

UNIDAD 1

INSTALACIÓN DE ANDROID STUDIO

En el desarrollo de aplicaciones Android se pueden utilizar diversos IDE (Entorno de Desarrollo Integrado) como Eclipse con ADT (Android Developer Tools) ó NetBeans con NBAndroid.

En el módulo "Programación Multimedia y de Dispositivos Móviles" vamos a utilizar Android Studio que es el IDE oficial para el desarrollo de aplicaciones Android y es el entorno recomendado por Google.



Android Studio incorpora todas las herramientas necesarias para desarrollar, generar ejecutables y realizar pruebas sobre las aplicaciones Android.

- Potente editor de código que permite trabajar con diferentes lenguajes incluido Java que será el lenguaje utilizado en nuestras aplicaciones.
- Panel de diseño para crear o modificar la interfaz gráfica de nuestras aplicaciones sin necesidad de escribir código.
- Gestor de SDK que nos va a permitir obtener y configurar tanto la imagen del sistema Android sobre el que vamos a trabajar como las herramientas necesarias.
- Herramientas para la compilación y la generación de paquetes (apk).
- Emulador de Android para probar nuestras aplicaciones. Aunque más que un simple emulador, vamos a disponer de una herramienta con la que crear y configurar distintos emuladores (teléfonos, tabletas, wearables, televisiones) de diferentes modelos y recursos sobre los que realizar las pruebas.

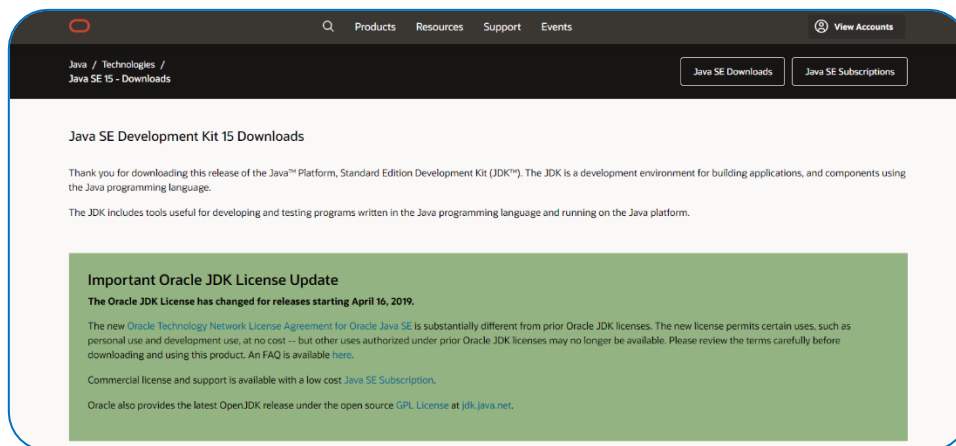
REQUISITOS PREVIOS: JDK

Vamos a programar nuestras aplicaciones Android en Java y por tanto necesitamos su kit de desarrollo: **JDK**.

No se debe confundir JDK (*Java Development Kit*), que es el kit de desarrollo de Java que nos permite crear aplicaciones Java, con JRE (*Java Runtime Environment*) o entorno de ejecución Java que nos permite la ejecución de aplicaciones creadas en Java pero no su desarrollo.

Si no lo tenemos instalado deberemos proceder a su descarga e instalación antes de instalar Android Studio. Podemos obtener la última versión en el siguiente enlace:

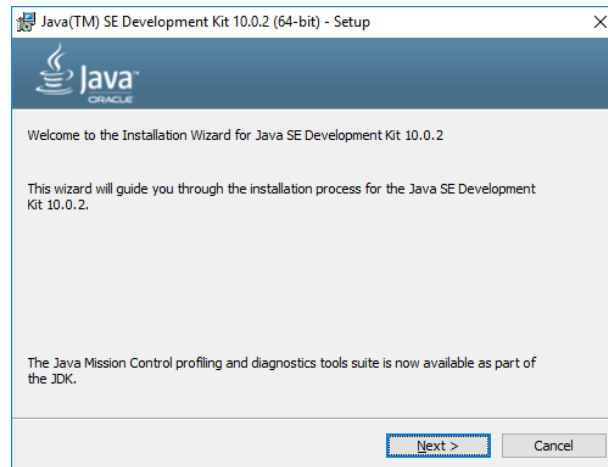
<https://www.oracle.com/java/technologies/javase-jdk15-downloads.html>



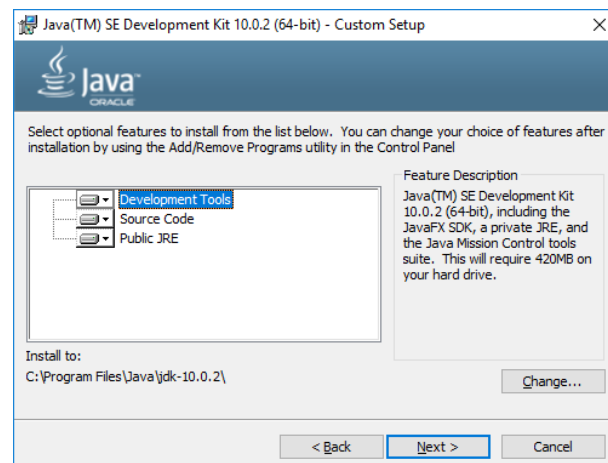
En la parte inferior de la página seleccionaremos la versión adecuada a nuestro sistema operativo. Por ejemplo, Windows 64 bits:

Java SE Development Kit 15		
This software is licensed under the Oracle Technology Network License Agreement for Oracle Java SE		
Product / File Description	File Size	Download
Linux ARM64 RPM Package	141.79 MB	jdk-15_linux-aarch64_bin.rpm
Linux ARM64 Compressed Archive	156.98 MB	jdk-15_linux-aarch64_bin.tar.gz
Linux Debian Package	154.77 MB	jdk-15_linux-x64_bin.deb
Linux RPM Package	161.99 MB	jdk-15_linux-x64_bin.rpm
Linux Compressed Archive	179.31 MB	jdk-15_linux-x64_bin.tar.gz
macOS Installer	175.46 MB	jdk-15_osx-x64_bin.dmg
macOS Compressed Archive	176.07 MB	jdk-15_osx-x64_bin.tar.gz
Windows x64 Installer	159.68 MB	jdk-15_windows-x64_bin.exe

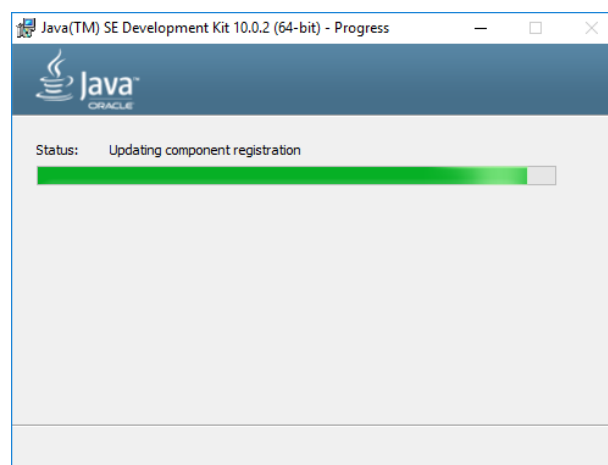
Una vez descargada la versión de la JDK adecuada para nuestra máquina procedemos a la instalación.



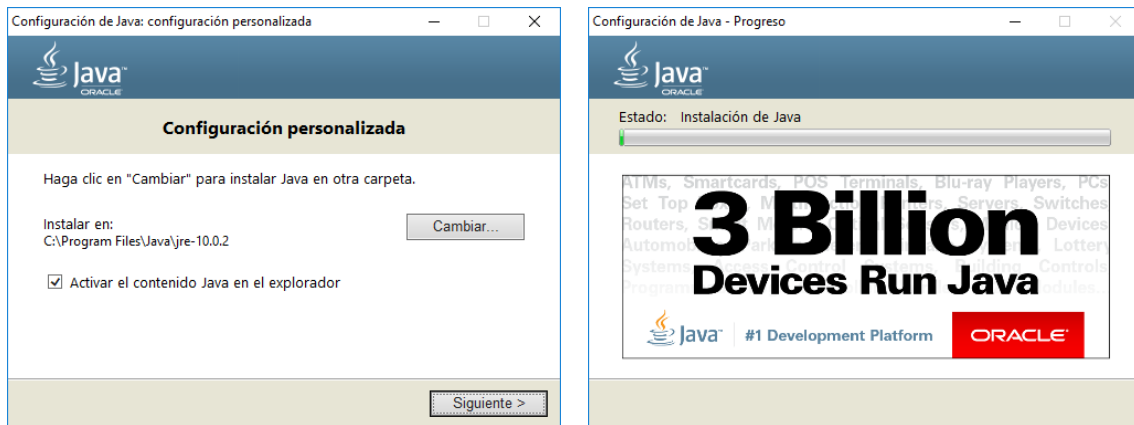
Establecemos los componentes que queremos instalar y la ruta donde queremos que se instale JDK (podemos dejar las opciones por defecto):



Si hemos tenido alguna versión anterior se procederá a eliminarla:



Y posteriormente comenzará la instalación:



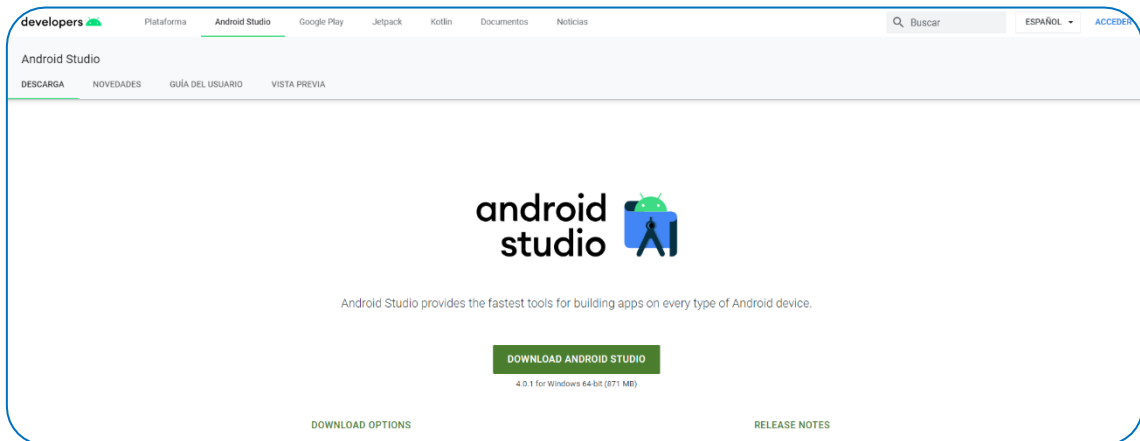
Una vez finalizada la instalación pulsamos sobre "Cerrar" y ya estaremos listos para instalar Android Studio.



INSTALACIÓN DE ANDROID STUDIO

Podemos obtener Android Studio desde la página oficial, aunque dado el tamaño de la descarga se facilitará en clase.

<https://developer.android.com/studio/index.html>

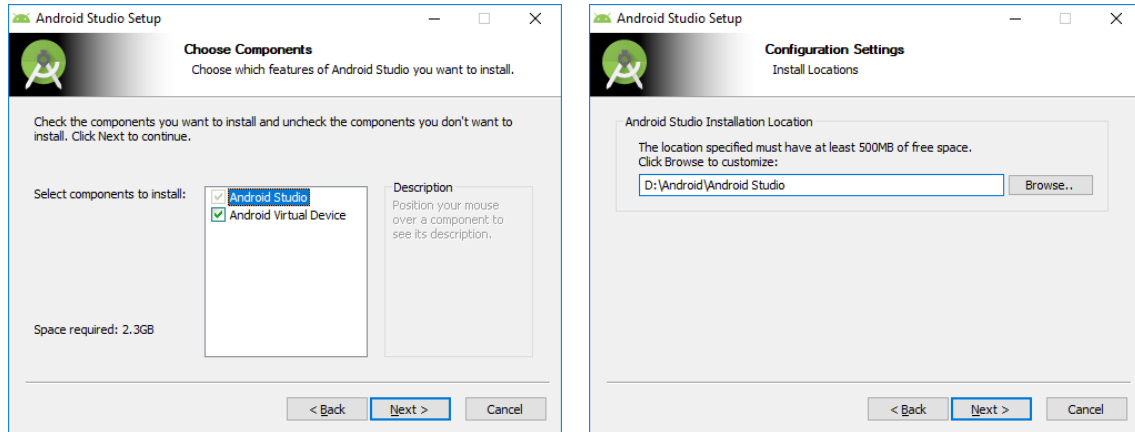


La instalación de Android Studio no tiene ninguna complicación, pero dado que ocupa gran cantidad de espacio en disco, vamos a instalarlo en la unidad D.

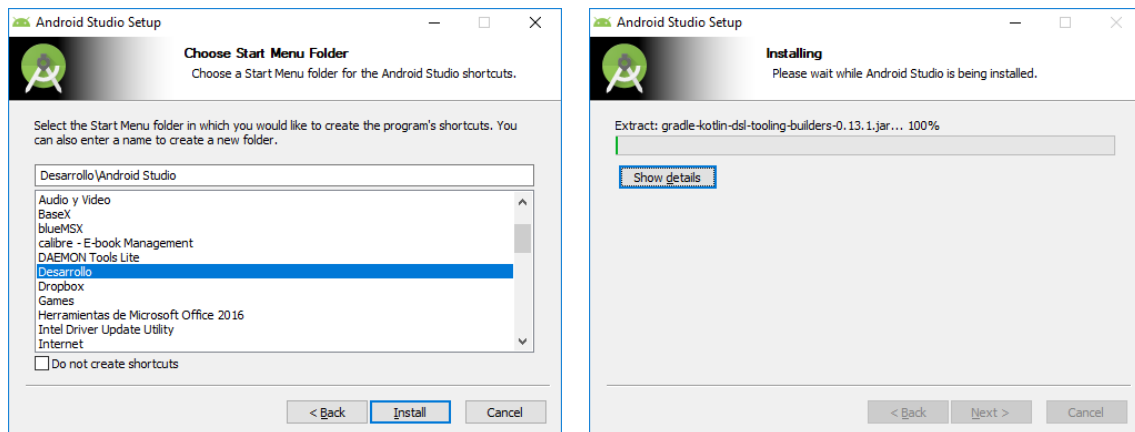


En primer lugar, seleccionamos los componentes (tal y como se muestra en la imagen hemos de marcar todos), y establecemos la ruta de instalación:

“D:\Android\Android Studio”



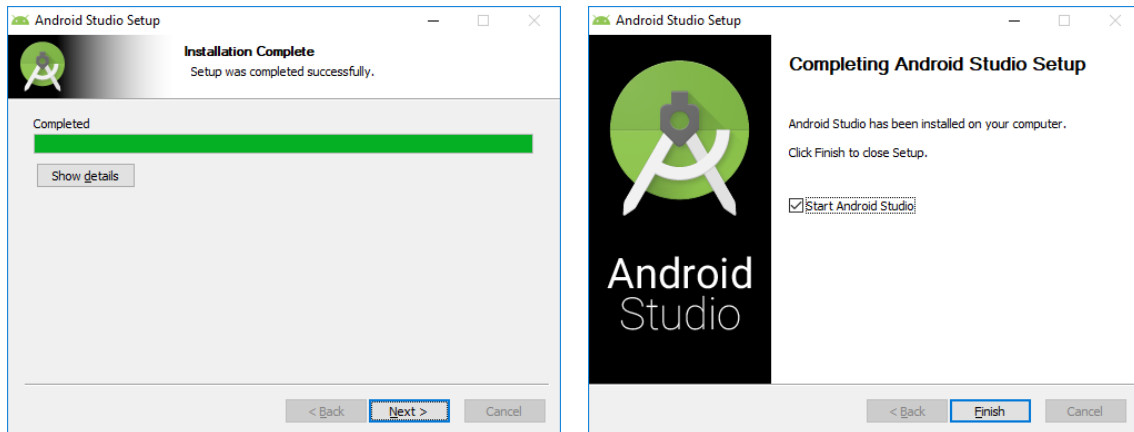
A continuación, establecemos la ruta donde queremos que se muestre la aplicación dentro del menú de inicio y comenzamos la instalación.



Mientras se realiza la instalación podemos aprovechar para leer acerca de las funciones de Android Studio en la página oficial, e informarnos por ejemplo sobre qué es “Instant Run”.

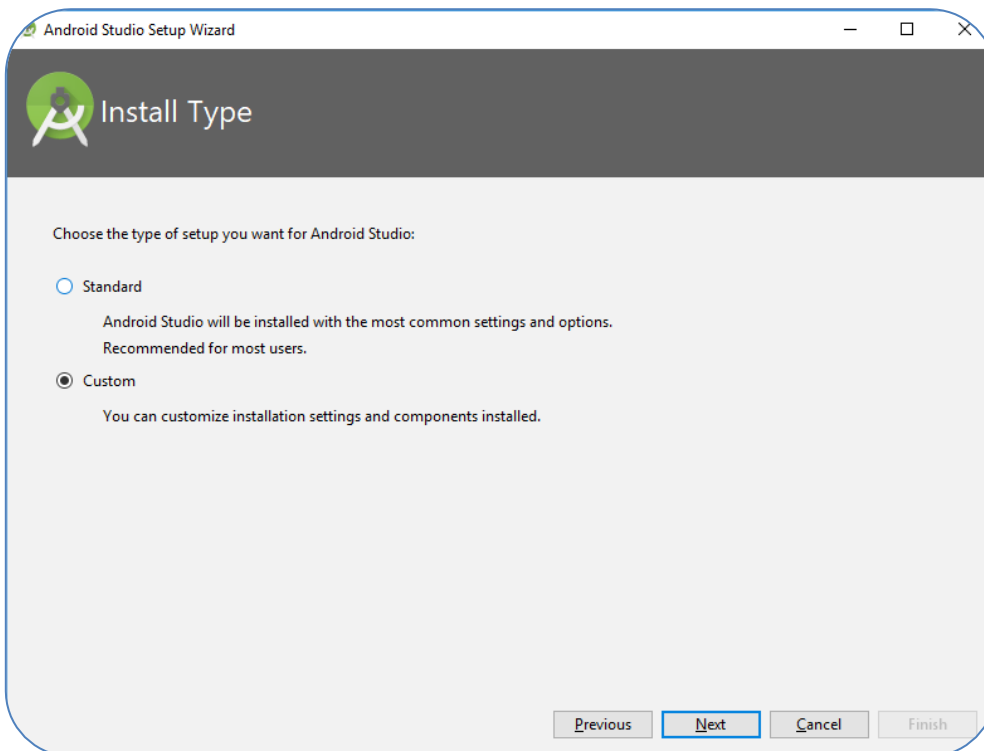
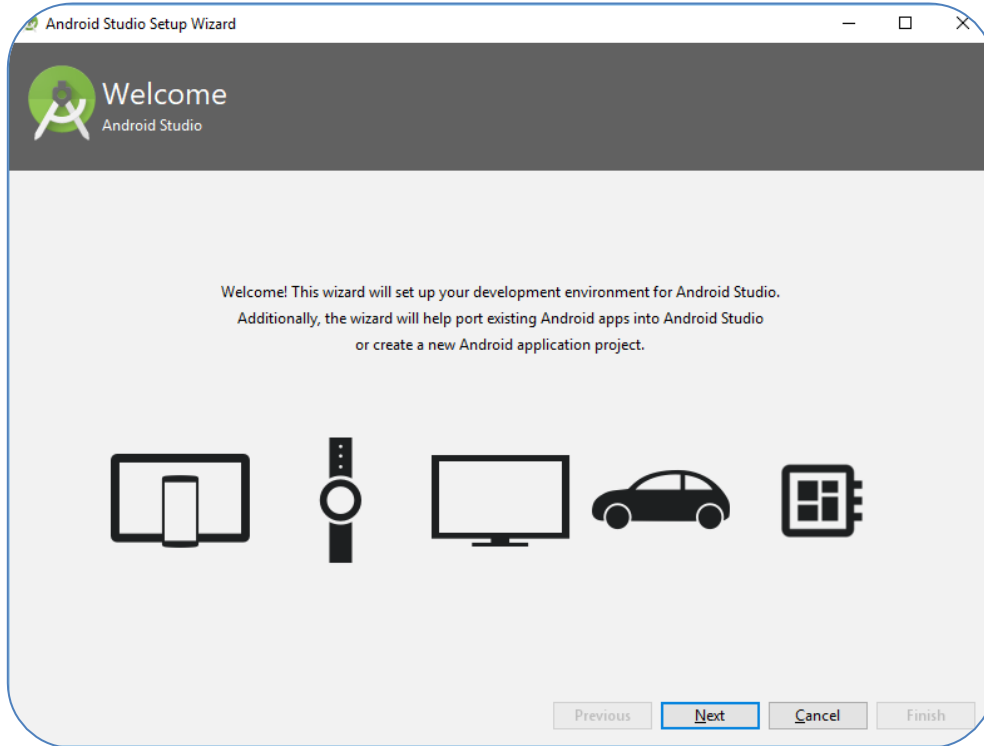
<https://developer.android.com/studio/intro/?hl=es-419>

En unos minutos finalizará la instalación de Android Studio.

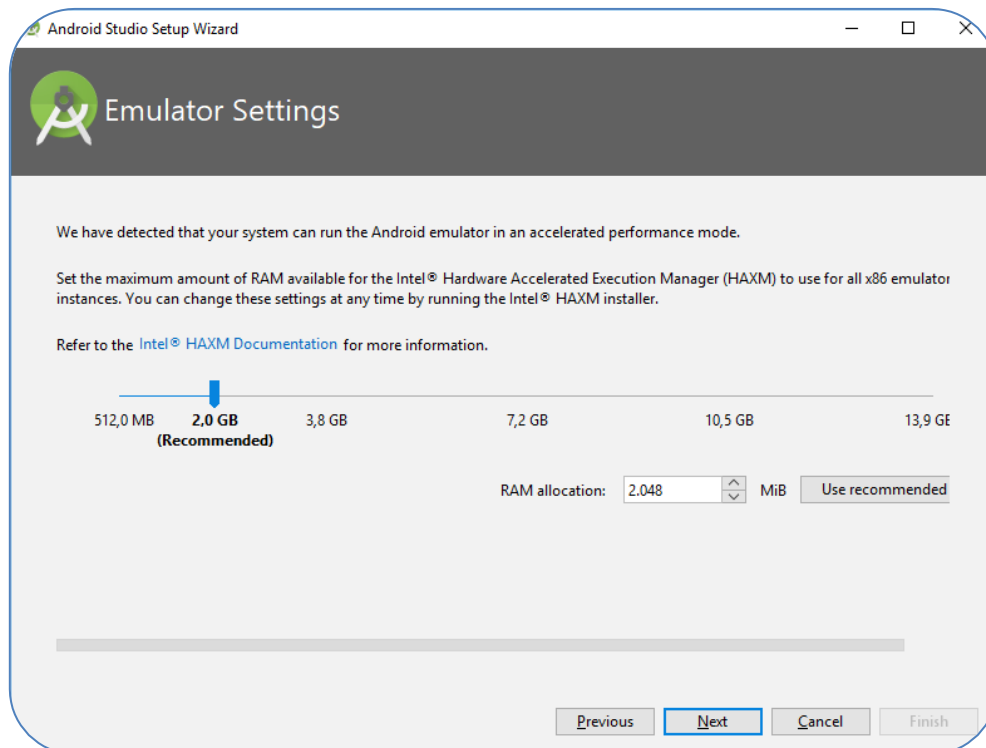


CONFIGURACIÓN DE ANDROID STUDIO

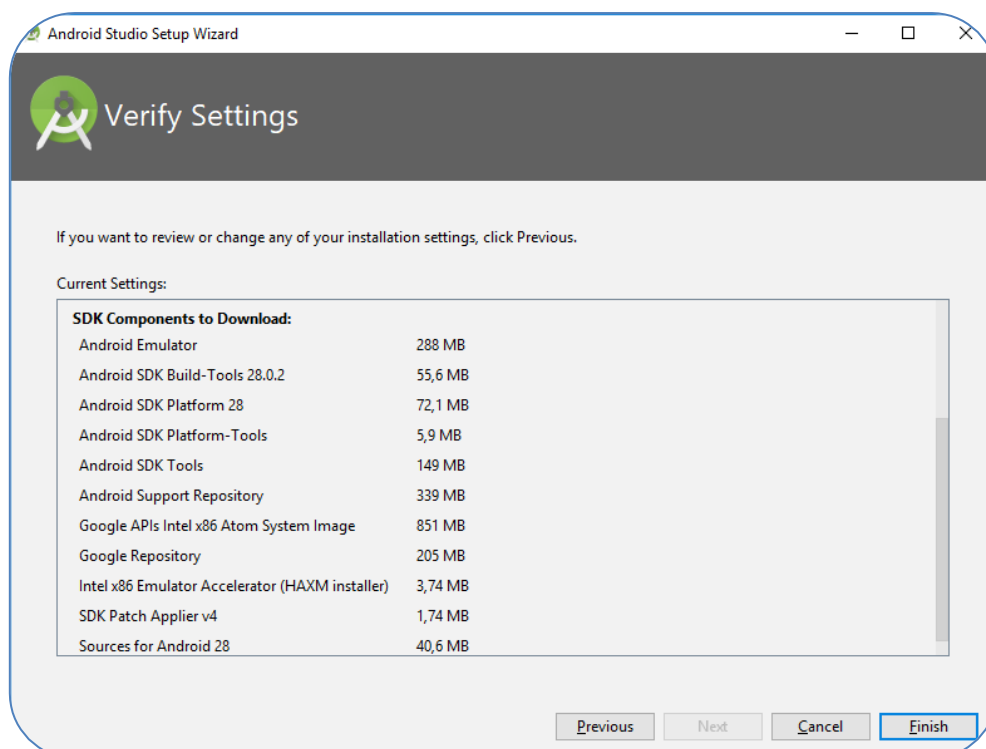
La primera vez que ejecutemos Android Studio se lanzará un asistente de configuración, en el que seleccionaremos “**Custom**” para poder personalizarla:



En el paso siguiente hemos de establecer la configuración del emulador, podemos dejar la configuración recomendada:



Por último, se muestra un resumen de la configuración y le damos a finalizar para que se inicie el proceso de descarga e instalación de los componentes necesarios.

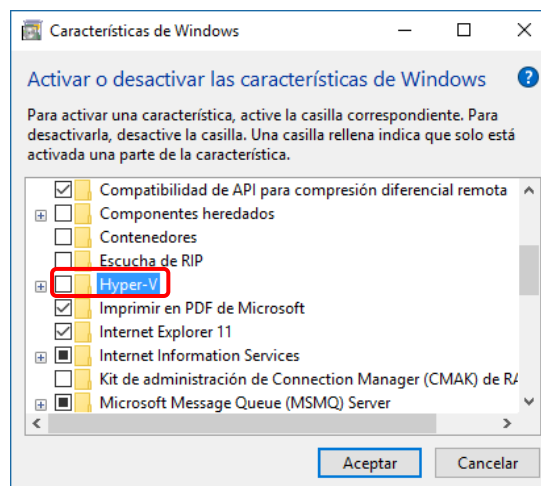


PROBLEMAS FRECUENTES

1) Unable to install Intel HAXM

Los terminales virtuales para probar nuestras aplicaciones utilizan HAXM (Hardware Accelerated Execution Manager). Si nos aparece este mensaje puede deberse a varias causas:

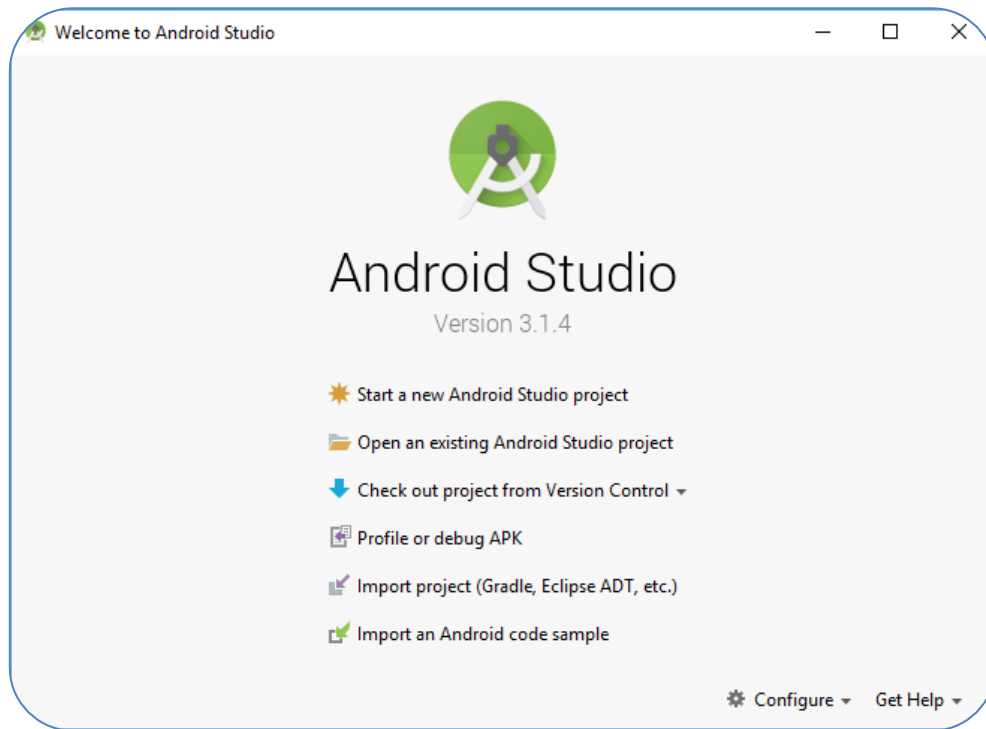
- a) La virtualización no está habilitada: la forma de habilitarlo dependerá de la BIOS que tengamos, pero en general habrá que entrar en modo avanzado y buscar entre las opciones de la CPU para habilitar "Intel Virtualization Technology".
- b) Windows tiene activada la virtualización Hyper-V: habrá que desactivarla, para ello hemos de abrir el "Panel de Control", "Programas", "Activar o desactivar las características de Windows". En la ventana que nos aparece hemos de buscar "Hyper-V" y desactivarla.



- c) Nuestro equipo no soporta "VT-x": para probar nuestras aplicaciones podremos optar por utilizar un dispositivo físico (móvil) conectado a nuestro ordenador por USB, ó utilizar un dispositivo virtual basado en ARM, aunque éste será 10 veces más lento que HAXM.

PRIMER CONTACTO CON ANDROID STUDIO

Una vez finalizada la instalación ya podemos abrir Android Studio para empezar a familiarizarnos con el entorno de trabajo.



Desde la pantalla de bienvenida podemos acceder a las siguientes opciones:

- Crear un nuevo proyecto.
- Abrir un proyecto existente.
- Recuperar un proyecto desde control de versiones (Git, SVN, etc).
- Importar un proyecto desde otro IDE (Eclipse ADT, Gradle, etc).
- Importar código de ejemplo Android.

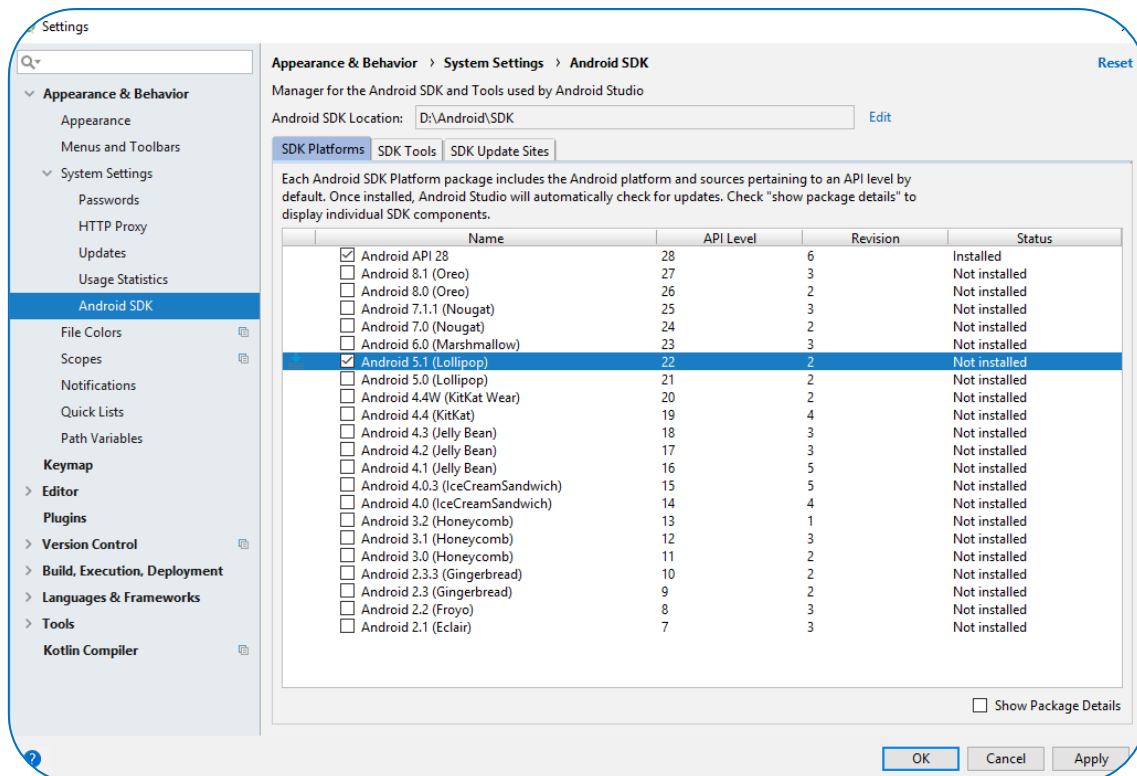
En la parte inferior de la pantalla podemos acceder a tres opciones adicionales:

- Visualizar los eventos relativos a Android Studio.
- Configurar Android Studio.
- Obtener ayuda.

CONFIGURACIÓN DE ANDROID STUDIO: SDK MANAGER

El **SDK** (*Software Development Kit*) de Android es el conjunto de herramientas necesarias para el desarrollo de aplicaciones Android.

Para configurar el SDK disponemos de la herramienta **SDK Manager**, a la que podemos acceder tanto desde la opción de configuración de la pantalla de inicio, como desde el menú: *Tools, SDK Manager*.

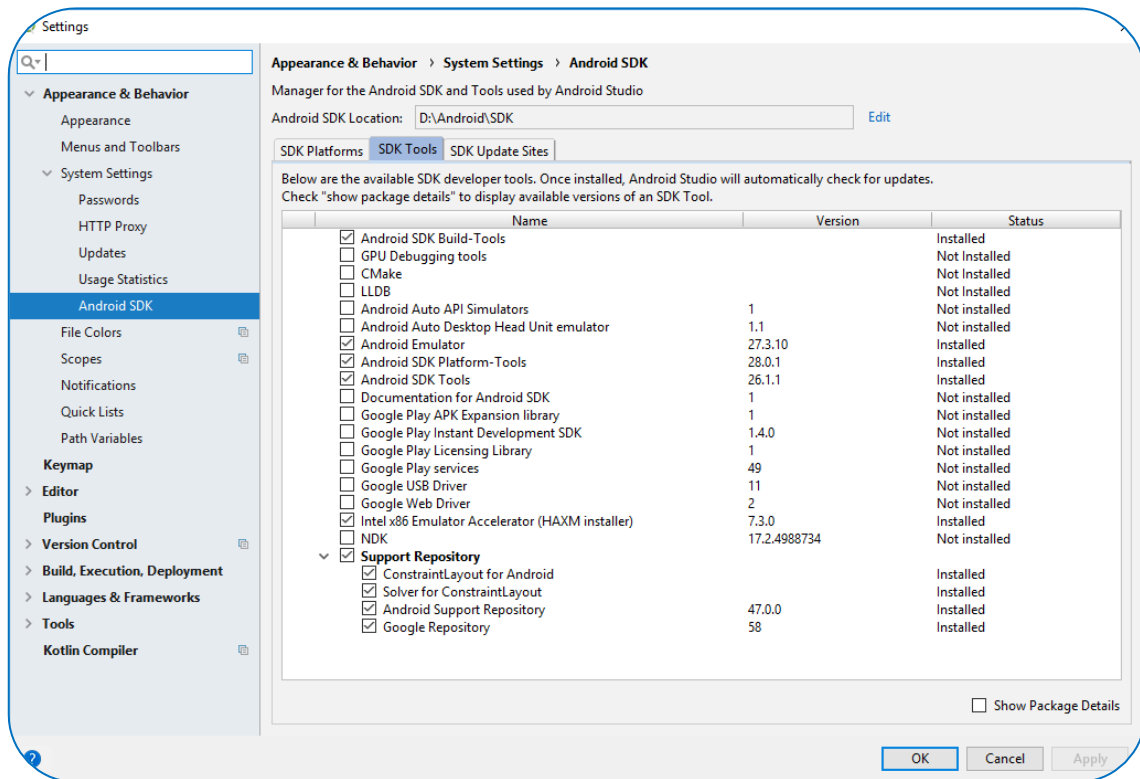


- **SDK Platforms:** en esta pestaña podemos establecer que versiones del sistema operativo Android vamos a utilizar para el desarrollo de nuestras aplicaciones.

Como desarrolladores nos interesa crear aplicaciones que lleguen al mayor número de usuarios y por ello debemos estar al tanto de cuáles son las versiones de Android que están más presentes en los terminales en cada momento (en el *AVD manager* podremos ver el porcentaje de móviles que podremos alcanzar con cada API). Por otro lado, también nos interesa familiarizarnos con las novedades que van incorporando las nuevas versiones por lo que también deberíamos tener instaladas las últimas versiones.

De momento nos quedaremos con la última versión y con Android 5.1 (*Lollipop*), y más adelante instalaremos alguna versión más.

- **SDK Tools:** en esta pestaña podemos establecer las herramientas de desarrollo que vamos a utilizar.

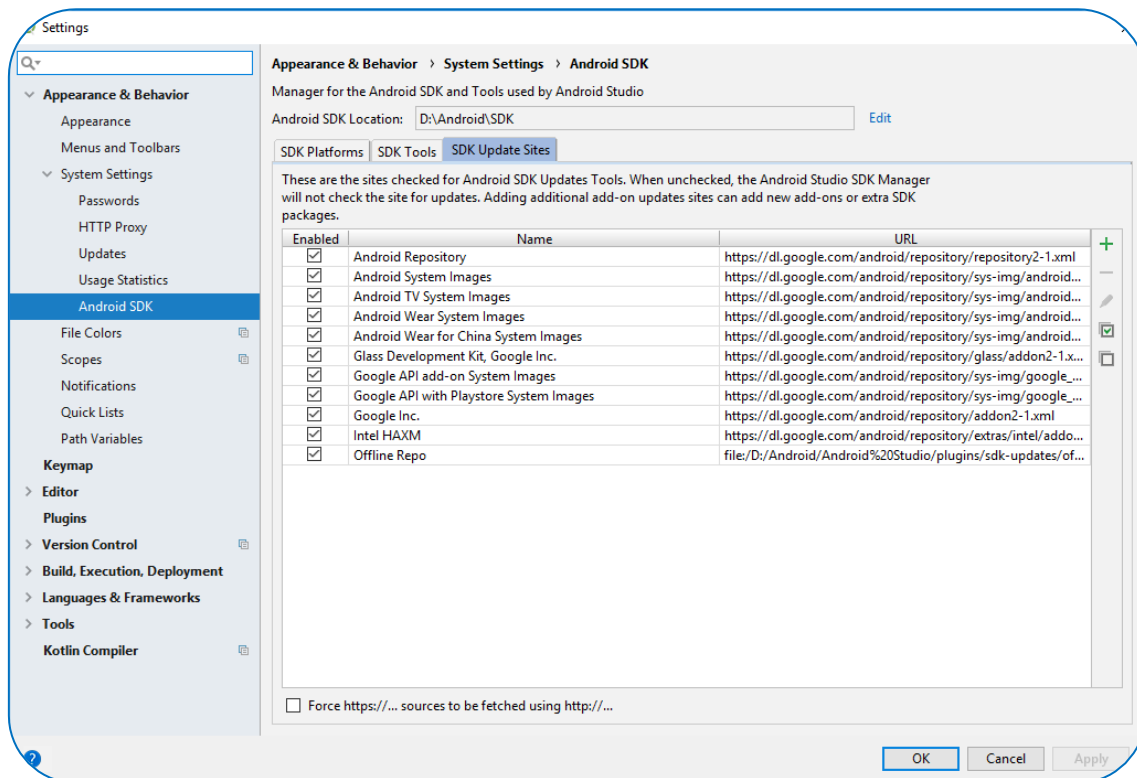


En principio deberemos tener instaladas las siguientes:

- **Android SDK Tools** (instalado por defecto): elementos necesarios para desarrollar, probar y depurar su aplicación.
- **Android SDK Platform-tools**: herramientas de desarrollo, pruebas y depuración dependientes de la plataforma (versión de Android).
- **Android SDK Build-Tools**: herramientas para construir y configurar las aplicaciones.
- **Android Emulator**: permite crear dispositivos virtuales para probar nuestras aplicaciones.
- **Intel x86 Emulator Accelerator**: permite incrementar el rendimiento del emulador siempre que nuestro procesador sea Intel o AMD y soporte virtualización VT-x.
- **Support Repository** (compatibilidad): permite utilizar funcionalidades no incluidas en todas las versiones.

- **SDK Update Sites:** permite configurar los repositorios, es decir los lugares desde los que se descargan las actualizaciones, imágenes de sistema, herramientas, complementos...

De momento no realizaremos ninguna modificación en este apartado.



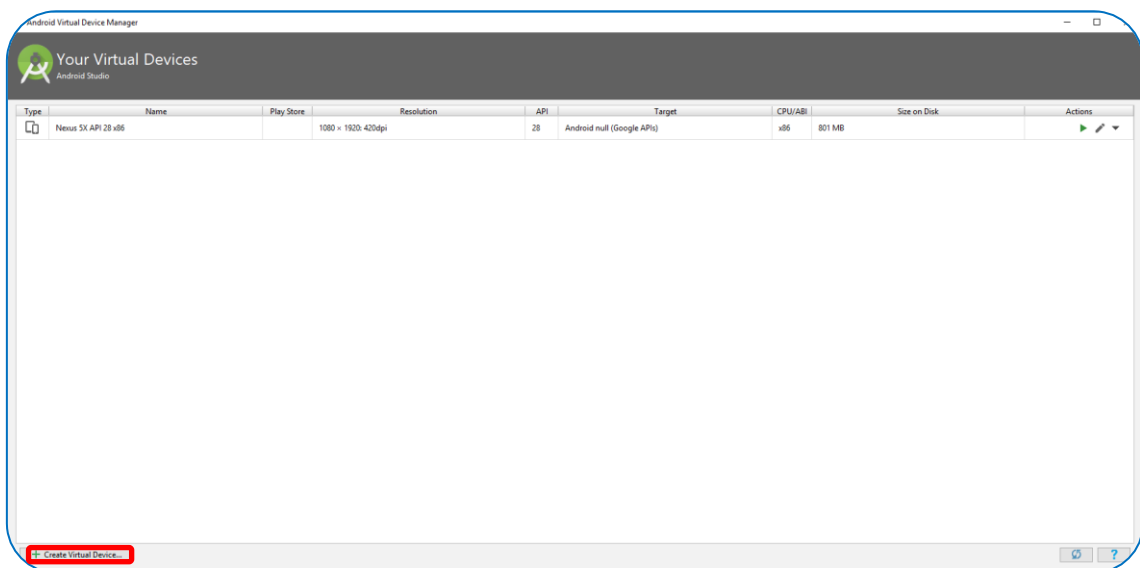
CONFIGURACIÓN DE ANDROID STUDIO: AVD MANAGER

El **AVD Manager** (*Android Virtual Device Manager*) nos va a permitir crear y gestionar emuladores Android para probar nuestras aplicaciones.

Necesitamos disponer de al menos un emulador para poder realizar las pruebas, pero podemos tener tantos como queramos con diferentes características en cuanto a tipo de dispositivo (teléfono, tableta, televisión, wearable, etc.), características hardware (tamaño de pantalla, procesador, memoria, etc.) o versión de Android (4.4, 5.1, 6.0, etc.), y de esta forma poder realizar pruebas de nuestra aplicación en diferentes entornos.

Para crear un nuevo dispositivo virtual (emulador) iremos al menú “*Tools*” de Android Studio y seleccionaremos “*AVD Manager*”.

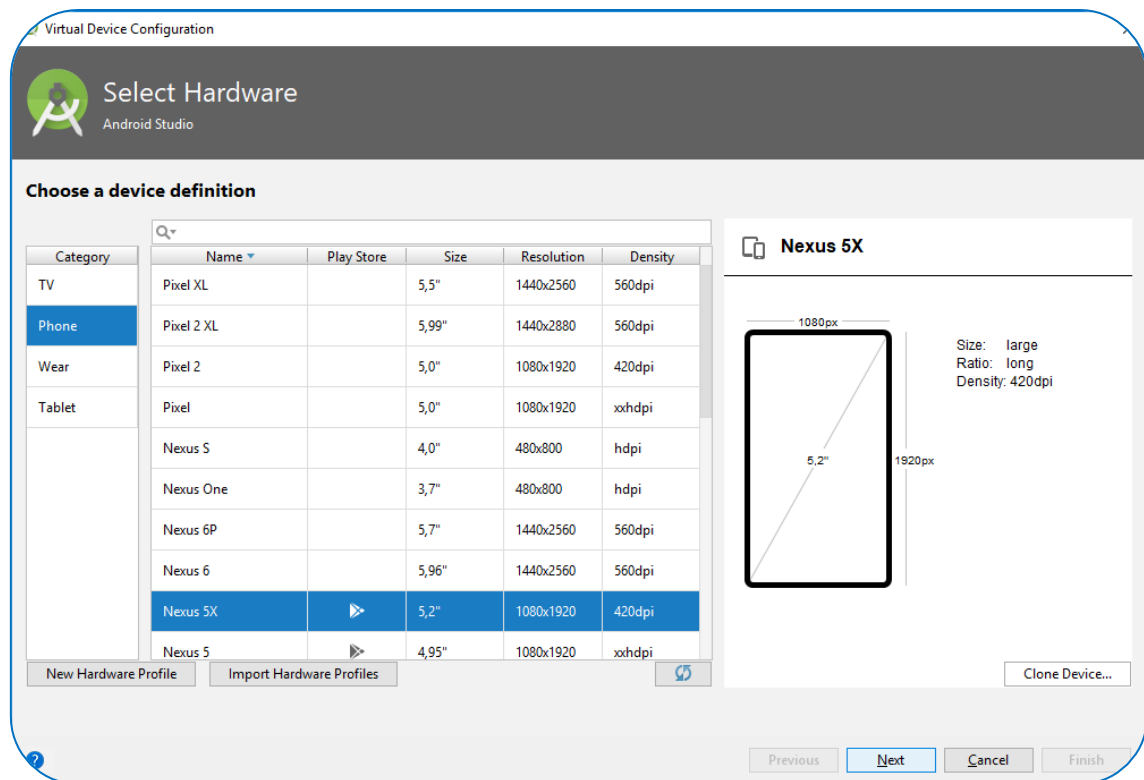
Se nos mostrará una pantalla con todas las máquinas virtuales que hayamos creado y podremos editar dichas máquinas o crear otra nueva mediante “*Create Virtual Device...*”



En la ventana que nos aparece, a la izquierda podemos seleccionar el tipo de dispositivo (*TV*, *Wear*, *Phone* ó *Tablet*), y una vez seleccionado uno, nos aparecen en la parte central los modelos disponibles, y a la derecha una vista esquemática del dispositivo.

También tenemos la opción de crear nuestro propio perfil hardware estableciendo “a mano” sus características o importar un perfil.

Una vez que tengamos seleccionado el dispositivo que deseemos pulsaremos sobre el botón “*Next*”.

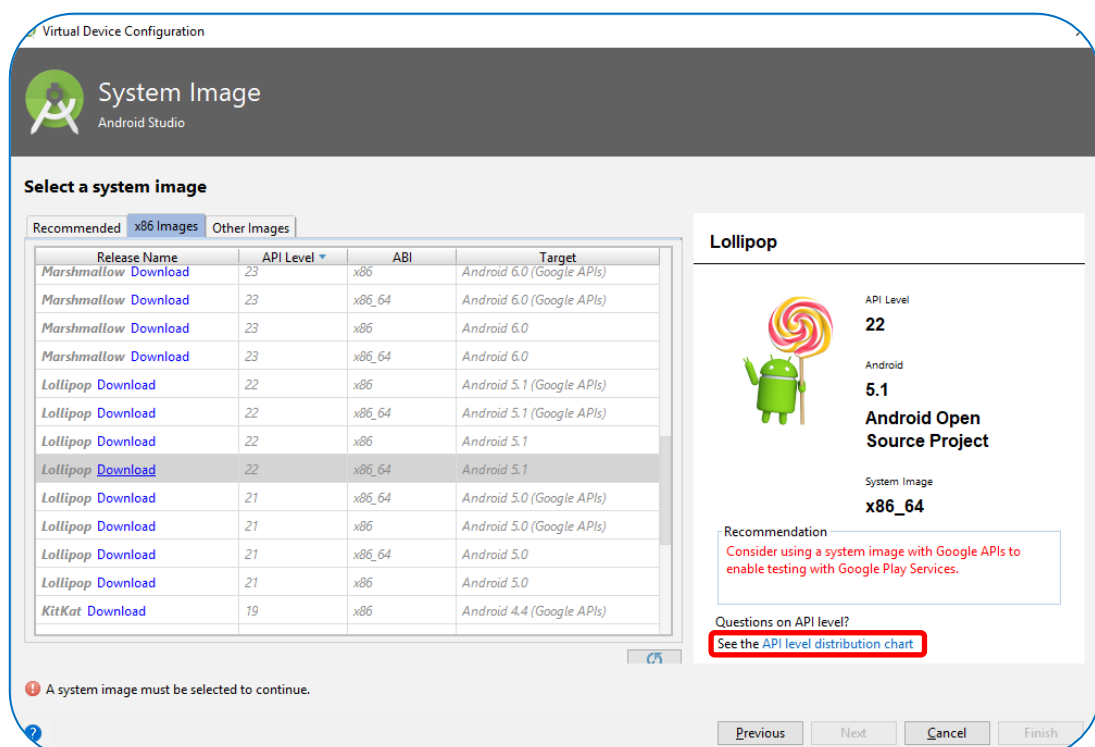


En el paso siguiente seleccionaremos la versión de Android (*system image*) que queremos utilizar en nuestro dispositivo virtual.

Las imágenes de sistema aparecen clasificadas en 3 pestañas:

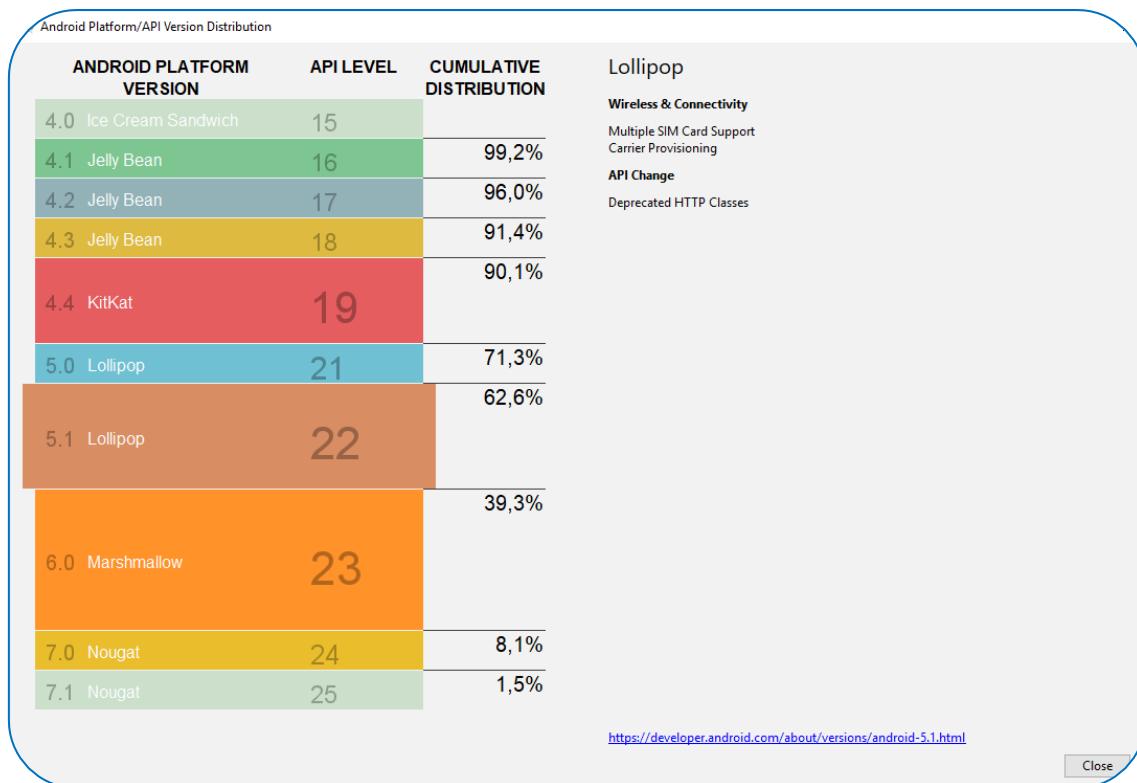
- *Recommended*: son las versiones recomendadas por Google y siempre van a ser las últimas versiones de Android.
- *x86 Images*: al utilizar el mismo conjunto de instrucciones que nuestro procesador (Intel o AMD) permiten obtener un rendimiento mucho mayor, sobre todo si nuestro procesador soporta VT-x y podemos utilizar HAXM (*Hardware Accelerated Execution Manager*).
- *Other Images*: incluye las imágenes ARM que son las que realmente utilizan los dispositivos móviles, pero que al requerir una traducción desde nuestro procesador van a hacer que la emulación sea mucho más lenta.

Sólo se recomienda utilizar este tipo de imágenes si nuestro procesador no soporta emulación VT-x haciendo imposible la utilización de HAXM.



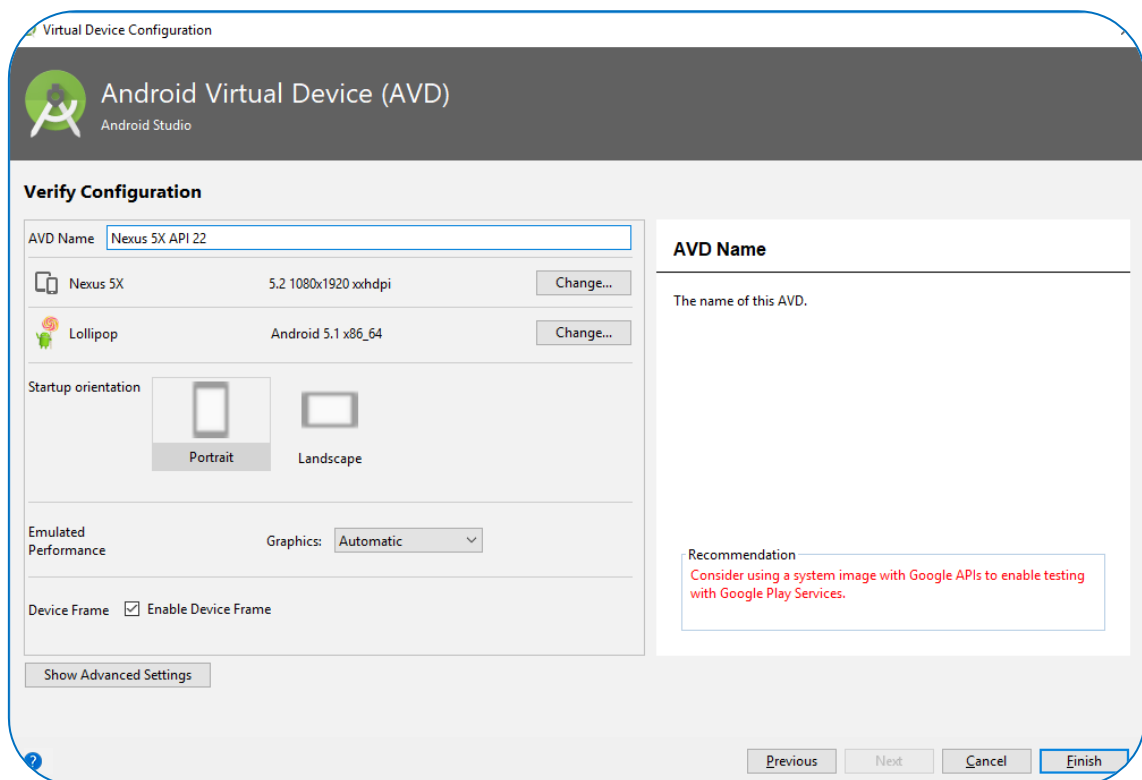
Si pulsamos sobre “*See the API distribution chart*” se nos muestra un gráfico con el porcentaje de terminales a los que podremos llegar con cada nivel de API (versión de Android) y las principales características de la versión seleccionada, para ayudarnos a tomar la decisión sobre para qué versión desarrollar nuestras aplicaciones.

Por ejemplo, con la versión de Android 5.1 (API 22) podríamos llegar hasta el 62,6% de los terminales, ya que cualquier versión superior soporta este nivel de API. Mientras que con la versión de Android 7.1 (API 25) sólo podríamos llegar al 1,5% de los terminales.

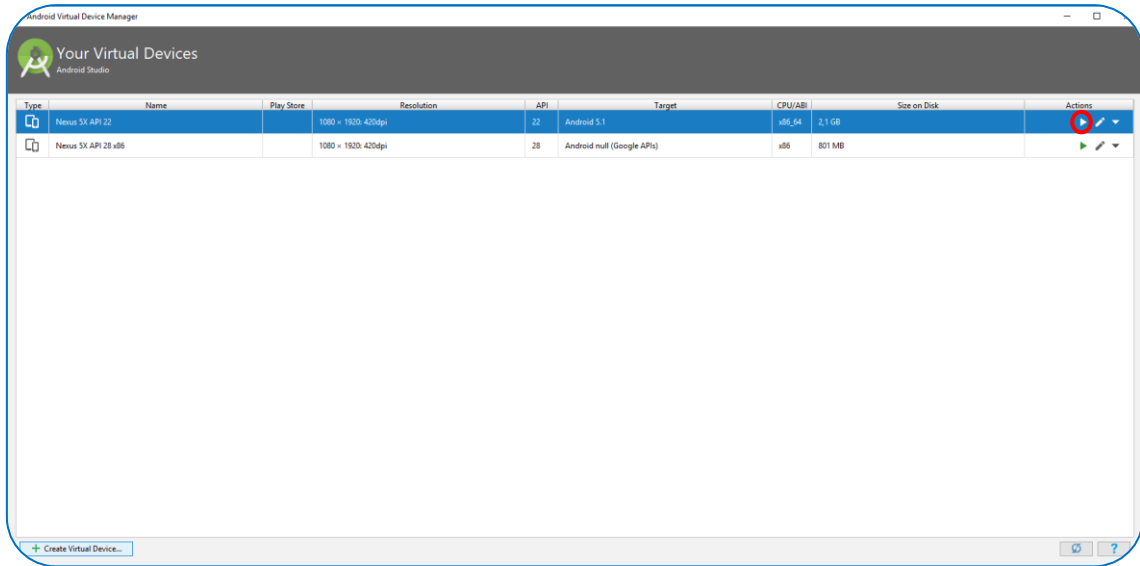


Por último, nos aparece un resumen con la configuración que hemos seleccionado, en la que podemos establecer un nombre para el dispositivo. Resulta conveniente poner un nombre que nos ofrezca toda la información en cuanto al dispositivo, versión de Android y nivel de API, para diferenciarlo de otros dispositivos virtuales.

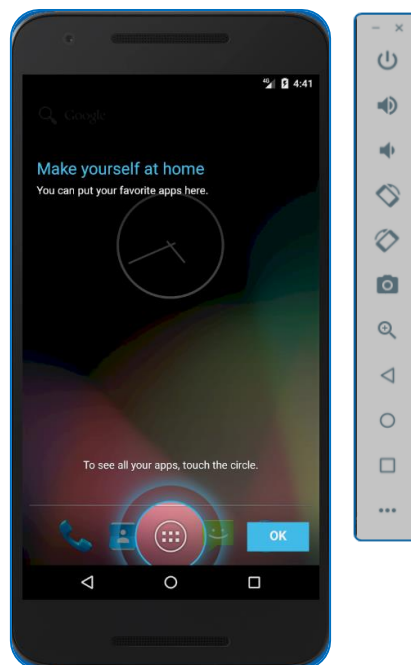
También podemos establecer la orientación de la pantalla (vertical u horizontal), el rendimiento de la emulación gráfica (automático) y en las opciones avanzadas podríamos establecer de forma más detalladas las características del hardware (procesador, memoria, almacenamiento, cámara, red, etc.).



Una vez que tengamos toda la configuración establecida pulsaremos el botón “*Finish*” y ya tendremos nuestro dispositivo disponible para probar las aplicaciones. Para ponerlo en marcha sólo tendremos que pulsar sobre el botón de ejecución.



Esta será la apariencia de nuestro dispositivo virtual:

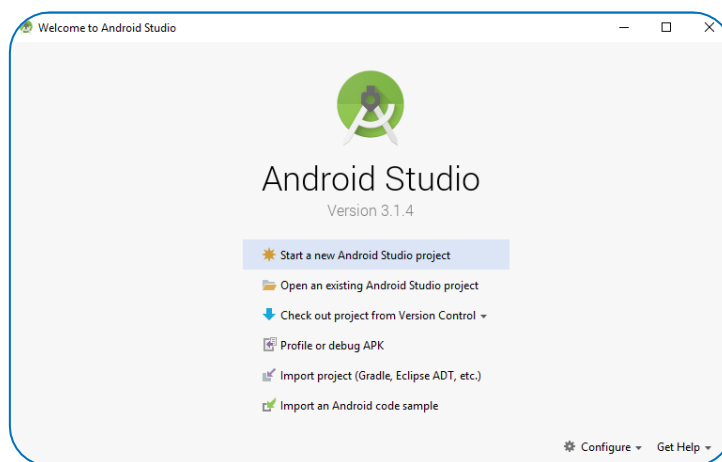


PRIMERA APLICACIÓN ANDROID: “HOLA ANDROID”

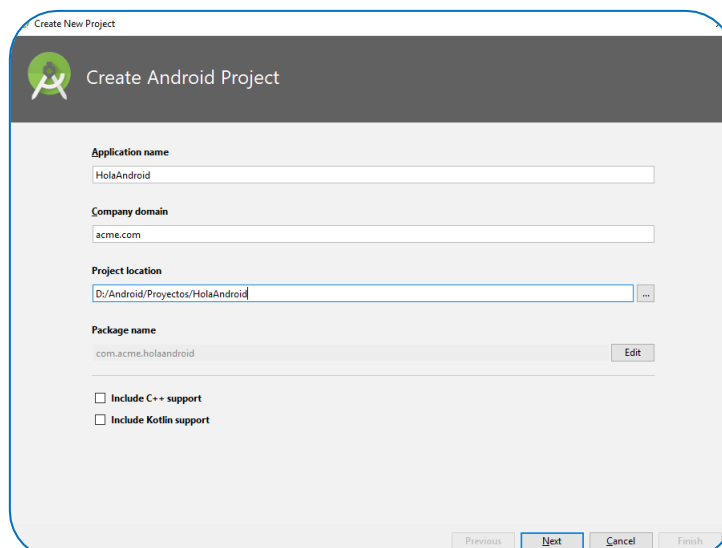
Nuestra primera aplicación Android va a ser una adaptación del famoso “*Hello World*” que sirvió para presentar el lenguaje C allá por 1974: “Hola Android”.

Se trata de una aplicación bastante modesta pues se va a limitar a mostrar en pantalla el mensaje: “Hola Android”, pero será suficiente para comprobar que nuestro entorno de trabajo está preparado y nuestro terminal virtual funciona correctamente.

Para crear una nueva aplicación pulsaremos sobre “*Start a new Android Studio project*” en la pantalla de bienvenida de Android Studio.

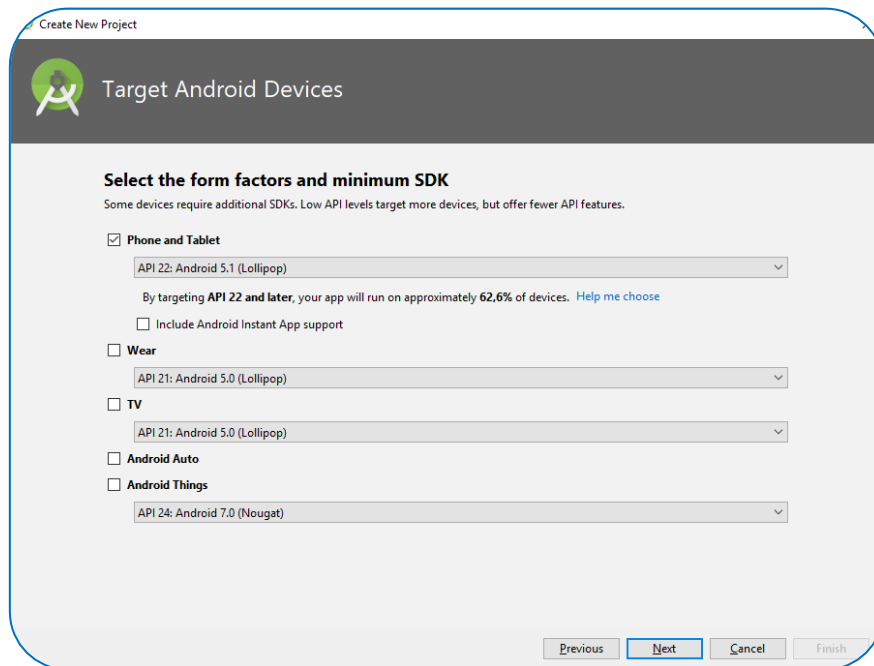


En la siguiente pantalla estableceremos el nombre de la aplicación, el nombre de la compañía y la ruta donde queremos que se cree el proyecto. Si quisiéramos implementar todo o alguna parte de nuestro proyecto en C++ deberíamos marcar la opción “*Include C++ support*”, pero no es el caso.

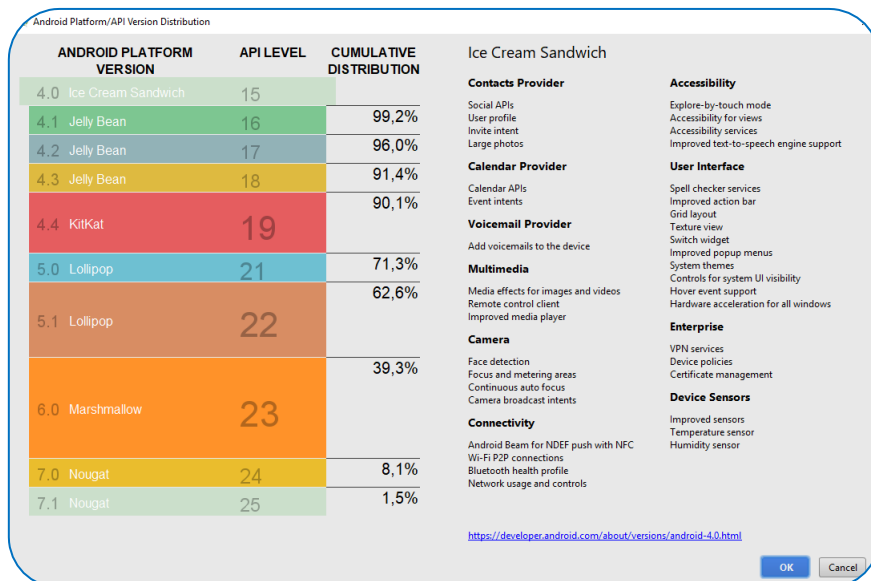


A continuación, seleccionaremos el dispositivo y la versión mínima del sistema Android requerida para que nuestra aplicación funcione.

Hay que tener en cuenta que tal y como nos advierte un mensaje, cuanto menor sea el nivel de API mayor será el número de dispositivos sobre los que se puede ejecutar nuestra aplicación, pero menor será también el número da características disponibles.

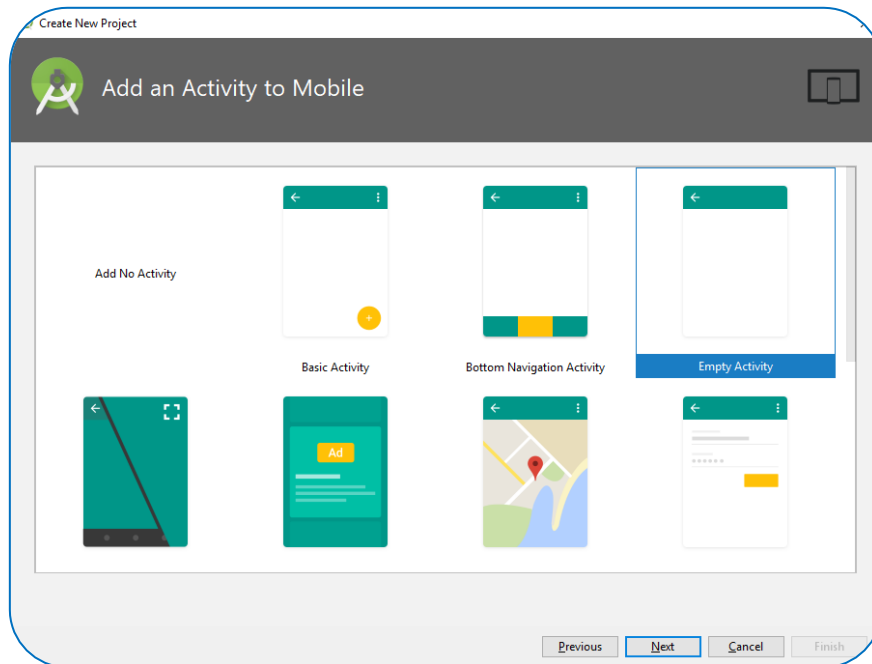


Si pulsamos sobre el texto “*Help me choose*” se nos mostrará el porcentaje de móviles que podrán ejecutar nuestra aplicación en función del nivel de API seleccionado y las características que podremos utilizar en cada uno de ellos.

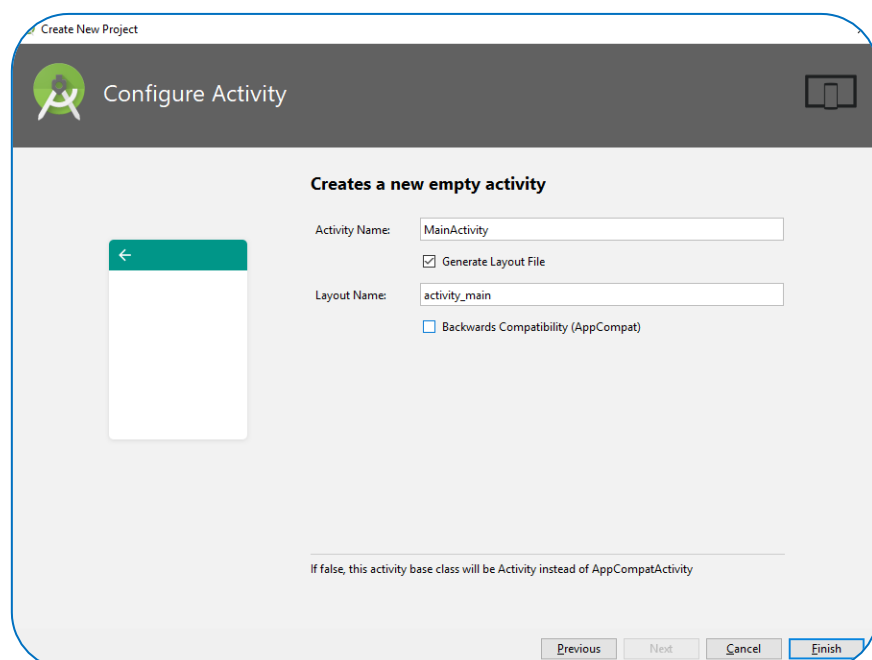


Según esta información con el nivel de API 22 (Android 5.1) podremos alcanzar el 62,6% de los dispositivos móviles Android activos.

En la siguiente pantalla seleccionaremos el tipo de Actividad, variando las opciones disponibles en función del nivel de API seleccionado en el paso anterior. Ya veremos más adelante que son las Actividades y sus diferentes tipos, de momento nos limitaremos a seleccionar una actividad vacía “*Empty Activity*”.

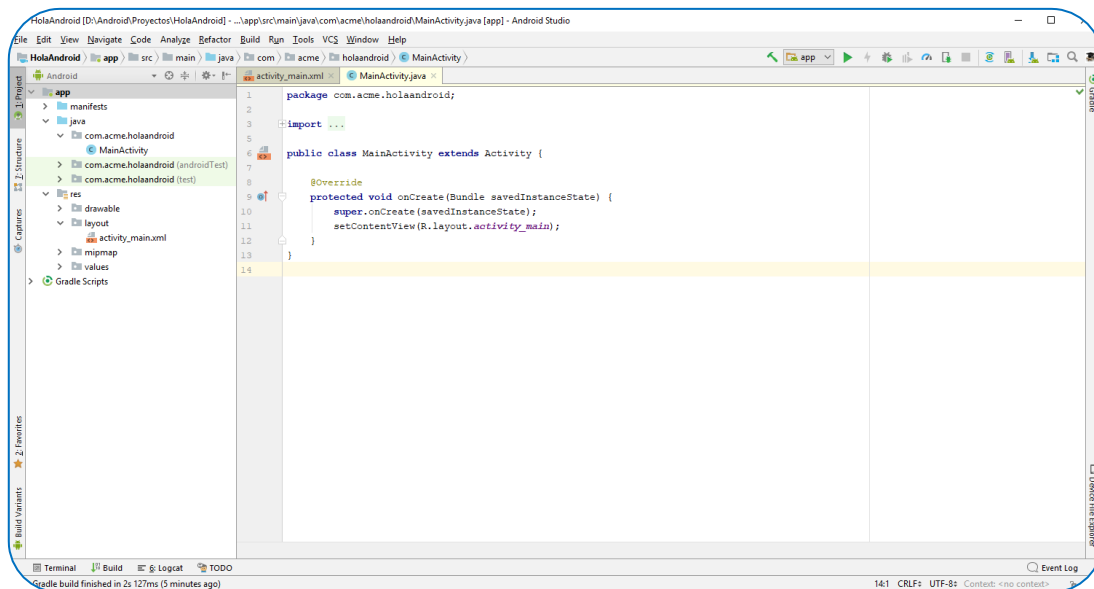


Por último, podemos personalizar el nombre de la actividad (*activity*) y de la pantalla de diseño (*layout*) y desactivar la compatibilidad hacia atrás (AppCompat).

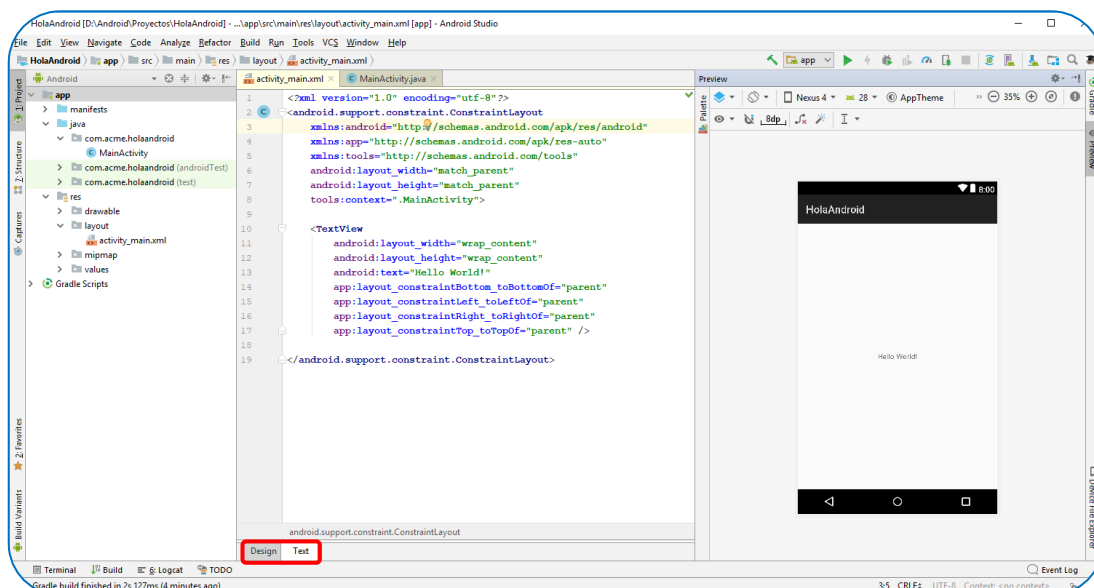


Tras unos instantes tendremos creados nuestro proyecto y se nos mostrarán en pantalla dos archivos:

- **MainActivity.java**: es la actividad principal de nuestra aplicación, tiene un único método “*onCreate*” que se limita a mostrar el diseño de pantalla establecido en el archivo de recursos *activity_main.xml*.

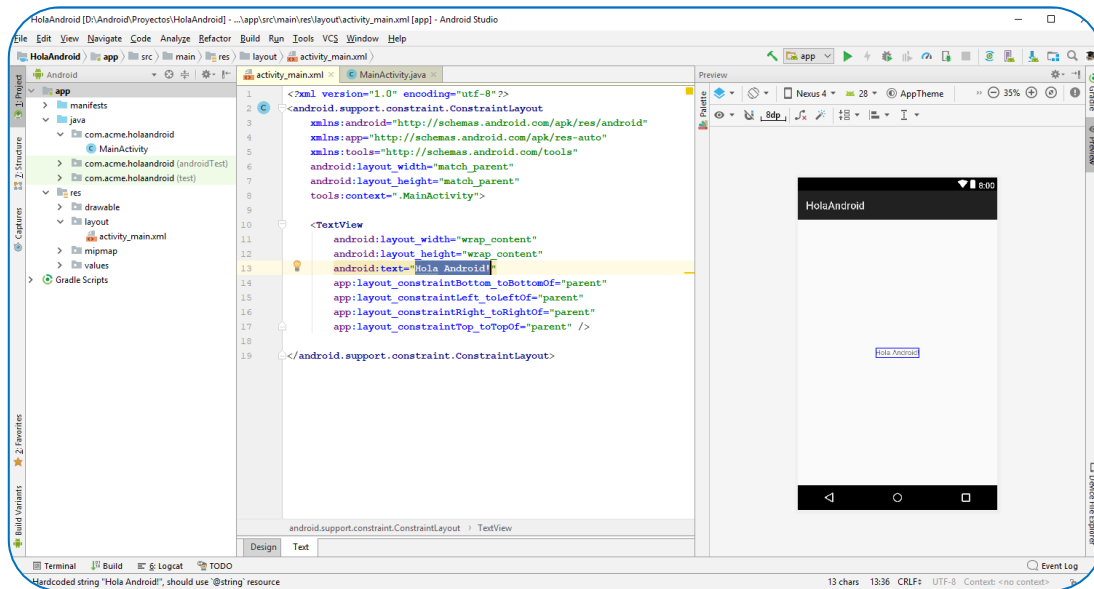


- **activity_main.xml**: es un archivo de recursos, en concreto de diseño de pantalla (*layout*) que establece los elementos que se van a mostrar en pantalla y su disposición. Cuando abramos un archivo de este tipo podremos elegir en la parte inferior entre vista de diseño (paleta de controles) o vista de texto (contenido del archivo).

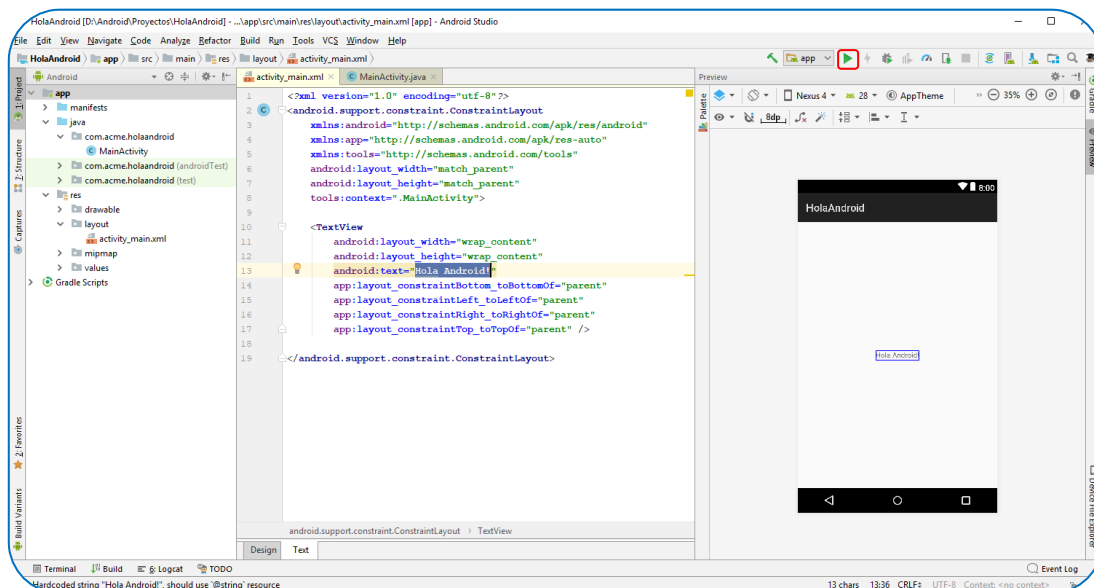


Nos vamos a la vista “Text”, buscamos dentro del elemento “TextView” la propiedad “Android:text” y modificamos su valor por “Hola Android!”.

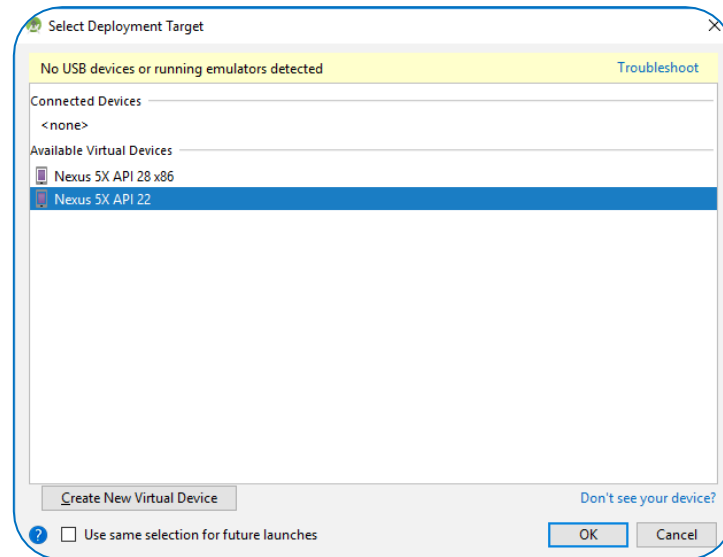
En la parte derecha tenemos una vista previa para ver cómo van a quedar nuestros cambios.



Ya hemos terminado nuestra primera aplicación Android y es el momento de probarla, para ello pulsaremos sobre el botón “Run app” de la barra de herramientas superior.



Se nos mostrará una pantalla donde podremos seleccionar el dispositivo (físico o virtual) sobre el que queremos ejecutar la aplicación. Si no tuviéramos ningún dispositivo disponible podríamos crearlo desde aquí.



Una vez seleccionado el dispositivo virtual se lanzará el emulador y tendrá que iniciar el sistema Android antes de instalar y ejecutar la aplicación.

Este proceso puede ser lento por lo que se recomienda que una vez lanzado el emulador no se cierre para no tener que repetir todo el proceso en futuras ejecuciones.

Además, como se ha comentado, Android Studio incorpora un sistema denominado “*Instant Run*” que permite visualizar los cambios realizados en el código sin tener que recompilar toda la aplicación lo que acelera mucho las siguientes ejecuciones.

Este sería el aspecto de nuestra aplicación...

