

Conceptos básicos y entornos de desarrollo

1. Características de Android	2
2. Arquitectura de Android	2
3. Versiones de Android	3
4. Android Studio.....	4
4.1. Instalación de la JDK (Java Development kit).....	4
4.2. Instalación Android Studio	4
4.3. Ejecutando Android Studio	5
4.4. Interfaz	6
4.5. SDK Manager	8
4.6. AVD.....	9
5. Creando un proyecto Android.....	10
6. Estructura de un proyecto Android.....	10

1. Características de Android

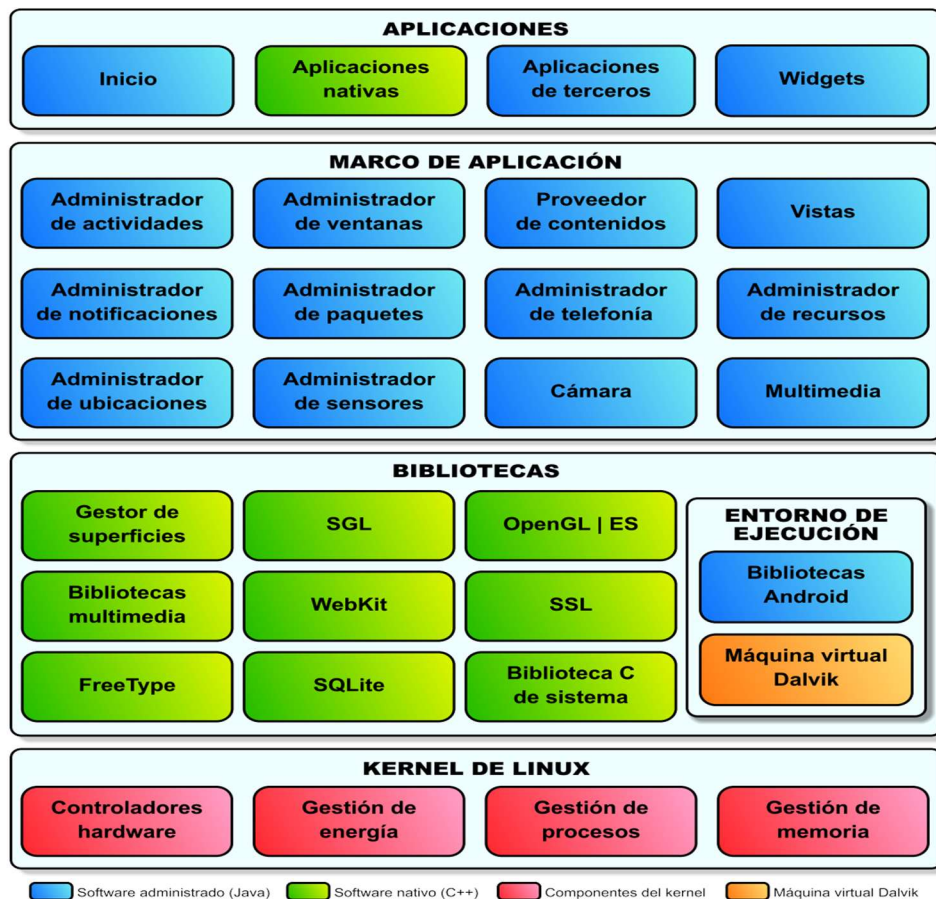
Aunque existen varias plataformas para móviles, el sistema operativo para dispositivos móviles Android ofrece una serie de características que lo diferencian de todos ellos:

- Plataforma abierta. Es una plataforma de desarrollo libre basada en Linux y de código abierto. Entre sus ventajas se encuentra la de poder usar y adaptar el sistema sin pagar royalties.
- Portabilidad asegurada. Gracias al concepto de máquina virtual las aplicaciones finales se desarrollan en Java, asegurando la ejecución y funcionamiento en gran variedad de dispositivos, tanto actuales como futuros.
- Arquitectura basada en componentes inspirados en Internet. Por ejemplo, el diseño de la interfaz de usuario se hace en XML, permitiendo que una misma aplicación se ejecute en un móvil de pantalla reducida o en un Netbook.
- Filosofía de dispositivo siempre conectado a Internet.
- Gran cantidad de servicios incorporados. Por ejemplo, localización basada en GPS, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, mapas, etc.
- Aceptable nivel de seguridad. Los programas están aislados unos de otros gracias al concepto de ejecución dentro de una caja que incorpora la máquina virtual.
- Optimización para baja potencia y poca memoria.
- Alta calidad de gráficos y sonido. Gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3D basados en OpenGL, códecs estándar de audio y vídeo.

2. Arquitectura de Android

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (*surface manager*), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C.

El siguiente gráfico muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en *software* libre.



Android implementa un sistema operativo basado en Linux y su máquina virtual (Dalvik) compila los archivos de Java. Las aplicaciones de Android se distribuyen en formato .apk. Un apk contiene clases de Java compiladas en un formato de bytecode conocido como DEX. El formato DEX es independiente de arquitectura de procesador, por lo que para ser ejecutado necesita traducirse a “código máquina” nativo para el procesador de nuestro dispositivo. Lógicamente la ejecución es más lenta con respecto a la ejecución directa.

Como solución a este problema, Google decidió desde la versión KitKat 4.4 ir introduciendo la nueva máquina virtual ART.

La diferencia fundamental entre Dalvik y ART es que Dalvik utiliza compilación “justo a tiempo” (just-in-time, o JIT), mientras que ART utiliza compilación previa (ahead-of-time, o AOT). Dalvik ejecuta una máquina virtual que interpreta el código a la vez que se ejecuta la aplicación mientras que ART realiza una pre-compilación durante el proceso de instalación de la aplicación.

Entre otras, también presenta mejoras en el recolector de basura y en el desarrollo y depuración de aplicaciones.

3. Versiones de Android

Android tiene numerosas versiones que arreglan fallos anteriores y agregan nuevas funciones. El nombre de cada versión era el de un dulce en orden alfabético hasta el 23

de agosto del 2019, fecha en que Google notificó que Android dejaría de tener nombres basados en postres y que numeraría las siguientes versiones:

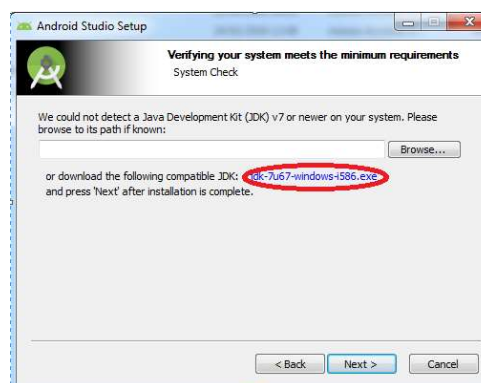
Android 11.0	8 de Septiembre 2020
Android 10.0	3 de Septiembre 2019
Android 9.0 Pie	6 de Agosto 2018
Android 8.0 Oreo	21 de Agosto 2017
Android 7.0 Nougat	Agosto 2016
Android 6.0 Marshmallow	Octubre 2015
Android 5.0 Lollipop	Noviembre 2014
Android 4.4 KitKat	Noviembre 2013
Android 4.3 Jelly Bean	Julio de 2013
Android 4.2 Jelly Bean	Noviembre 2012
Android 4.1 Jelly Bean	Julio de 2012
Android 4.0 Ice Cream Sandwich	Octubre de 2011
Android 3.0 / 3.1 / 3.2 Honeycomb	Febrero de 2011
Android 2.3 Gingerbread	Diciembre 2010
Android 2.2 Froyo	Mayo 2010
Android 2.0 Eclair	Octubre 2009
Android 1.6 Donut	Septiembre 2009
Android 1.5 Cupcake	Abril 2009
Android 1	Septiembre 2008

El problema es que muchas veces las nuevas versiones no funcionan correctamente en el hardware diseñado para versiones previas.

4. Android Studio

4.1. Instalación de la JDK (Java Development kit)

Este software nos permitirá ejecutar código Java en el equipo. Al instalar Android Studio y si no se tiene la versión adecuado de JDK, el propio instalador lo indica:



Al hacer clic nos redirige a la página desde la cual podemos efectuar la descarga.

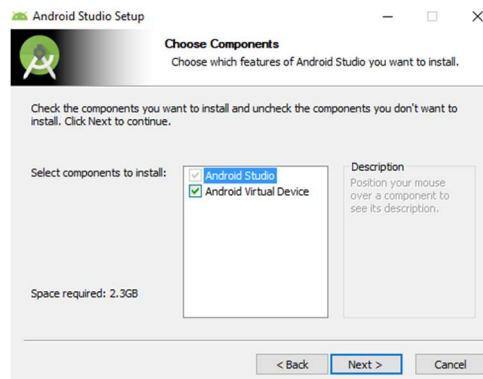
4.2. Instalación Android Studio

Android Studio es la plataforma de Google. Se trata de un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) pensado para el sistema operativo Android.

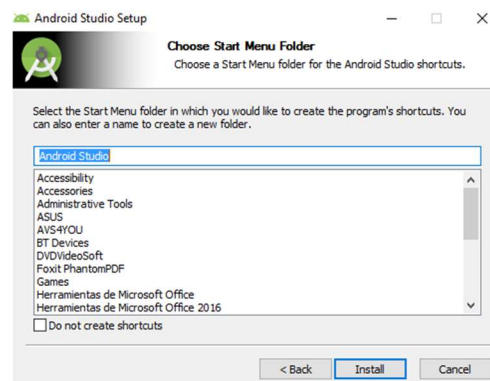
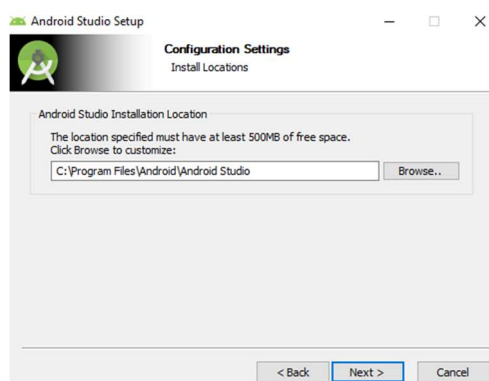
Este entorno ofrece un editor de diseño que permite arrastrar y soltar los componentes en la interfaz de usuario, presentaciones de vista previa de múltiples configuraciones de pantalla, asistentes basados en una plantilla para crear diseños y componentes comunes Android, así como una serie de herramientas para detectar rendimiento, facilidad de uso, compatibilidad de versiones y otros problemas.

Se puede descargar de <https://developer.android.com/studio/>. Nosotros trabajaremos con la versión para Windows, pero existen versiones para otros sistemas operativos.

Al ejecutar el fichero que nos hemos descargado muestra la siguiente pantalla:



Y donde se va a ubicar:

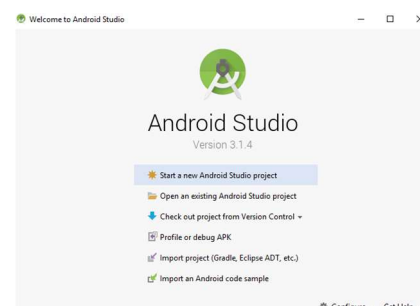


Le damos a Install para instalar Android Studio.

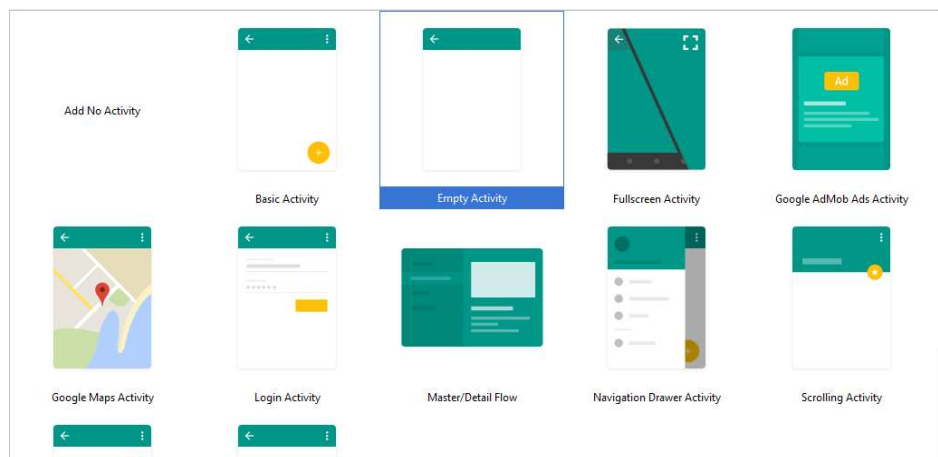
4.3. Ejecutando Android Studio

Al ejecutar por primera vez aceptamos en todas las ventanas que nos irán apareciendo para poder configurar Android Studio.

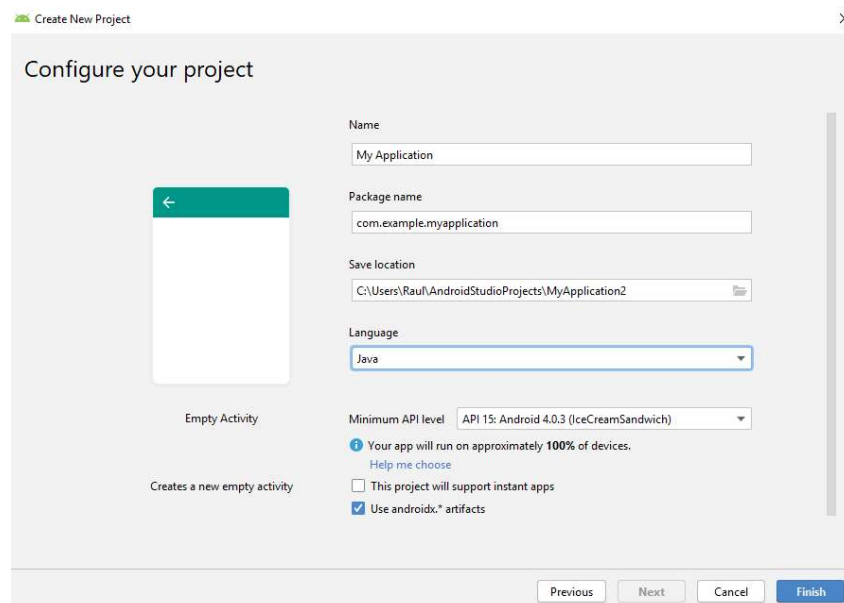
Al abrir Android Studio nos aparece:



También hay que seleccionar la apariencia que tendrá nuestra aplicación. En un principio seleccionamos la opción más básica:



Y por último aparece una nueva ventana donde se pide el nombre, la localización del proyecto, el lenguaje a utilizar y la versión mínima de Android con la que vamos a trabajar:



Como lenguaje permite elegir entre Java y Kotlin. Kotlin es un lenguaje de programación que ha sido desarrollado por JetBrains, la misma compañía responsable de IntelliJ IDE, el entorno de desarrollo en que se basa Android Studio lo que hace Kotlin esté perfectamente integrado en dicho entorno de desarrollo.

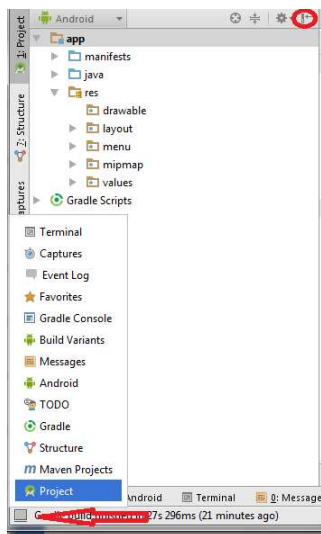
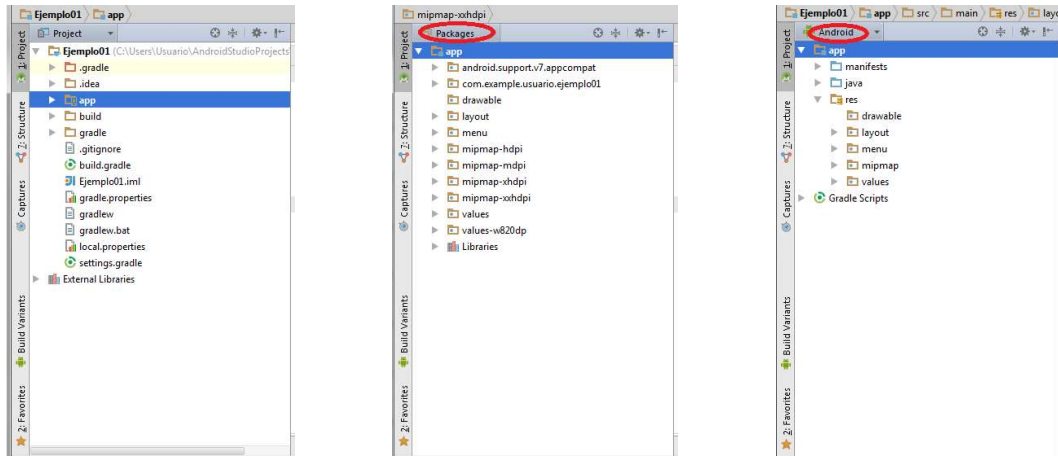
Kotlin presenta otras muchas ventajas con respecto a programar en Java. En principio usaremos Java pero también veremos aplicaciones con Kotlin más adelante.

4.4. Interfaz

Android Studio está basado en el entorno IntelliJ IDEA: asistente de configuración muy sencillo, importación de ejemplos y plantillas, edición y previsualización de diseños de

Android a través de múltiples tamaños de pantalla, idiomas e incluso versiones de API, análisis del rendimiento de las aplicaciones, ...

En la parte izquierda existe un panel (tool window según la nomenclatura IntelliJ) donde se muestra el árbol correspondiente a nuestro proyecto. Si deseamos cambiar la información mostrada podemos hacerlo con el panel desplegable que aparece en la parte superior:



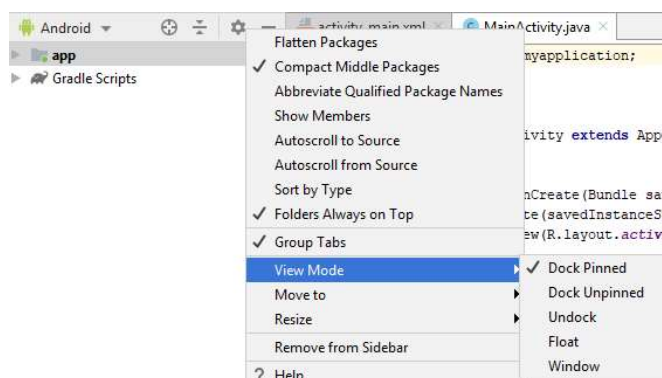
En la parte inferior izquierda hay un botón que, si pasamos el ratón por encima, muestra los paneles que se pueden desplegar. Los paneles que aparecen depende del tipo de fichero que esté en el panel central.

Se se pulsa dicho botón aparecerá o desaparecerá el marco que hay alrededor del entorno de programación y que permite el acceso directo a los diferentes paneles.

En la parte superior derecha tenemos un icono que oculta el panel, si así lo queremos.

También existe un desplegable con las siguientes opciones:

- Dock Pinned: Siempre visible.
- Dock Unpinned: El panel se oculta cuando pierde el foco pero cuando está visible se adapta al resto de ventanas.
- Undock: El panel se oculta cuando pierde el foco pero cuando está visible no se adapta al resto de ventanas.
- Float: Se visualiza en una nueva ventana.



4.5. SDK Manager

Android cuenta con varias versiones y no todas tienen las mismas posibilidades.

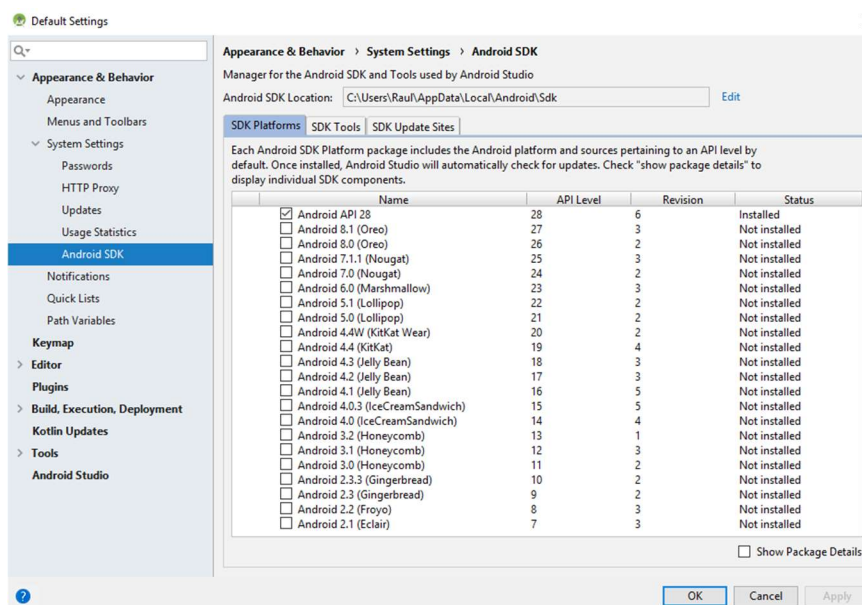
Cuando se crea una aplicación hay que tener en cuenta la versión con la que queremos trabajar y para ello existe la utilidad "SDK Manager":



Android SDK es un paquete que contiene:

- Las librerías específicas de Android
- El compilador
- Emulador Android, para poder ejecutar las aplicaciones sin necesidad de un dispositivo físico
- El entorno de depuración DDMS

Al ejecutar el programa SDK Manager aparece una ventana donde podremos seleccionar los paquetes a instalar. Aparece la última versión instalada y se pueden añadir más conforme se vayan necesitando.



SDK Tools incluye herramientas para el desarrollo y depuración de aplicaciones Android.

El historial de versiones de Android se inició con el lanzamiento de Android beta en noviembre de 2007. La primera versión comercial, Android 1.0, fue lanzada en

septiembre de 2008 y a partir de entonces han aparecido diferentes versiones, cada una con un nombre de postre diferente y en orden alfabético: Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat, Oreo y Pie. Estas actualizaciones típicamente corrigen fallos de programa y agregan nuevas funcionalidades.

Las últimas versiones incluyen numerosos cambios como el uso de un nuevo Android runtime (ART) por defecto, en sustitución del actual, Dalvik.

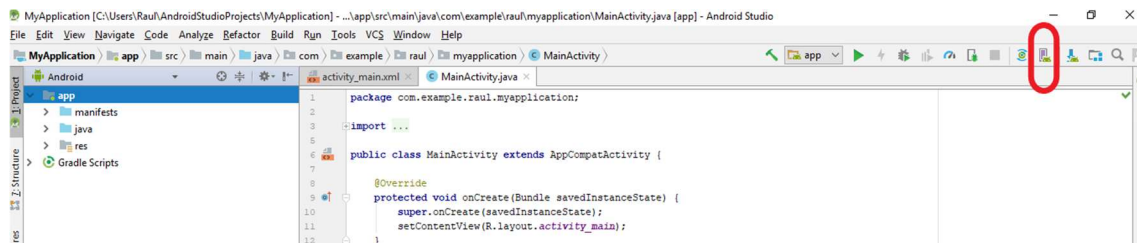
Android 4.4 es una versión de Android Wear. Wearable hace referencia al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo interactuando continuamente con el usuario y con otros dispositivos con la finalidad de realizar alguna función específica: relojes inteligentes o smartwatches, zapatillas de deportes con GPS incorporado o pulseras que monitorizan nuestro estado de salud son algunos ejemplos.

Si más adelante queremos instalar versiones posteriores de Android habrá que actualizar las SDK Tools para que funcione correctamente.

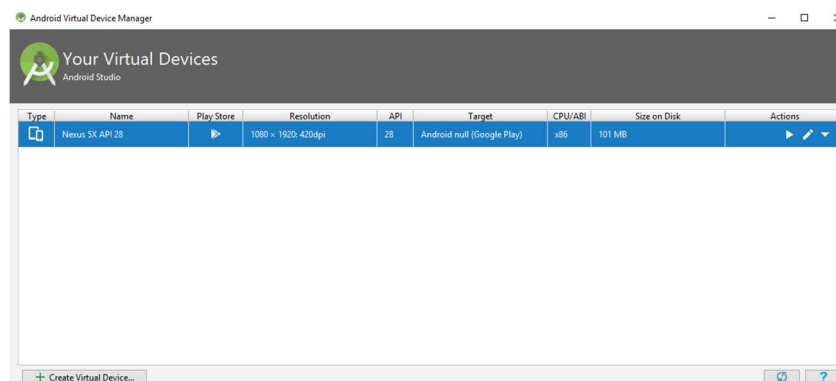
4.6. AVD

A continuación, crearemos un nuevo dispositivo virtual Android (AVD).

Un dispositivo virtual Android permite emular en tu ordenador cualquier tipo de dispositivos móvil. De esta forma podrás probar tus aplicaciones en gran variedad de teléfonos y tabletas. Para ello abrimos el *AVD Manager*:



Donde podemos crear todos los emuladores que queramos:



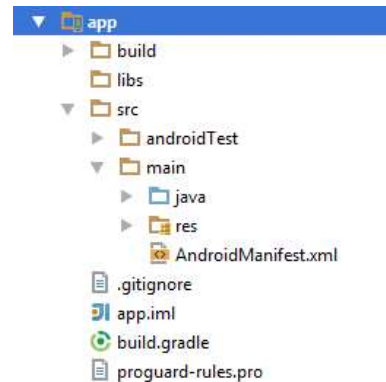
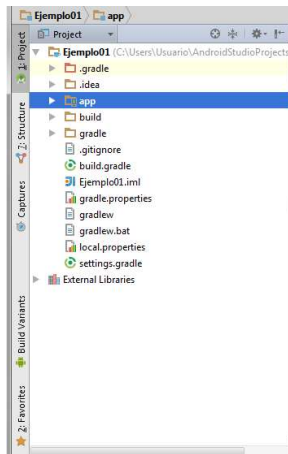
Para crear nuevos emuladores hay que seguir los pasos que se detallan en <https://developer.android.com/studio/run/managing-avds>.

5. Creando un proyecto Android

Para crear nuestra primera aplicación vamos a seguir las indicaciones que aparecen en <https://developer.android.com/training/basics/firstapp/>.

6. Estructura de un proyecto Android

Teniendo seleccionada la opción *Project* podemos comprobar que existe un nodo de nombre *app* que contiene:



build	Subdirectorio que no se debe tocar ya que es donde se guardan todos los ficheros compilados en formato dex, los recursos ya preparados, ... Cada vez que modificamos el proyecto se actualiza.		
libs	Subdirectorio que contiene las librerías que usa nuestro proyecto.		
src	androidTest	Subdirectorio donde se construye nuestro programa de pruebas	
	main	java	Subdirectorio que contiene el código fuente de la aplicación
		res	Subdirectorio que contiene los recursos de nuestra aplicación
		AndroidManifest.xml	Archivo de configuración.
.gitignore	Repositorio central en el caso de que existan varias personas trabajando en el mismo fichero.		

app.iml	Fichero de configuración del módulo propio del entorno IntelliJ IDEA.
build.gradle	Fichero de configuración de la utilidad Gradle, que es la encargada de compilar la aplicación.
proguard-rules.pro	Herramienta que optimiza el código