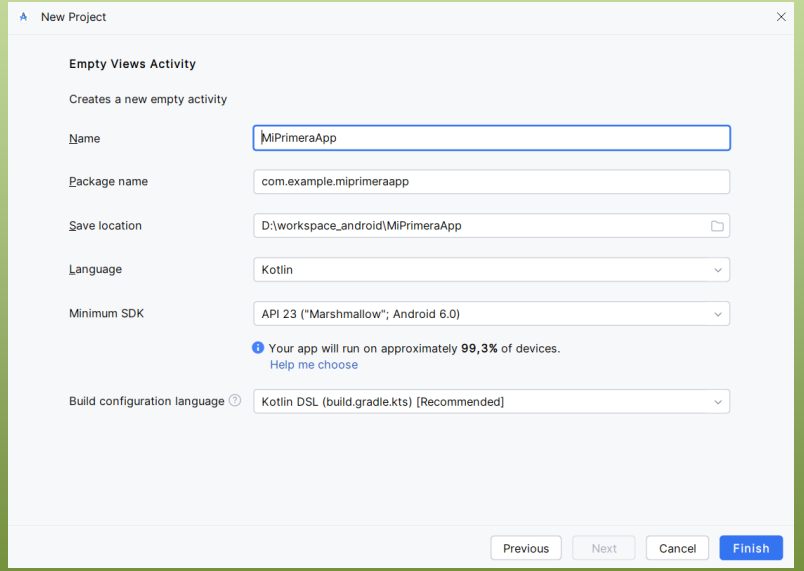
2.- Estructura de un proyecto

- En la siguiente pantalla elegiremos el tipo de actividad principal de la aplicación (Una actividad es una "ventana" o "pantalla" de la aplicación con interfaz de usuario)
- Para empezar, elegiremos "*Empty Views Activity*", que en realidad es una plantilla vacía.



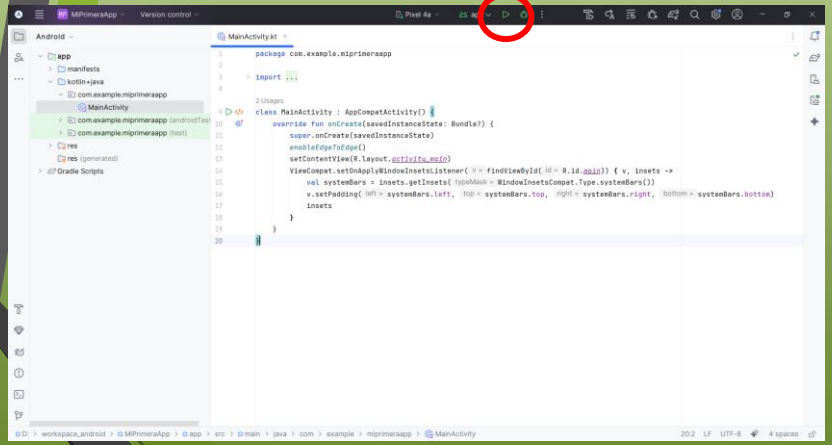
2.- Estructura de un proyecto

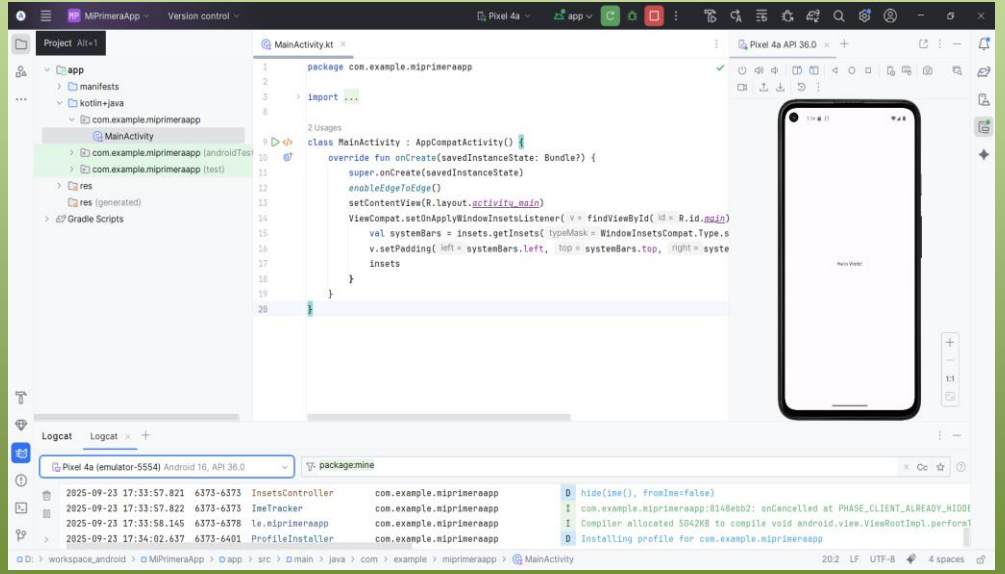
- En la primera pantalla indicaremos el nombre de la aplicación, el nombre del paquete por defecto, y la ruta donde crear el proyecto (IMPORTANTE: Comprobar que al final de la ruta añade el nombre de la app, para que así se cree una carpeta específica para el proyecto).
- API Mínima que soportará nuestra aplicación. La opción "Help me choose" nos ayudará a elegir.
- **El lenguaje que utilizaremos será Kotlin**



2.- Estructura de un proyecto

- Si pulsamos el botón "run App" compilaremos y ejecutaremos nuestra aplicación:



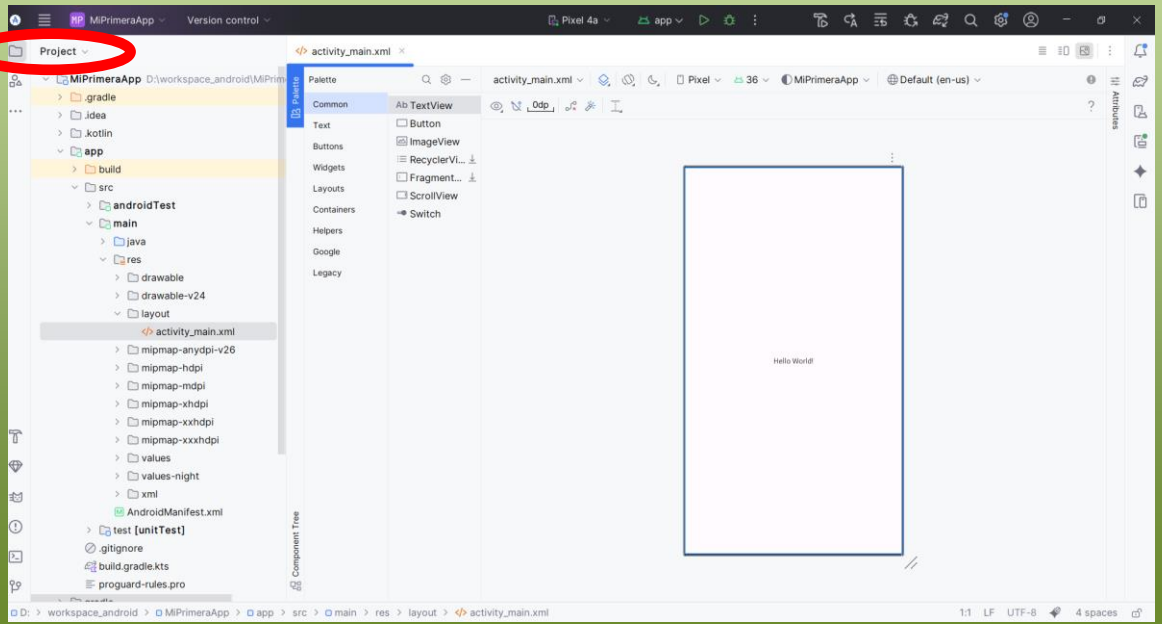


EJERCICIOS

- **Ejercicio 01.** Crea un proyecto Android llamado *MiPrimeraApp*, ejecútalo y comprueba que se abre correctamente el emulador.

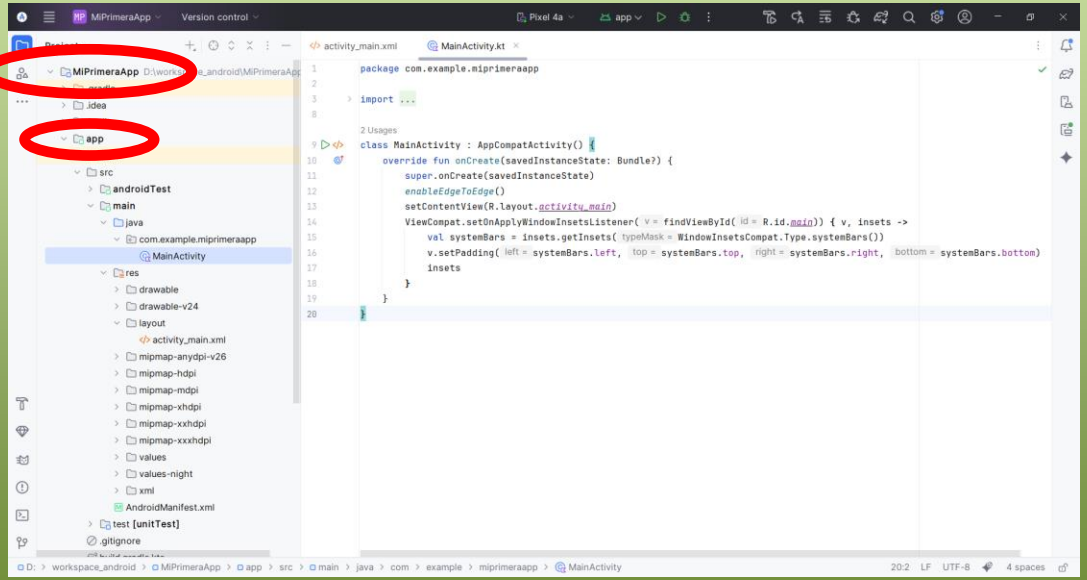
2.- Estructura de un proyecto

- Para un mejor entendimiento de los archivos que forman parte de nuestro proyecto es recomendable cambiar la vista al modo "Project":

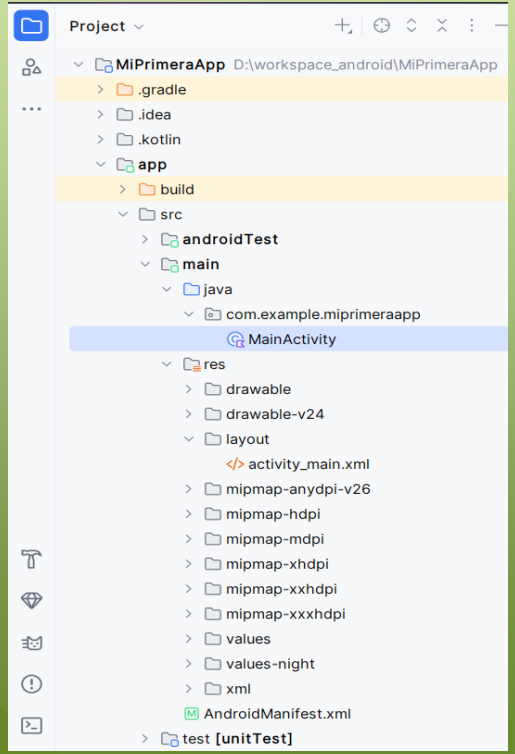


2.- Estructura de un proyecto

- La entidad proyecto es única, y engloba a todos los demás elementos.
- Dentro de un proyecto podemos incluir varios módulos, que pueden representar aplicaciones distintas, versiones diferentes de una misma aplicación, o distintos componentes de un sistema (aplicación móvil, aplicación servidor, librerías, ...).
- En este caso que estamos creando tenemos el proyecto "MiPrimeraApp" que contiene al módulo "app" que contendrá todo el software de la aplicación.



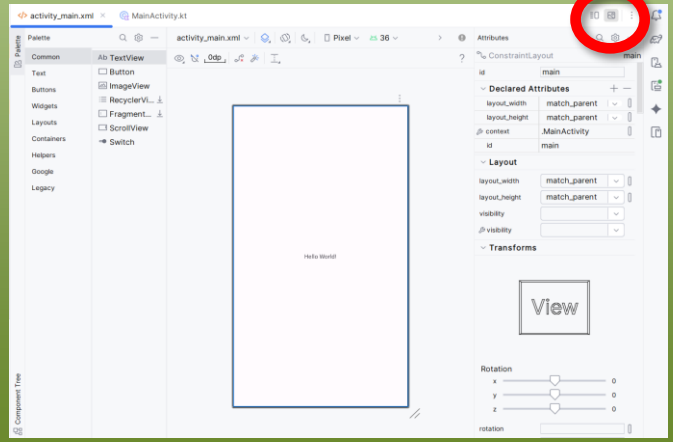
2.- Estructura de un proyecto



- **/src:** En este directorio es donde se almacenan los paquetes (en la carpeta java), y dentro lo archivos de código fuente KOTLIN (con extensión .kt)
- **/res:** Es el directorio principal de recursos (resources). Aquí guardaremos imágenes o archivos multimedia que utilice nuestra aplicación. Para definir diferentes recursos dependiendo de la resolución de la pantalla del dispositivo se suele dividir en varias subcarpetas:
  - /drawable (recursos independientes de la densidad)
  - /drawable-ldpi (densidad baja)
  - /drawable-mdpi (densidad media)
  - /drawable-hdpi (densidad alta)

2.- Estructura de un proyecto

- **/res/layout:** Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Como en nuestro proyecto solo tenemos una Activity pues solo tenemos un fichero que define su interfaz, activity_main.xml.

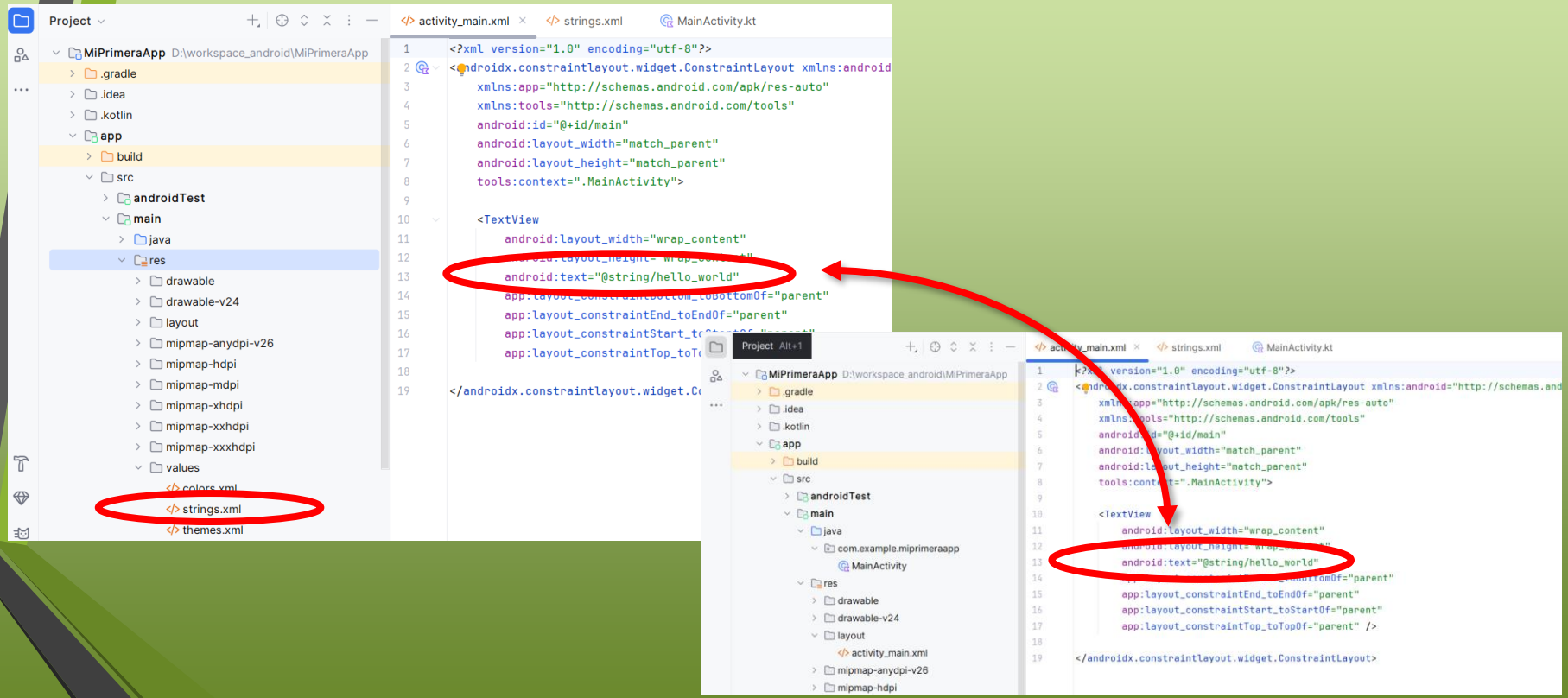


2.- Estructura de un proyecto

- **/res/values:** Contiene otros ficheros XML de recursos de la aplicación, como por ejemplo cadenas de texto (*strings.xml*), estilos (*styles.xml*), colores (*colors.xml*), arrays de valores (*arrays.xml*), tamaños (*dimens.xml*), etc.

De igual forma que separamos el código Kotlin y la interfaz gráfica, Android separa también las cadenas constantes de texto (para la internacionalización de la aplicación), las matrices, la paleta de colores... Veamos un ejemplo:

2.- Estructura de un proyecto

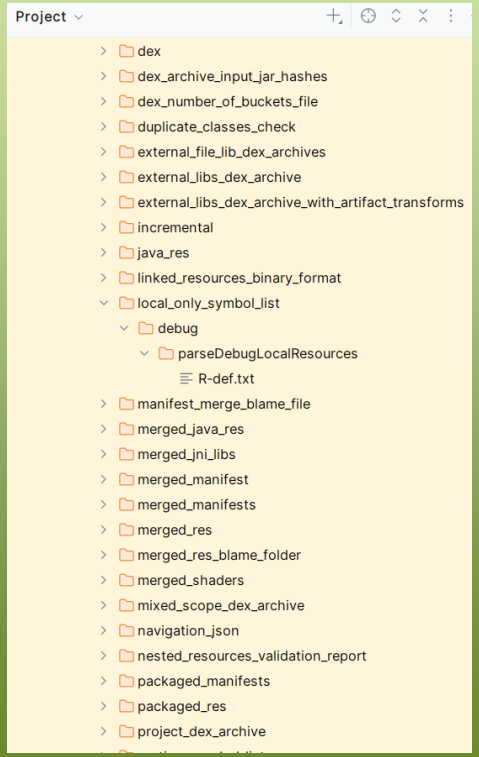


2.- Estructura de un proyecto

- **AndroidManifest.xml:** Es el archivo de configuración de la aplicación en el que se define lo que puede hacer nuestra aplicación, es decir, en él informamos al sistema operativo de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, icono, ...), sus componentes (pantallas, servicios, ...) y de las capacidades que tiene esta aplicación.
- También indicaremos en él las actividades o servicios que ejecutará nuestra aplicación, y los permisos a recursos compartidos del sistema, como por ejemplo el acceso al listado de contactos, utilización del GPS o la posibilidad de enviar mensajes SMS.

2.- Estructura de un proyecto

- **/app/libs:** Puede contener las librerías java externas (ficheros .jar) que utilice nuestra aplicación.
- **/app/build:** Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Destacamos sobre todo el fichero que aparece desplegado en la imagen, llamado "R-def.txt", relativo a ***la clase R***. Este fichero se encuentra en intermediates/local_only_symbol_list/debug/parseDebugLocalResources/. Esta clase R contiene una serie de constantes con los identificadores (ID) de todos los recursos de la aplicación incluidos en la carpeta /app/src/main/res/, lo cual nos permite acceder fácilmente a estos recursos desde nuestro código kotlin a través de dicho dato. Así, por ejemplo, la constante *R.layout.activity_main* contendrá el ID del layout "activity_main.xml", fichero XML que podemos encontrar en la carpeta /app/src/main/res/layout/.

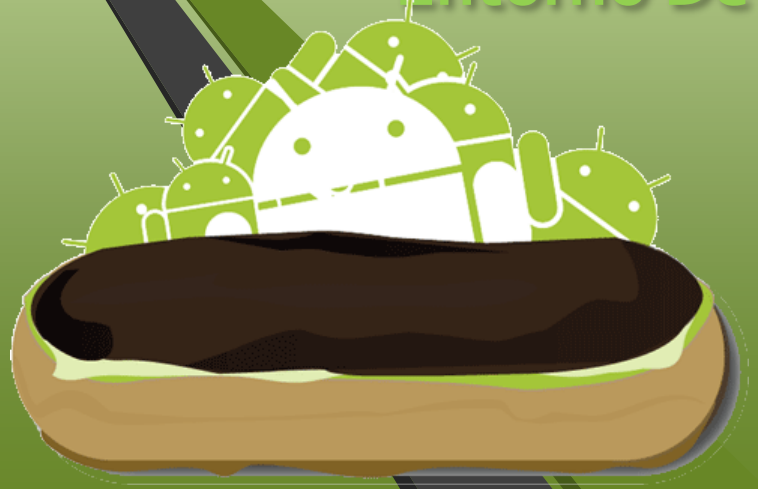


2.- Estructura de un proyecto

- Para más información sobre la estructura de los proyectos Android Studio visita:
  - Descripción general de proyectos (Android Developers):
    https://developer.android.com/studio/projects?hl=es-419

PMDM

Instalación, Configuración Y Uso Del Entorno De Desarrollo Android Studio. Emuladores.



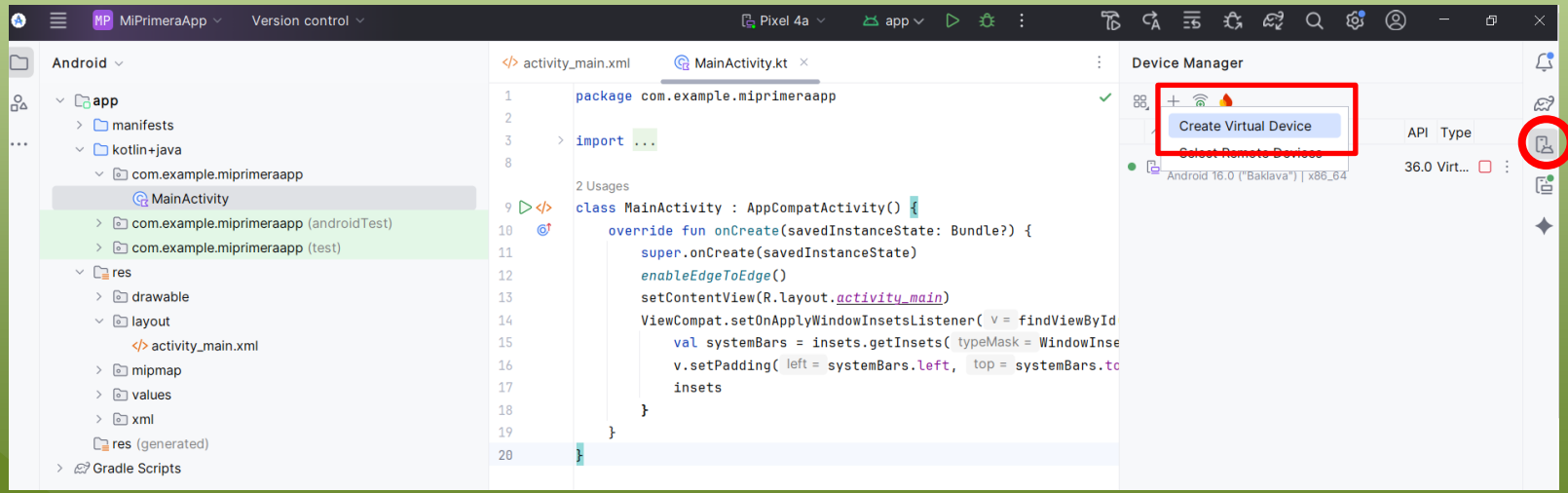
3. Creación de un emulador

3.- Creación de un emulador

- Para poder probar el funcionamiento de nuestra aplicación tendremos dos opciones:
  1. **Crear un emulador de un dispositivo Android.**
  2. **Utilizar un dispositivo físico Android conectado por USB a nuestro ordenador.**

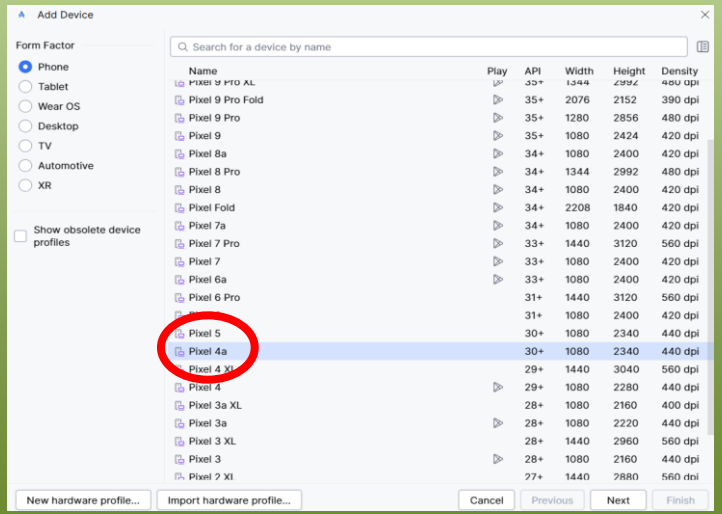
3.- Creación de un emulador

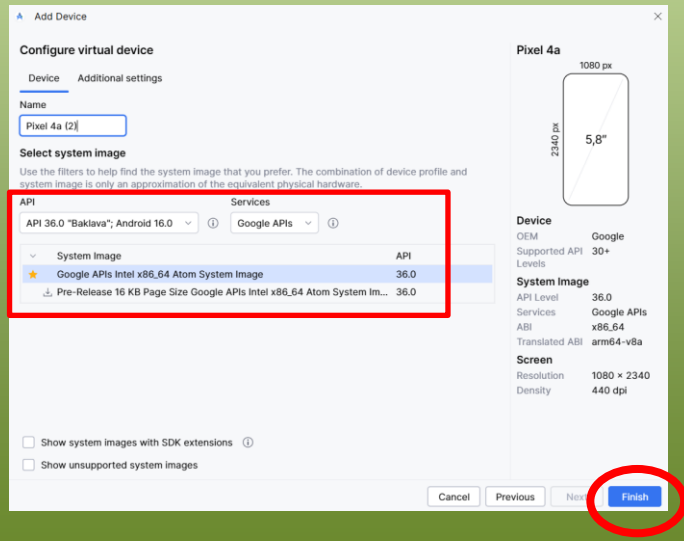
- Si en la instalación no elegimos la opción de instalar también un emulador tendremos que crearlo posteriormente. Para crear un emulador de un dispositivo utilizaremos el **Device Manager**:



3.- Creación de un emulador

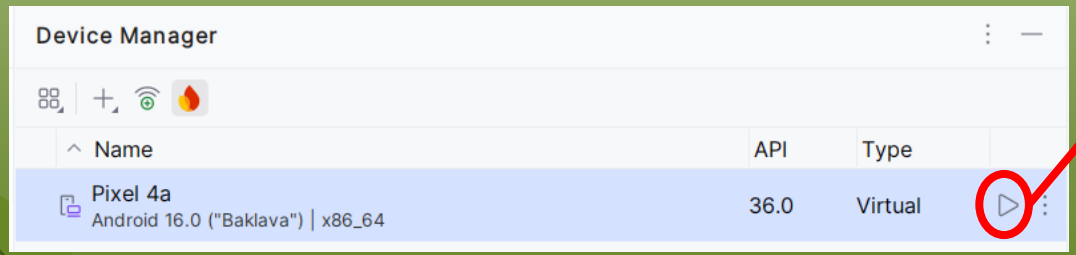
- Elegiremos una configuración que nos guste, por ejemplo un dispositivo Pixel 4a con la última versión de Android, que utiliza la **API 36**.





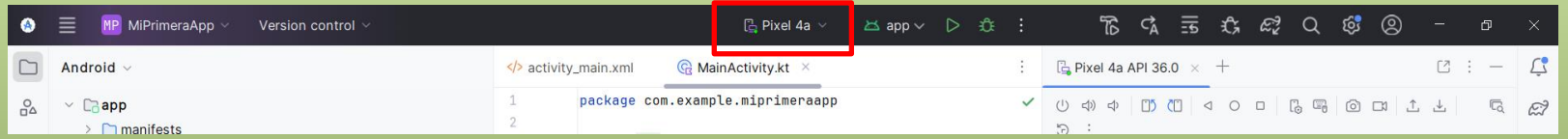
3.- Creación de un emulador

- Para dejar iniciado el emulador y poder probar posteriormente nuestras aplicaciones en él, pulsamos en el triangulo verde. La primera vez que lo iniciamos tardará un rato (minuto y medio) ya que tiene que reservar un espacio en nuestro disco duro. Los posteriores inicios serán más rápidos (unos 30 segundos)

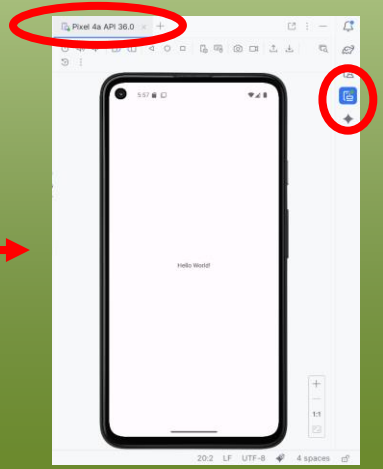


3.- Creación de un emulador

- Una vez que tenemos el emulador iniciado, marcamos el emulador en el que queremos ejecutar nuestra aplicación antes de lanzarla:



- En la ventana de **Running Devices** podemos ver nuestra app, la cual se ha ejecutado en el emulador seleccionado:



3.- Creación de un emulador

- Para más información sobre otros emuladores Android para PC (BlueStacks, Genymotion, NoxPlayer, GameLoop…) visita:

https://www.xataka.com/basics/mejores-emuladores-android-para-pc

PMDM

Instalación, Configuración Y Uso Del Entorno De Desarrollo Android Studio. Emuladores.

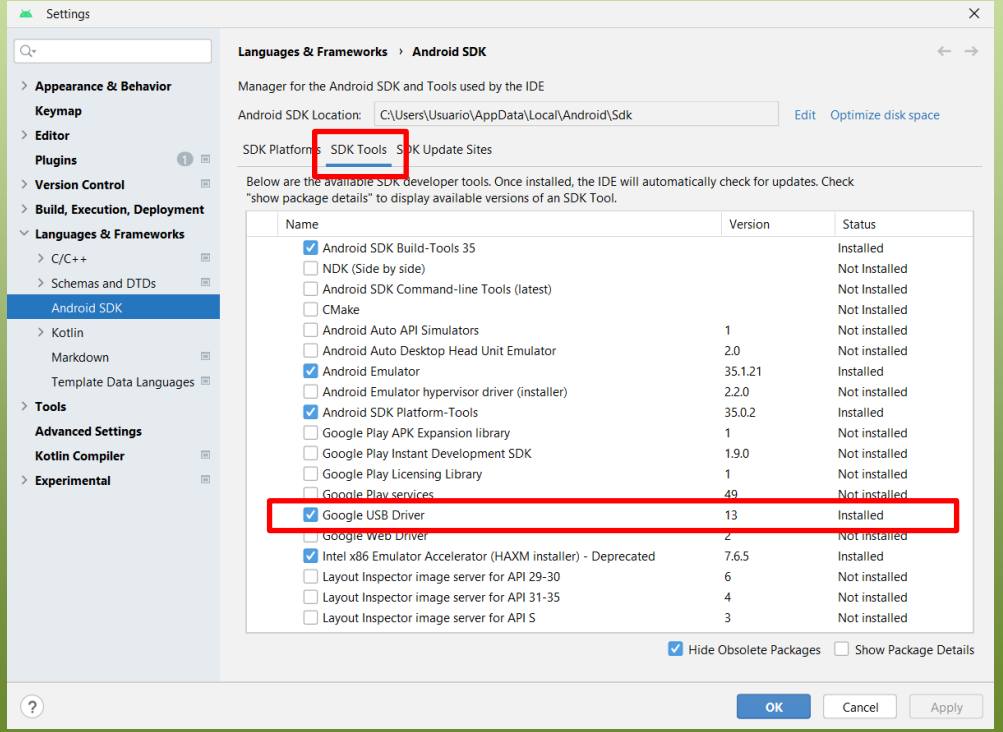
4. Ejecución de aplicaciones en dispositivos físicos

4.- Ejecución de aplicaciones en dispositivos físicos

- Si queremos probar y depurar nuestras app en nuestros dispositivos Android (en vez de en un emulador), lo primero que debemos hacer es configurarlo para que acepte operaciones de depuración. Para ello, en el menú ***Ajustes***, entramos en las ***Opciones de desarrollo*** y activamos la ***Depuración USB***.
- *Nota: A partir de la versión 4.2 están ocultas y hay que activarlas pulsando 7 veces en el campo "Número de compilación" dentro del menú Acerca del dispositivo.*

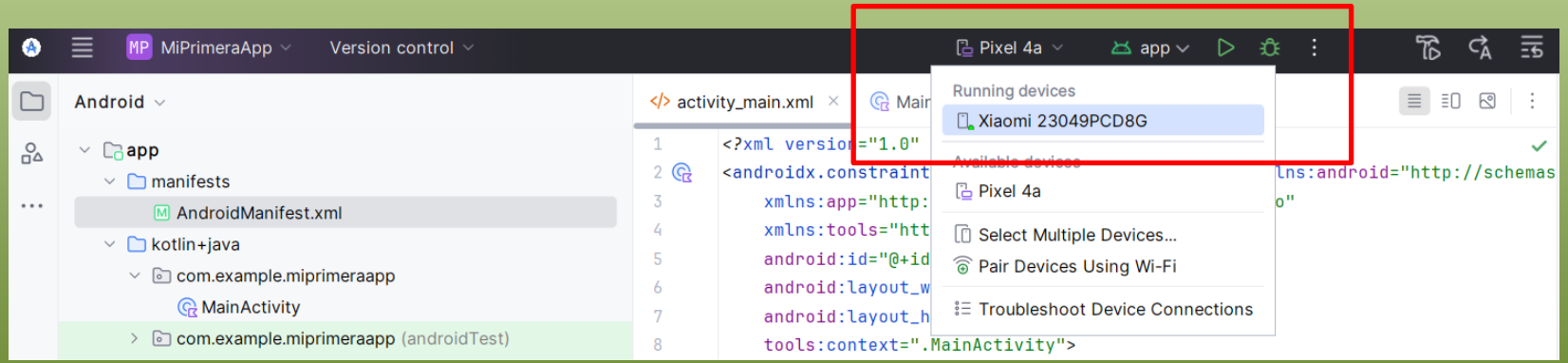
4.- Ejecución de aplicaciones en dispositivos físicos

- El siguiente paso es instalar el **Google USB Driver** en nuestro ordenador. Esto lo podemos hacer en Android Studio, desde la opción **File -> Settings -> Languages & Frameworks -> Android SDK -> SDK Tools**:



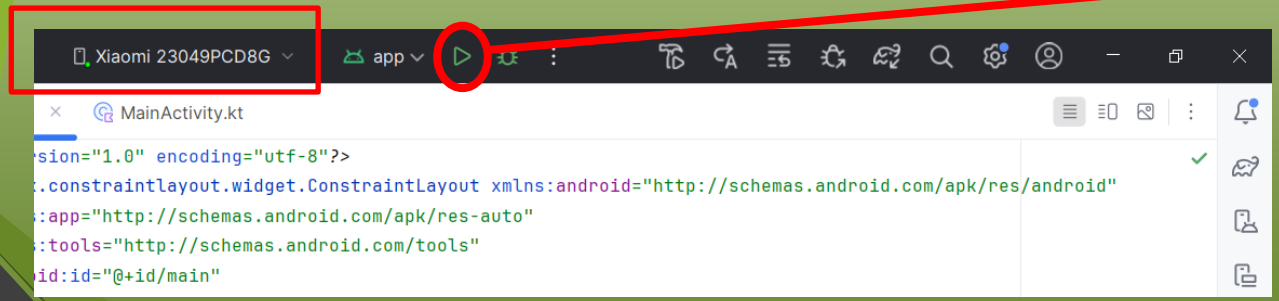
4.- Ejecución de aplicaciones en dispositivos físicos

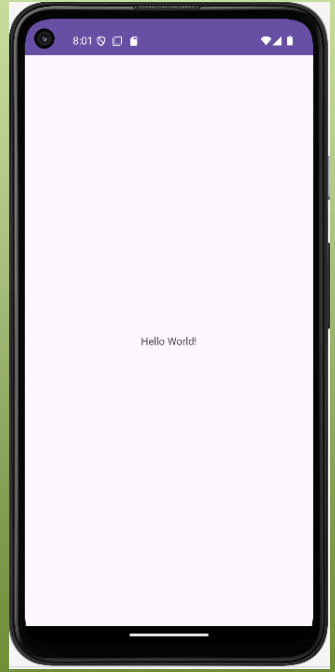
- Posteriormente, conectamos nuestro dispositivo al ordenador, aceptamos los permisos necesarios y vemos que aparece nuestro dispositivo disponible para ejecutar nuestra app:



4.- Ejecución de aplicaciones en dispositivos físicos

- Seleccionamos nuestro dispositivo, ejecutamos la aplicación y esta se instalará y lanzará en nuestro teléfono en vez de en el emulador.
- *<u>Nota</u>: Es necesario dar permisos para poder instalar la app en nuestro sistema operativo Android.*





Dispositivo físico