

# PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES



## TEMA 2:

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

# ÍNDICE

1. Instalación de Android Studio.
2. Estructura de un proyecto.
3. Creación de un emulador.
4. Ejecución de aplicaciones en dispositivos físicos.



# PMDM

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



## 1. Instalación de Android Studio

# 1.- Instalación de Android Studio

- Paso 1:** 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.

**Oracle** Productos Sectores Recursos Clientes Partners Desarrolladores Compañía

**Java Downloads**

Java downloads Tools and resources Java archive

Looking for other Java downloads? [OpenJDK Early Access Builds](#) [JRE for Consumers](#)

**Java 20 and Java 17 available now**

JDK 20 is the latest release of Java SE Platform and JDK 17 LTS is the latest long-term support release for the Java SE platform. [Learn about Java SE Subscription](#)

JDK 20 JDK 17 GraalVM for JDK 20 GraalVM for JDK 17

**JDK Development Kit 20.0.2 downloads**

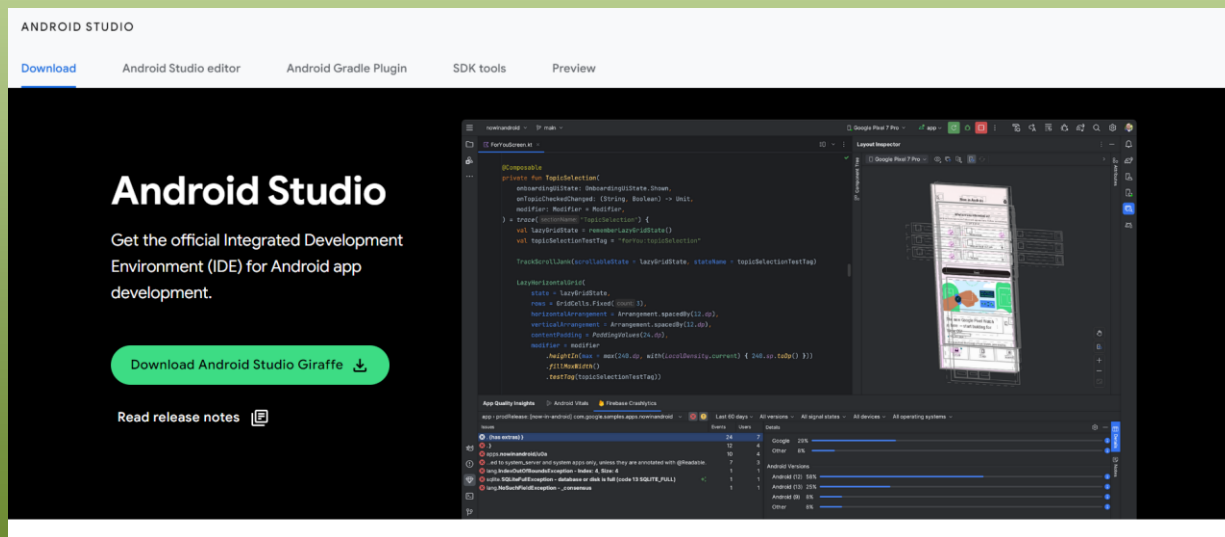
JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions. JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	180.99 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.zip">https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.zip</a> (sha256)
x64 Installer	160.12 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe">https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe</a> (sha256)

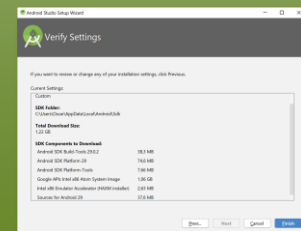
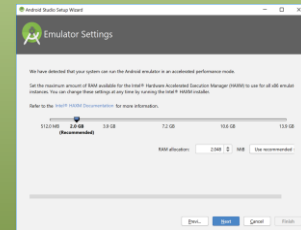
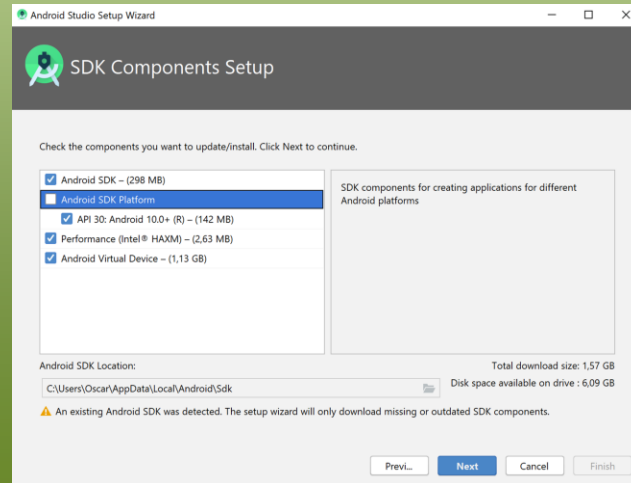
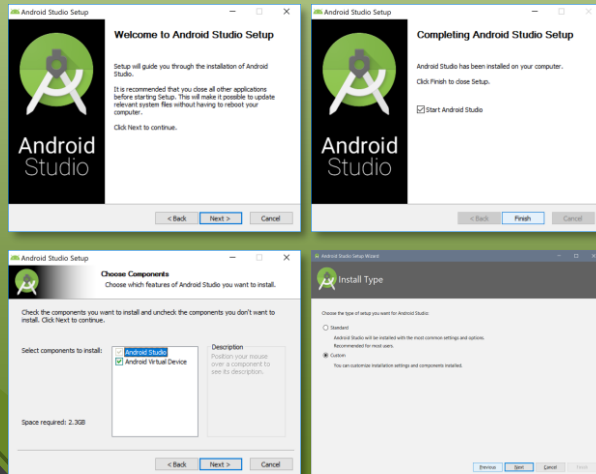
# 1.- Instalación de Android Studio

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



# 1.- Instalación de Android Studio

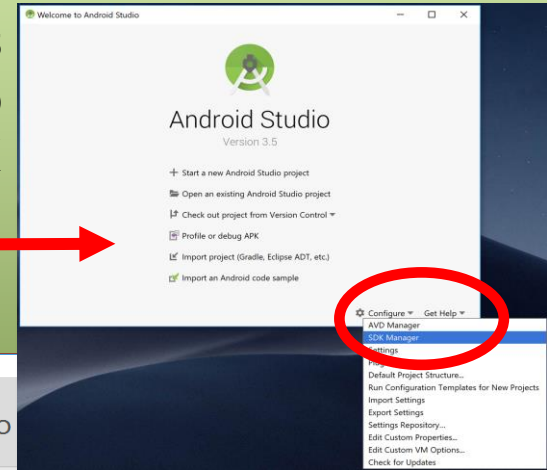
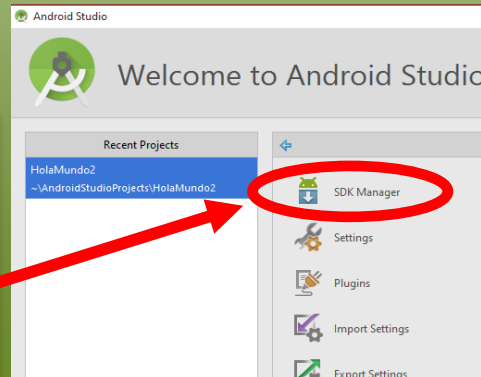
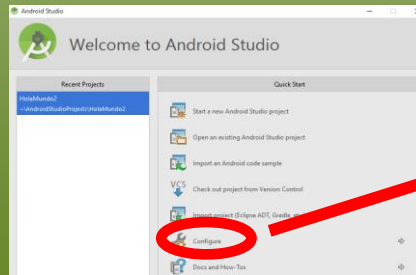
- Durante la instalación, si dispones de un procesador Intel en tu ordenador, no olvides marcar la casilla de Intel HAXM (*Hardware Accelerated Execution Manager*), que es un sistema de virtualización que ayudará a mejorar el rendimiento del emulador de Android:



# 1.- Instalación de Android Studio

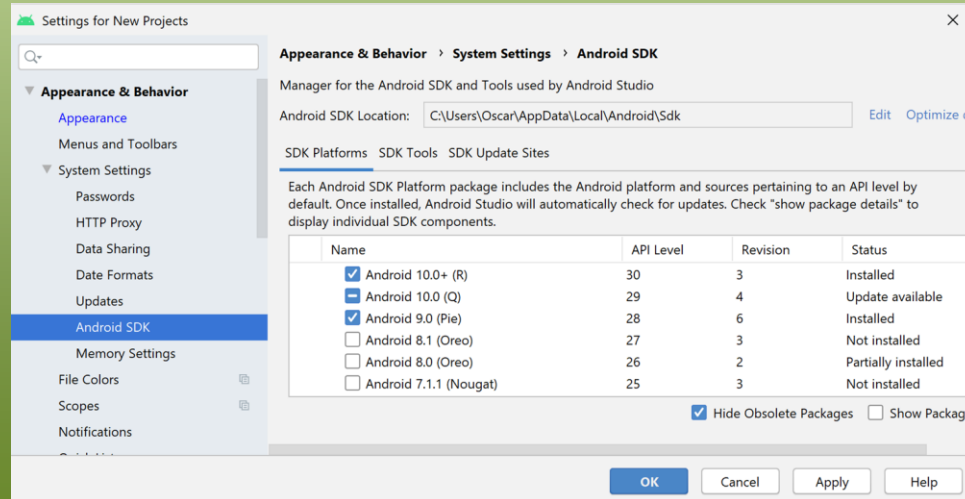
- **Paso 3:** Instalación y actualización de los componentes SDK de Android: Para ello accederemos al menú "Configure / SDK Manager" de la pantalla de bienvenida:

Si ya hemos creado un proyecto anteriormente, también podemos acceder así:



# 1.- Instalación de Android Studio

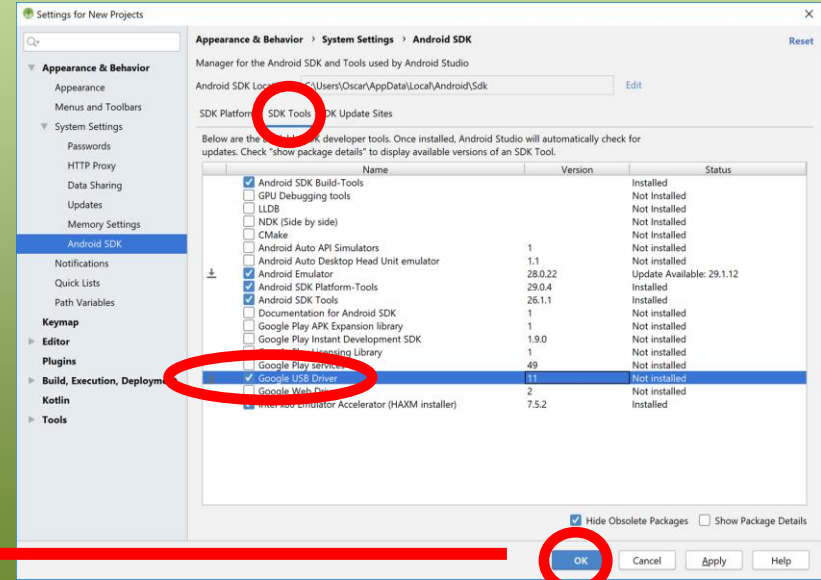
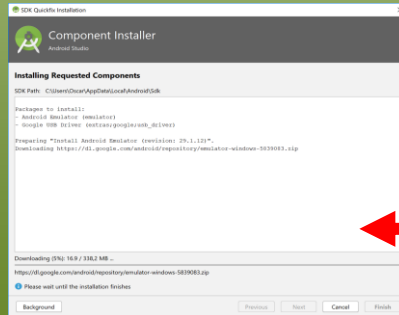
- Si queremos que la aplicación que desarrollemos sea compatible con versiones anteriores de Android, deberemos seleccionar las API's deseadas. En la ventana de un proyecto, esta configuración se encuentra en **"File -> Settings -> Android SDK"**.





# 1.- Instalación de Android Studio

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

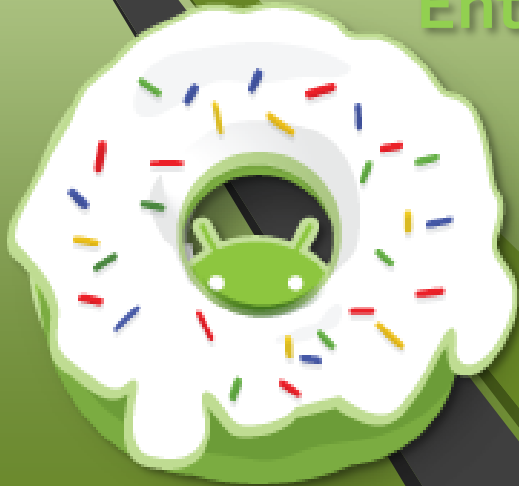


# 1.- Instalación de Android Studio

- Para más información sobre la instalación y configuración del entorno visita:
  - <https://developer.android.com/studio/install?hl=es-419>
  - Configurar Entorno de Desarrollo: <https://youtu.be/96ydnsCpyq4>
  - Instalación Android Studio parte 1: [https://youtu.be/\\_5OHCMEo4lk](https://youtu.be/_5OHCMEo4lk)
  - Instalación Android Studio parte 2: [https://youtu.be/JVNXsofVT\\_o](https://youtu.be/JVNXsofVT_o)

# PMDM

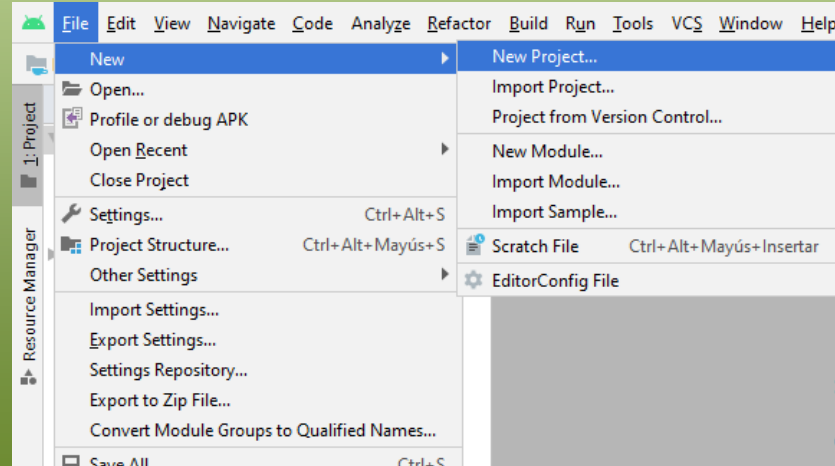
Instalación, Configuración Y Uso Del  
Entorno De Desarrollo Android Studio.  
Emuladores.



**2. Estructura de un proyecto**

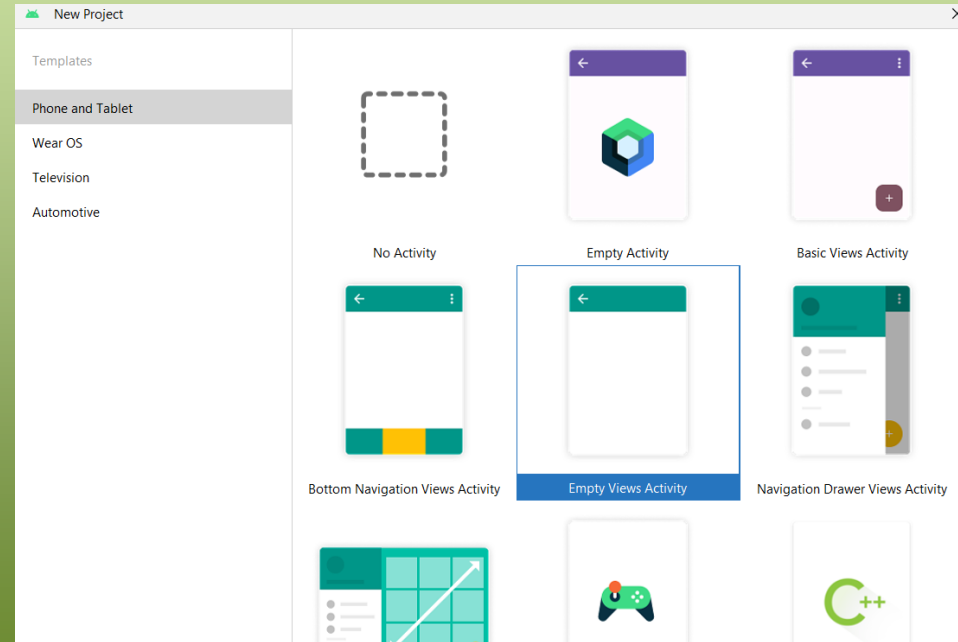
## 2.- Estructura de un proyecto

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



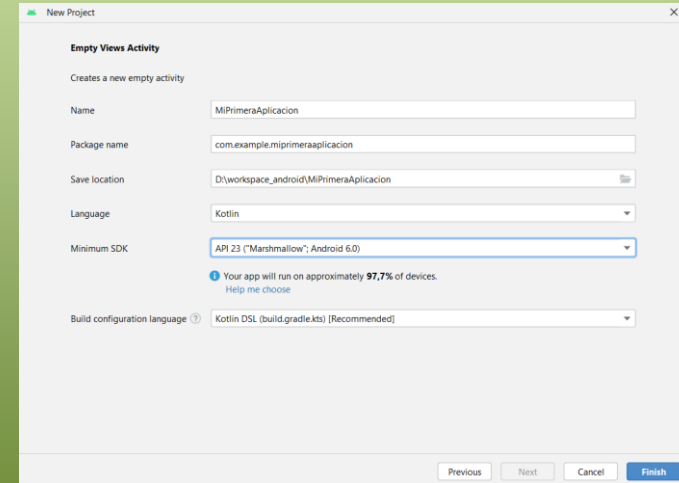
## 2.- Estructura de un proyecto

- 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.



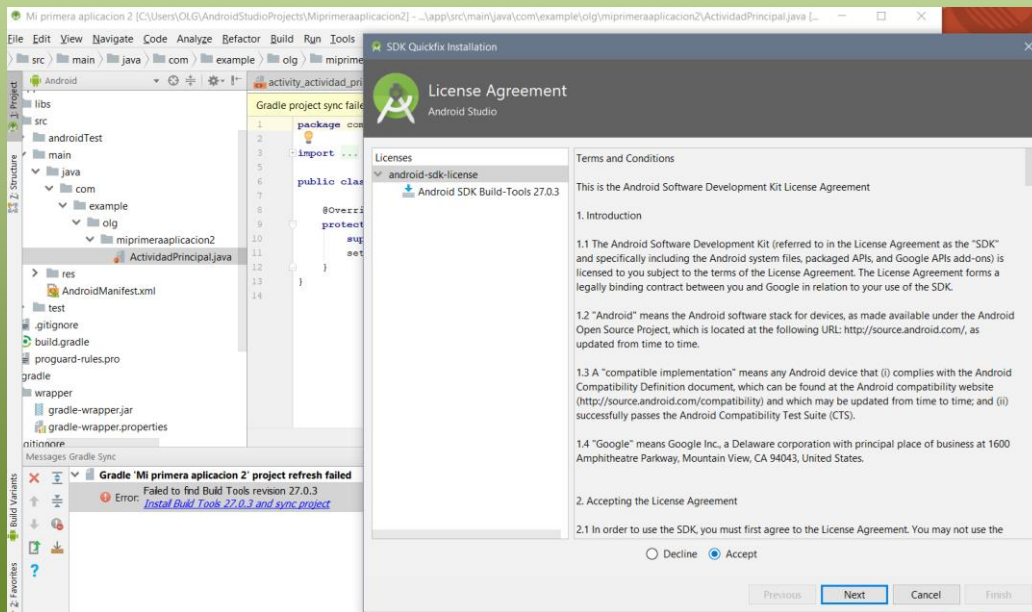
## 2.- Estructura de un proyecto

- 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



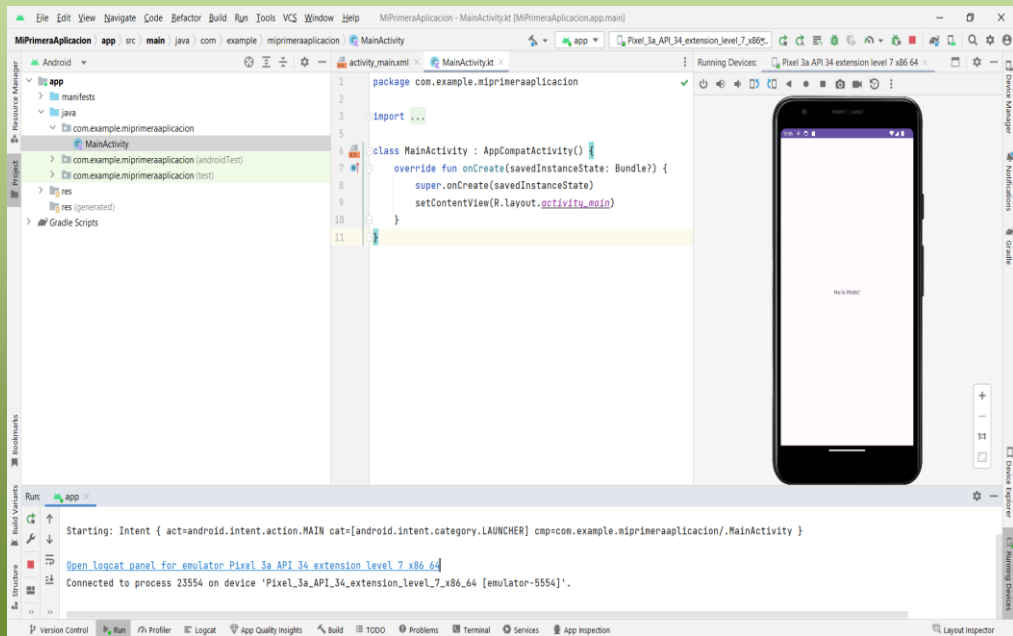
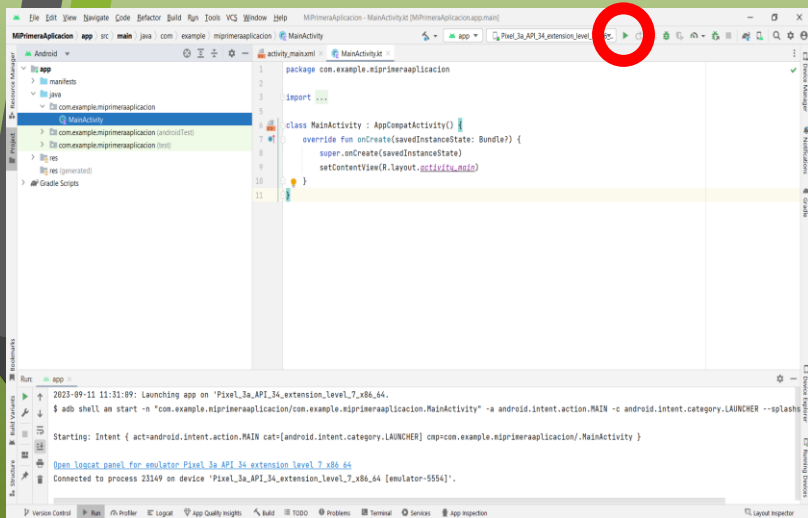
## 2.- Estructura de un proyecto

- Es probable tener que actualizar el Gradle antes de continuar:



## 2.- Estructura de un proyecto

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



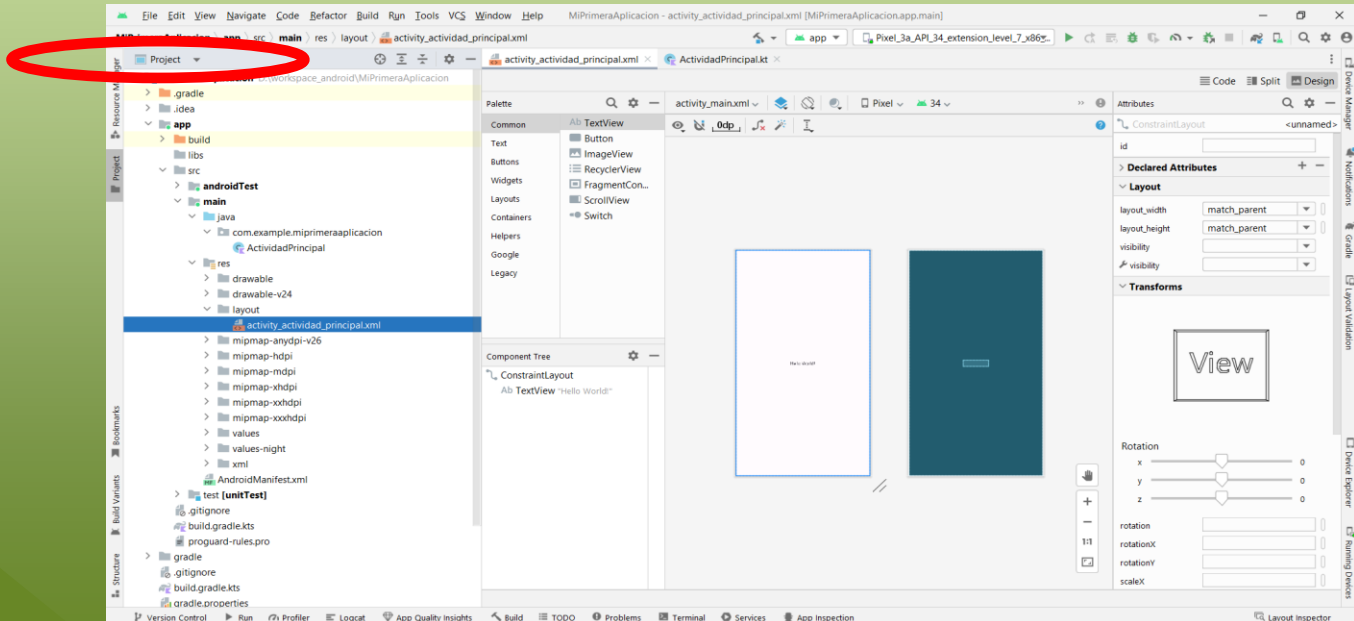


# EJERCICIOS

- **Ejercicio 01.** Crea un proyecto Android llamado *Ejercicio1T2*, ejecútalo y comprueba que se abre correctamente el emulador.

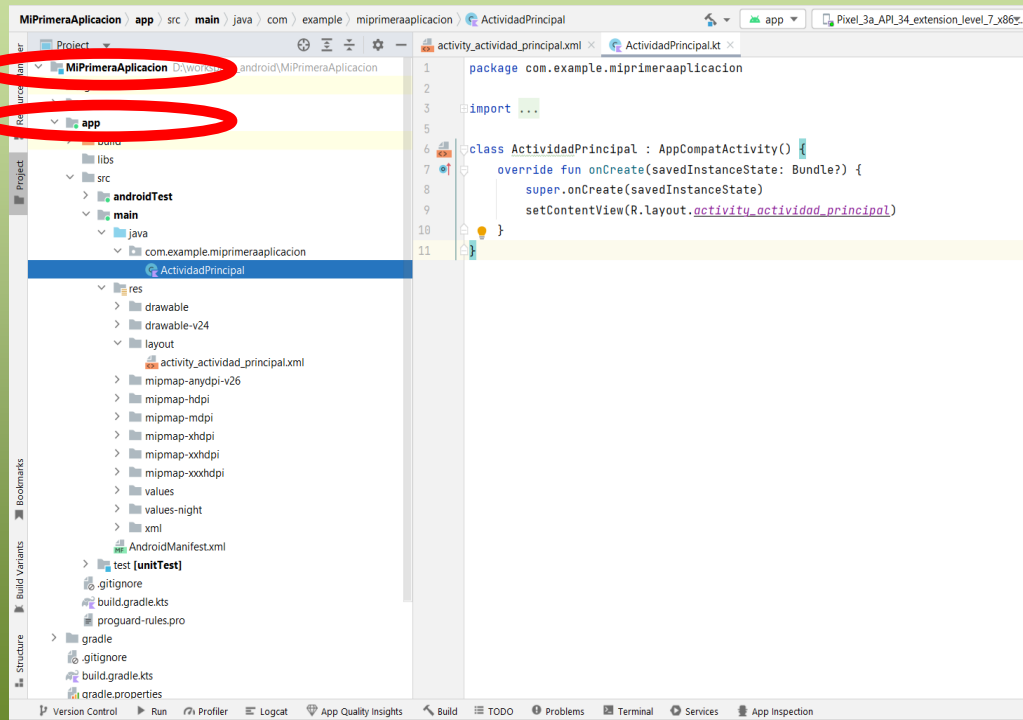
## 2.- Estructura de un proyecto

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

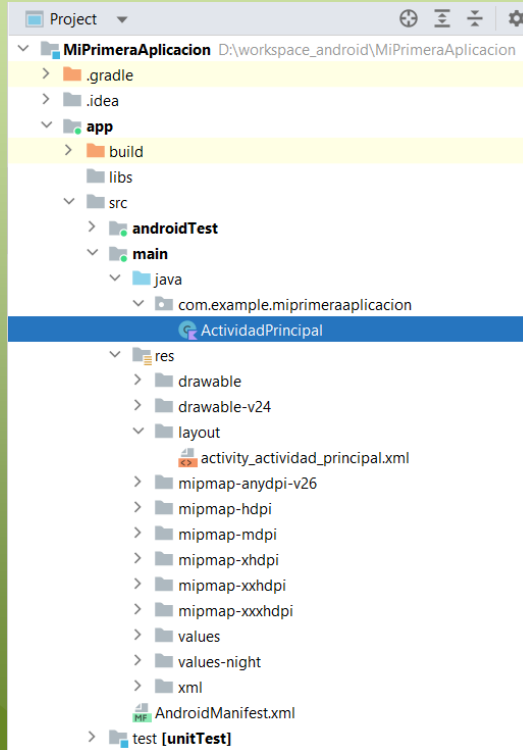


## 2.- Estructura de un proyecto

- 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 "MiPrimeraAplicacion" que contiene al módulo "app" que contendrá todo el software de la aplicación.



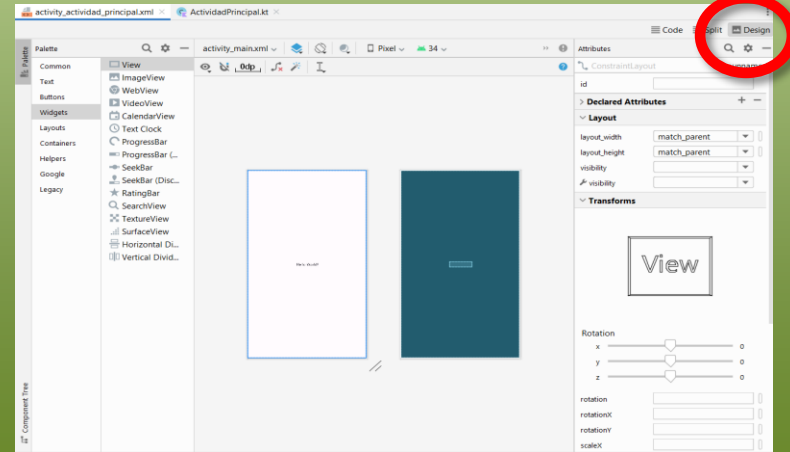
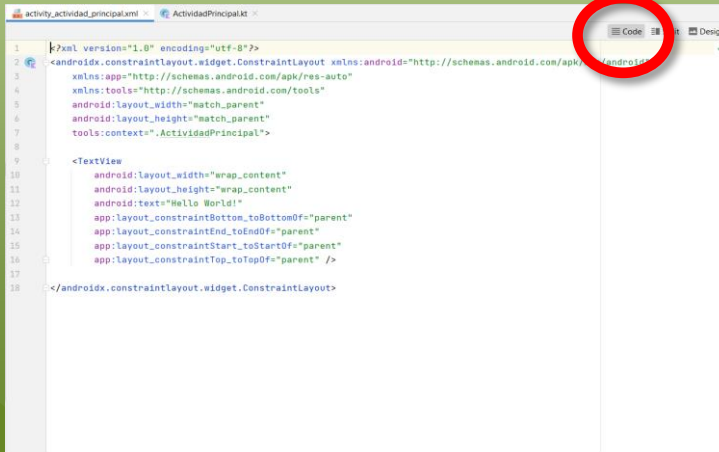
## 2.- Estructura de un proyecto



- **/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_actividad_principal.xml`.

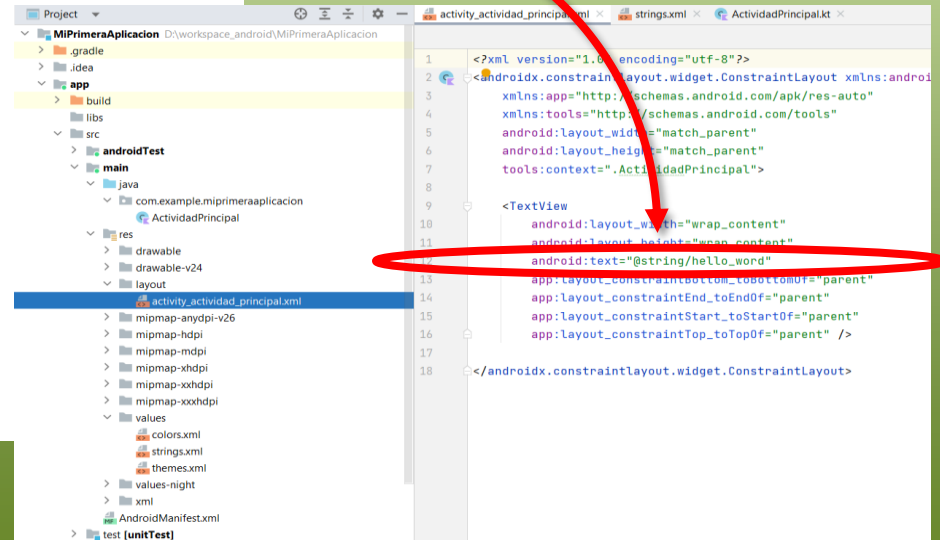
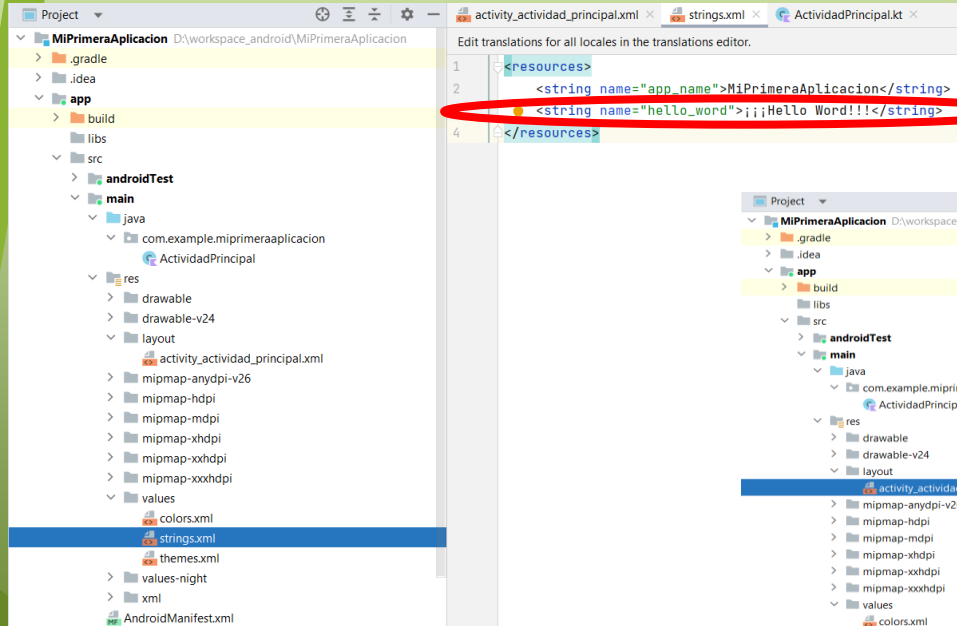


## 2.- Estructura de un proyecto

- **/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



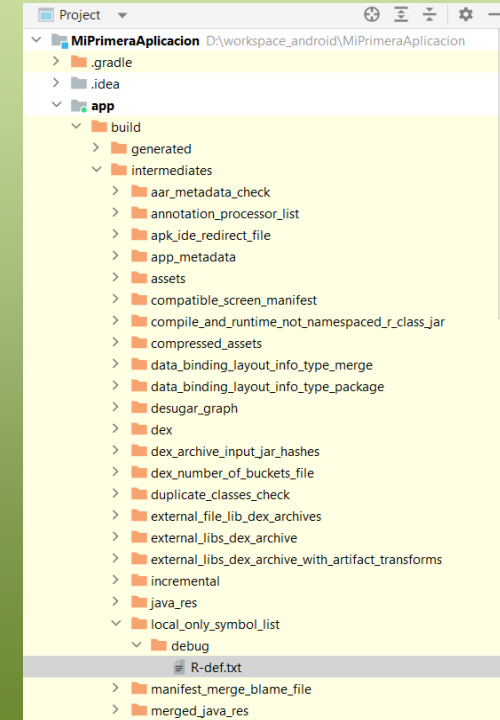
## 2.- Estructura de un proyecto

- **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, empleo 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", relativa a la clase R. 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/`, de forma que podamos 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" contenido en la carpeta `/app/src/main/res/layout/`.

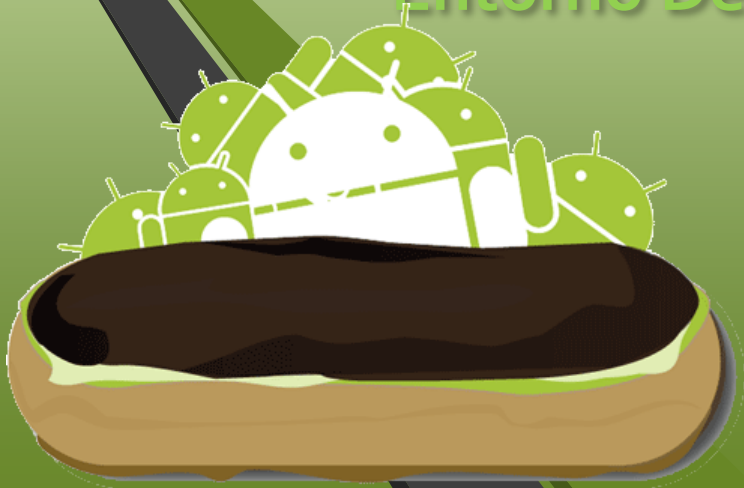


## 2.- Estructura de un proyecto

- Para más información sobre la estructura de los proyectos Android Studio visita:
  - <http://www.sgoliver.net/blog/estructura-de-un-proyecto-android-android-studio/>
  - Estructura de un proyecto en Android: <https://youtu.be/XWvT6OZH Zoe>

# PMDM

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

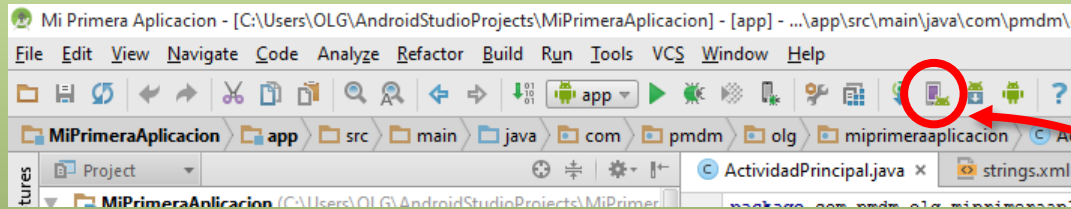


## 3. Creación de un emulador

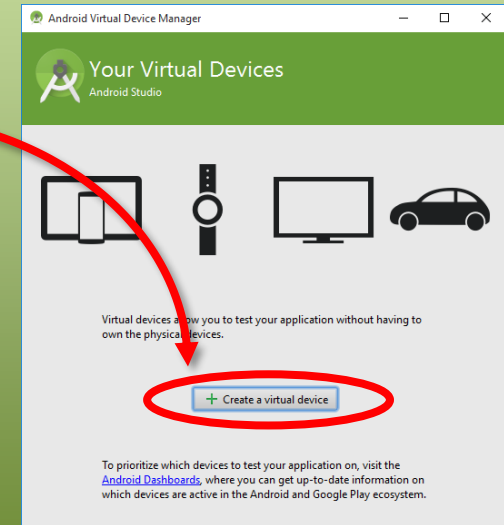
## 3.- Creación de un emulador

- 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

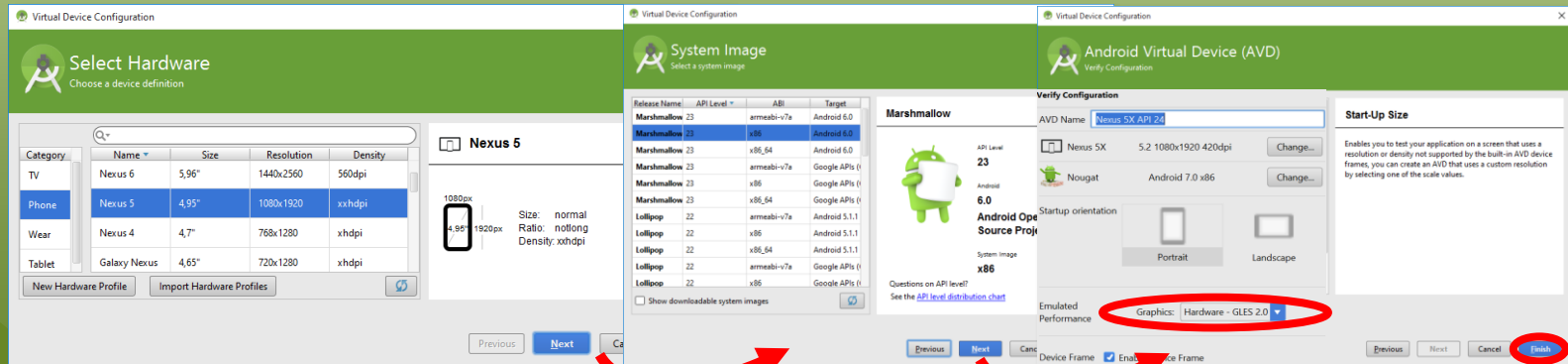


- Si en la instalación no elegimos la opción de instalar también un emulador, tendremos que crearlo. Para crear un emulador de un dispositivo utilizaremos el AVD Manager:



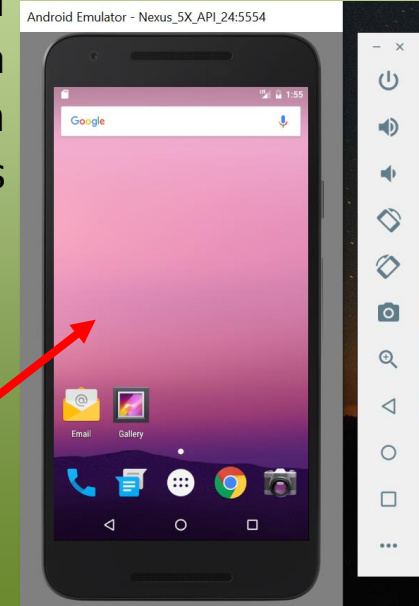
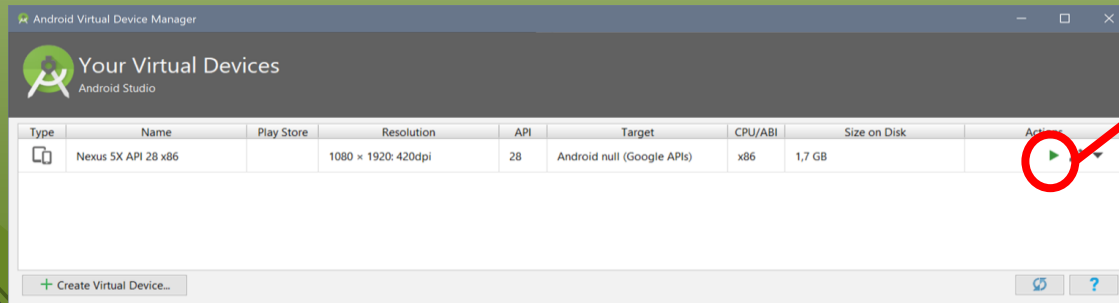
## 3.- Creación de un emulador

- Elegiremos una configuración que nos guste, por ejemplo, un dispositivo Pixel 2, con la última versión de Android, la 10 (o Q) que lleva la **API 29** y un procesador **x86** (para que el emulador funcione de manera fluida). Por último, si tenemos una buena tarjeta gráfica, podremos habilitar la aceleración hardware-GLES 2.0:



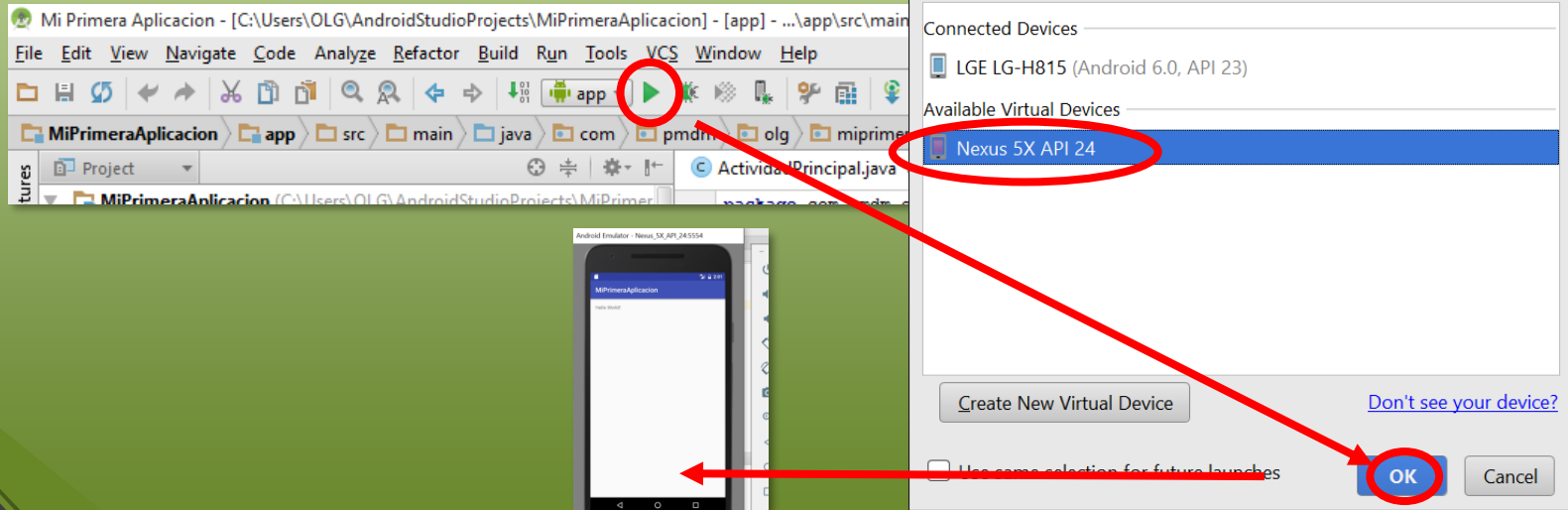
## 3.- Creación de un emulador

- Para dejar iniciado el emulador, para posteriormente probar nuestras aplicaciones en él, pulsamos en el triángulo 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, cuando ejecutemos nuestra aplicación, nos pedirá que le indiquemos en que dispositivo queremos ejecutarla:





## 2.- Estructura de un proyecto

- Para más información sobre otros emuladores Android bajo Pc (Bluestacks, Genymotion, Nox, ARChon...) visita:
  - <https://vivantic.org/mejores-emuladores-android-pc/>

# PMDM

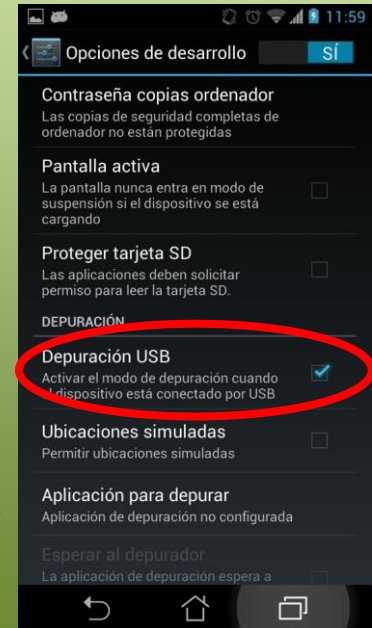
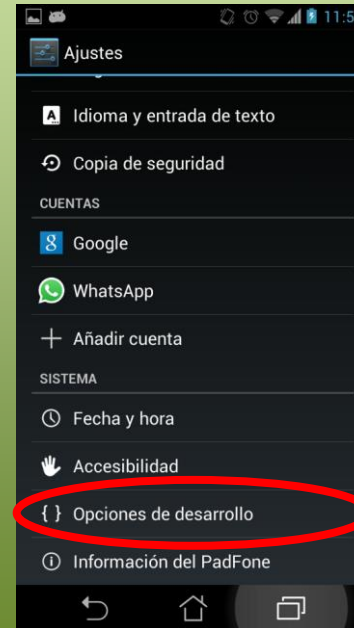
Instalación, Configuración Y Uso Del  
Entorno De Desarrollo Android Studio.  
Emuladores.



4. Ejecución de aplicaciones  
en dispositivos físicos

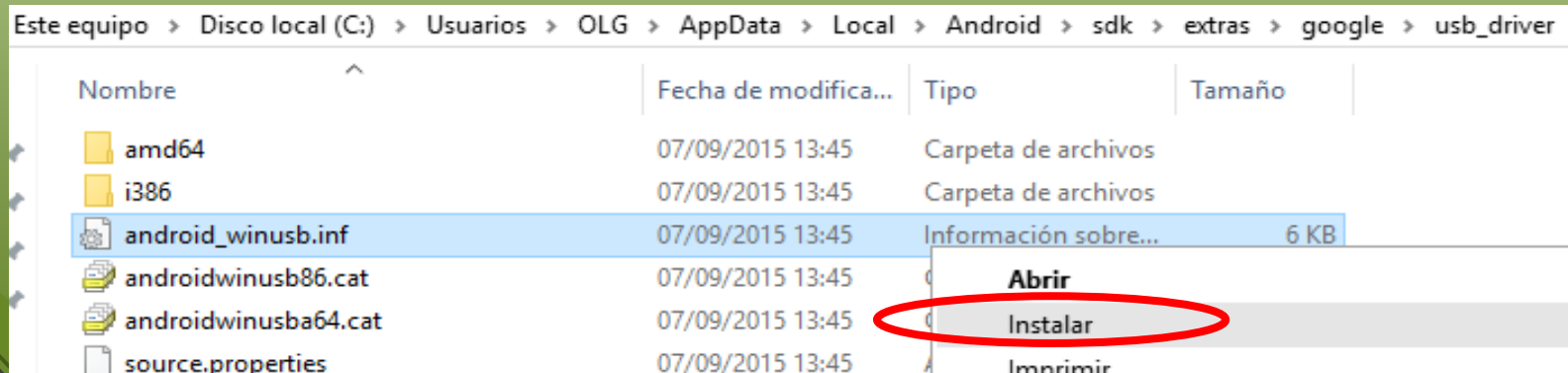
## 4.- Ejecución de aplicaciones en dispositivos físicos

- 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->Estado.



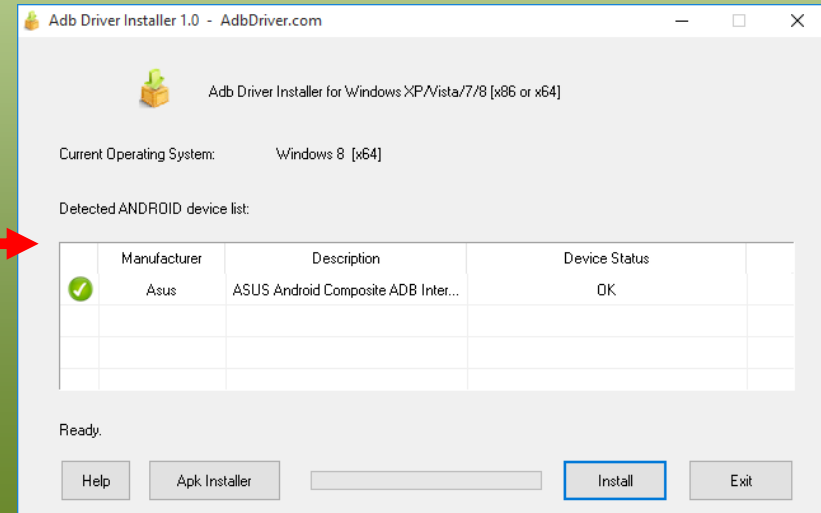
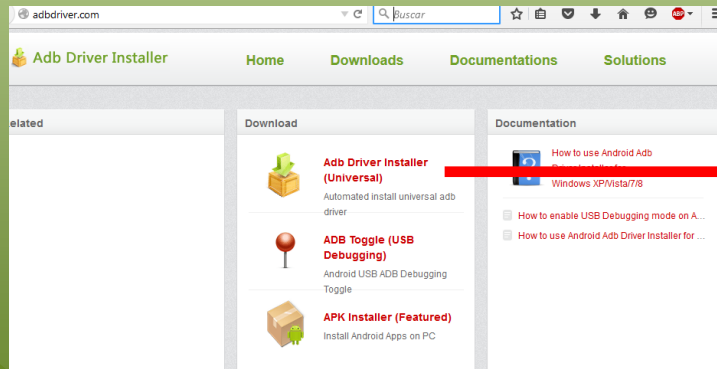
## 4.- Ejecución de aplicaciones en dispositivos físicos

- El siguiente paso es conectar nuestro dispositivo por USB e instalar el driver ADB USB en nuestro ordenador. La forma más sencilla es desde el archivo android\_winusb.inf que lo encontrarás dentro de la carpeta donde hayas instalado el SDK de Android (por defecto está en: **C:\Users\NombreUsuario\AppData\Local\Android\sdk\extras\google\usb\_driver**)



## 4.- Ejecución de aplicaciones en dispositivos físicos

- Otra manera de instalar el driver ADB USB en nuestro ordenador es descargarlo de <http://developer.android.com/sdk/win-usb.html> o también de <http://adbdriver.com/> e instalarlo.



## 4.- Ejecución de aplicaciones en dispositivos físicos

- Al ejecutar nuestra aplicación deberá aparecer nuestro dispositivo Android conectado por USB:

