



El futuro digital
es de todos

MinTIC

Ciclo 4

Fundamentos de programación

Aplicaciones móviles Android con Java

Sesión 1: IDE android studio

Programa Ciencias de la Computación e Inteligencia Artificial
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda
Bogotá



UNIVERSIDAD
SERGIO ARBOLEDA

Mision
TIC 2022



Agenda

1. ¿Que es Android?
2. Características de la plataforma
3. Arquitectura de Android
4. Descarga e instalación
5. Creación de un proyecto Android <<MiPrimerPrograma>>





1. ¿Qué es Android?

Android es un sistema operativo móvil basado en núcleo Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles con otros sistemas a través de Android Auto, al igual los automóviles con el sistema Android Automotive y televisores Leanback [1].

Inicialmente fue desarrollado por Android Inc., que adquirió Google en 2005.³ Android fue presentado en 2007 junto con la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.⁴ El código fuente principal de Android se conoce como Android Open Source Project (AOSP), que se licencia principalmente bajo la Licencia Apache.⁵ Android es el sistema operativo móvil más utilizado del mundo, con una cuota de mercado superior al 90 % al año 2018, muy por encima de IOS [1].

[1] <https://es.wikipedia.org/wiki/Android>





2. Características de la plataforma

Existen muchas plataformas para móviles (Apple iOS, Windows Phone, BlackBerry, Palm, Java Micro Edition, Linux Mobile (LiMo), Firefox OS, etc.); sin embargo, Android presenta una serie de características que lo hacen diferente. Es el primero que combina en una misma solución las siguientes cualidades [2]:

- **Plataforma abierta:** Es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y customizar el sistema sin pagar royalties (regalía) [2].
- **Adaptable a diversos tipos de hardware:** Android no ha sido diseñado exclusivamente para su uso en teléfonos y tabletas. Hoy en día podemos encontrar relojes, gafas, cámaras, TV, sistema para automóviles, electrodomésticos y una gran variedad de sistemas integrados que se basan en este sistema operativo, lo cual tiene sus evidentes ventajas, pero también un esfuerzo adicional para el programador. La aplicación ha de funcionar correctamente en dispositivos con una gran variedad de tipos de entrada, pantalla, memoria, etc. En iOS de Apple tenemos que desarrollar una aplicación para iPhone y otra diferente para iPad [2].
- **Portabilidad asegurada.** Las aplicaciones finales son desarrolladas en Java, lo que nos asegura que podrán ser ejecutadas en cualquier tipo de CPU. Esto se consigue gracias al concepto de máquina virtual [2].
- **Arquitectura basada en componentes inspirados en Internet.** Por ejemplo, el diseño de la interfaz de usuario se hace en XML, lo que permite que una misma aplicación se ejecute en un reloj de pantalla reducida o en un televisor [2].

[2]. <http://www.androidcurso.com/index.php/recursos/31-unidad-1-vision-general-y-entorno-de-desarrollo/97-que-hace-a-android-especial>





2. Características de la plataforma

- **Filosofía de dispositivo siempre conectado a Internet.** Muchas aplicaciones solo funcionan si disponemos de una conexión permanente a Internet. Por ejemplo, comunicaciones interpersonales o navegación con mapas [2].
- **Gran cantidad de servicios incorporados.** Por ejemplo, localización basada tanto en GPS como en redes, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia, etc [2].
- **Aceptable nivel de seguridad.** Los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja, que hereda de Linux. Además, cada aplicación dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.). Desde la versión 6.0 el usuario puede conceder o retirar permisos a las aplicaciones en cualquier momento [2].
- **Optimizado para baja potencia y poca memoria.** En el diseño de Android se ha tenido en cuenta el hardware específico de los dispositivos móviles. Por ejemplo, Android utiliza la máquina virtual ART (o Dalvik en versiones antiguas). Se trata de una implementación de Google de la máquina virtual Java optimizada para dispositivos móviles [2].
- **Alta calidad de gráficos y sonido.** Gráficos vectoriales suavizados, animaciones, gráficos en 3D basados en OpenGL. Incorpora los codecs estándares más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC, etc [2].

[2]. <http://www.androidcurso.com/index.php/recursos/31-unidad-1-vision-general-y-entorno-de-desarrollo/97-que-hace-a-android-especial>

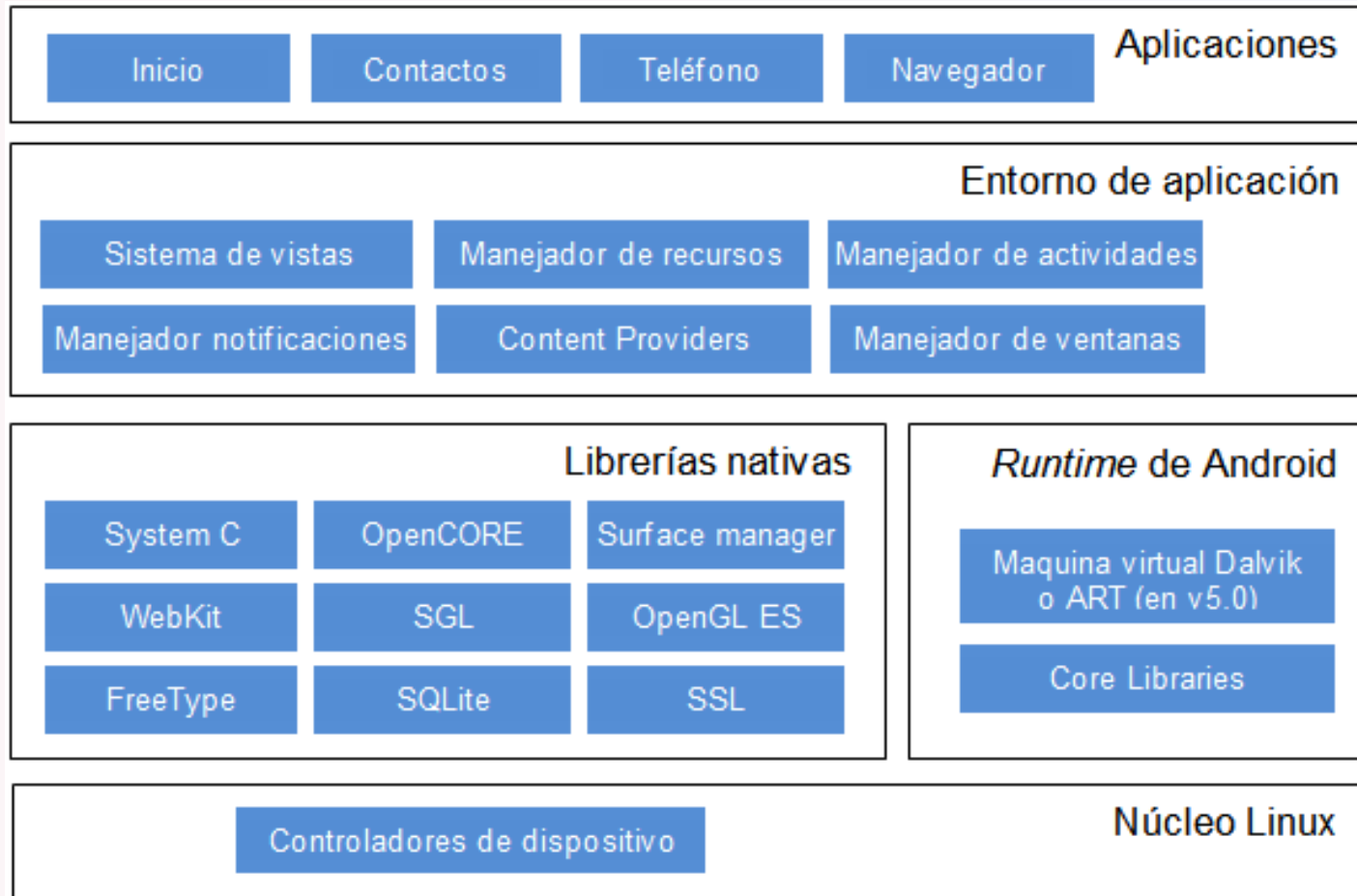




3. Arquitectura de Android

Está formada por cuatro capas:

Una de las características más importantes es que todas las capas están basadas en **software libre**.



El **núcleo** de Android está formado por el **sistema operativo Linux versión 2.6**. Esta capa proporciona servicios como la **seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos**.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Tomado de:

<http://www.androidcurso.com/index.php/recursos/31-unidad-1-vision-general-y-entorno-de-desarrollo/99-arquitectura-de-android>

<https://developer.android.com/guide/platform?hl=es-419>





3. Arquitectura de Android

Marco de trabajo de la API de Java

Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que necesitas para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- Un sistema de vista enriquecido y extensible que puedes usar para compilar la IU de una app; se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de recursos que te brinda acceso a recursos sin código, como strings localizadas, gráficos y archivos de diseño.
- Un administrador de notificaciones que permite que todas las apps muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que administra el ciclo de vida de las apps y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las apps accedan a datos desde otras apps, como la app de Contactos, o compartan sus propios datos.
- Los desarrolladores tienen acceso total a las mismas API del marco de trabajo que usan las apps del sistema Android.





4. Descarga e instalación

Entorno de desarrollo Integrado (IDE): Android Studio

Este es el ambiente de desarrollo oficial para la plataforma Android y usado por los programadores para el desarrollo de APPs ANDROID. Su lenguaje de programación es **Java**, Kotlin y C++ y utiliza una interfaz de diseño amigable.

Emulador rápido

Simula diferentes configuraciones
De hardware de diferentes dispositivos Android.

Código inteligente

Comprueba el código mas rápido y
detecta los errores de sintaxis antes de
que sea compilado.

APK Analyzer

Reduce el tamaño de las APPs y
comprueba los archivos.

Editor de diseño visual

Crea configuraciones interfaz de usuario (IU)
complejas y las prueba en cualquier tipo y
tamaño de pantalla.



Tomado de: <https://developer.android.com/studio/intro>





4. Descarga e instalación

Android Studio ofrece las siguientes funciones:

- Un sistema de compilación flexible basado en **Gradle**.
- Un emulador rápido y cargado de funciones.
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android.
- Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla.
- Integración con **GitHub** y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con **Google Cloud Platform**, que facilita la integración con Google Cloud Messaging y App Engine.

<https://developer.android.com/studio/intro>



UNIVERSIDAD
SERGIO ARBOLEDA

Misión
TIC 2022



4. Descarga e instalación

Descarga:

El **IDE** puede ser descargado del sitio oficial de Android:

https://www.google.com/aclk?sa=l&ai=DChcSEwiU79KKw6ryAhVQgVoFHQsLC20YABAAGgJ2dQ&sig=AOD64_2AUPvnT-v2PpvPtwhLaZUn6LYC8g&q&nis=1&adurl&ved=2ahUKEwjg7syKw6ryAhUIRDABHfobA6UQ0Qx6BAgCEAE

☒ He leído y acepto los términos y condiciones anteriores

Descargar Android Studio 2020.3.1 para Windows

Descargar Android Studio 2020.3.1 para Windows

android-studio-2020.3.1.22-windows.exe

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





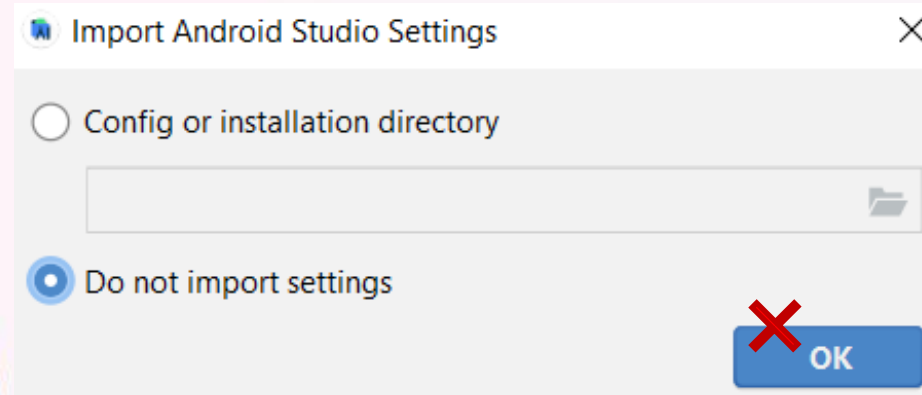
4. Descarga e instalación

Instalación:

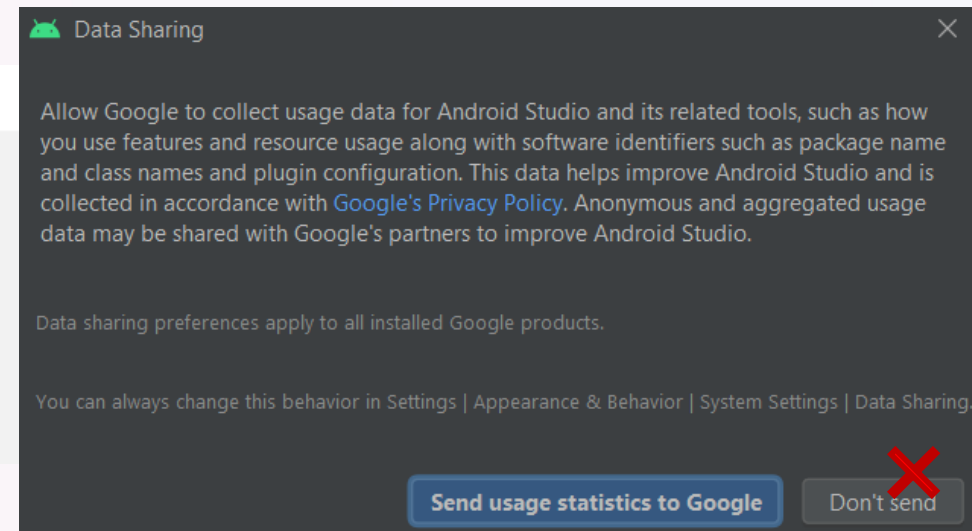
Una vez termine la descarga, el siguiente paso es dar click y darle permisos de administrador para comenzar la instalación. De ahí en adelante la instalación es: Next>, Next>, Next>, Install, Next>, Finish.

<<Don't send>>

<<OK>>



<https://developer.android.com/studio/install>



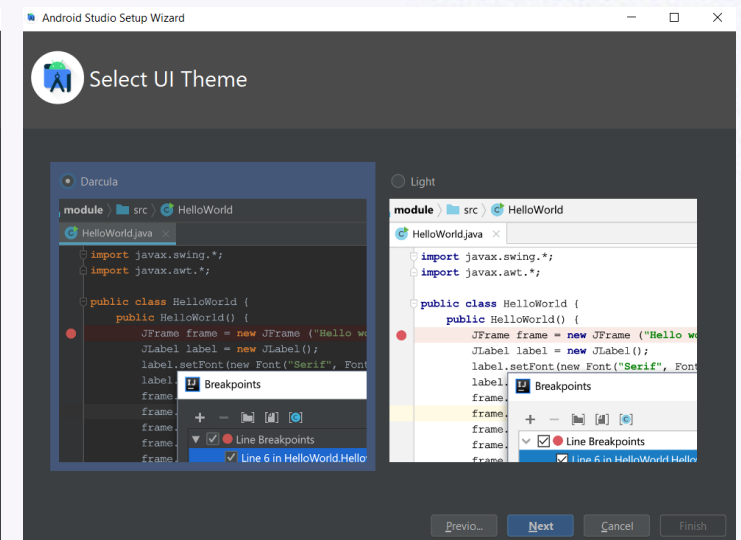
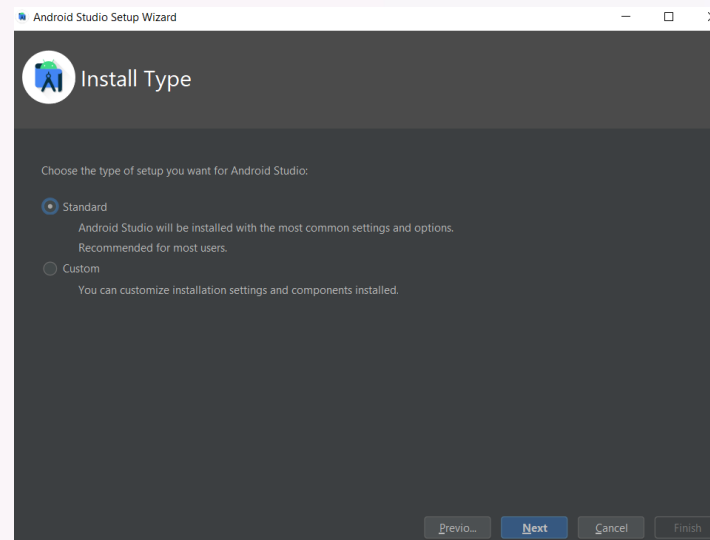
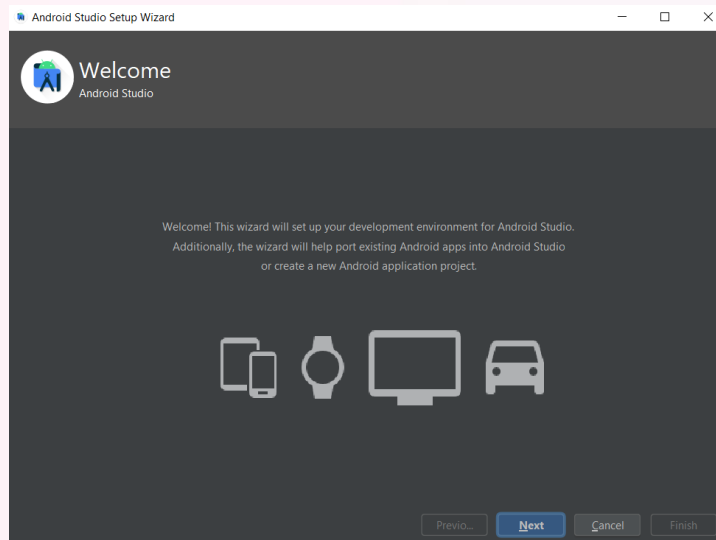


El futuro digital
es de todos

MinTIC

1. Descarga e instalación

Instalación:



De ahí en adelante la instalación es:
Next>, Next>, Next>, Finish.
Luego comienza la instalación,
dar permiso de administrador y Finish.
<https://developer.android.com/studio/install>



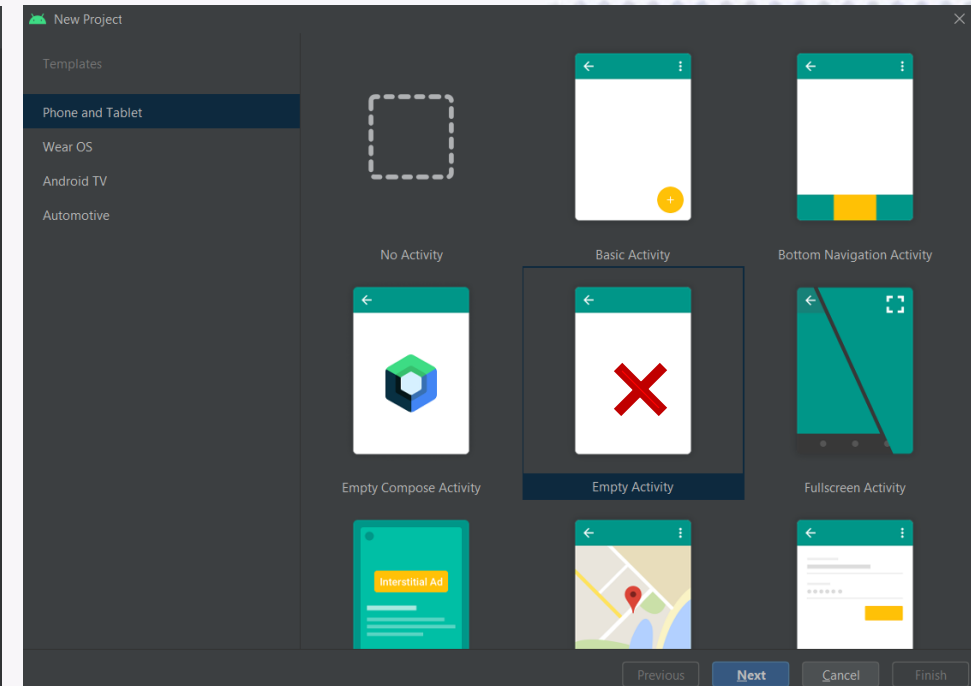
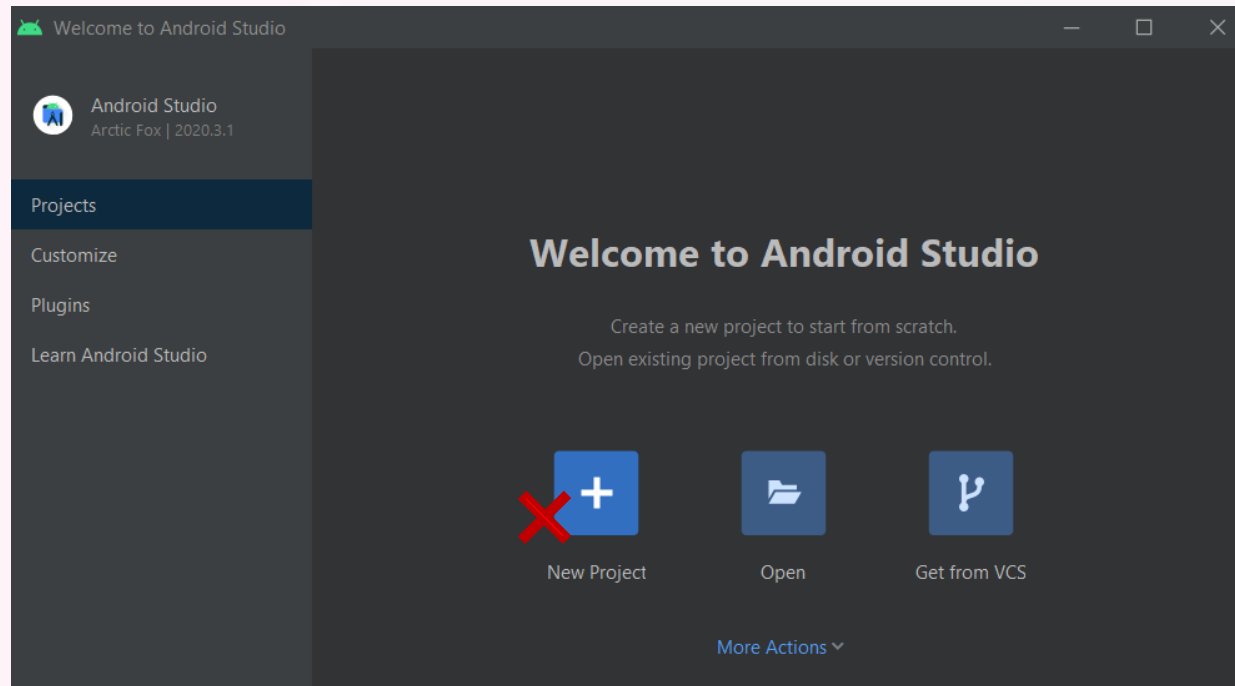
UNIVERSIDAD
SERGIO ARBOLEDA

Misión
TIC 2022



4. Cómo crear un proyecto de Android

Ventana de inicio y nuevo proyecto:



Dar click en <<New Project>>
y luego seleccionar <<Empty Activity>> y Next >

Tomado de: <https://developer.android.com/training/basics/firstapp/creating-project?hl=es>





4. Cómo crear un proyecto de Android

Configuración de un nuevo proyecto:

Nombre: MiPrimerPrograma

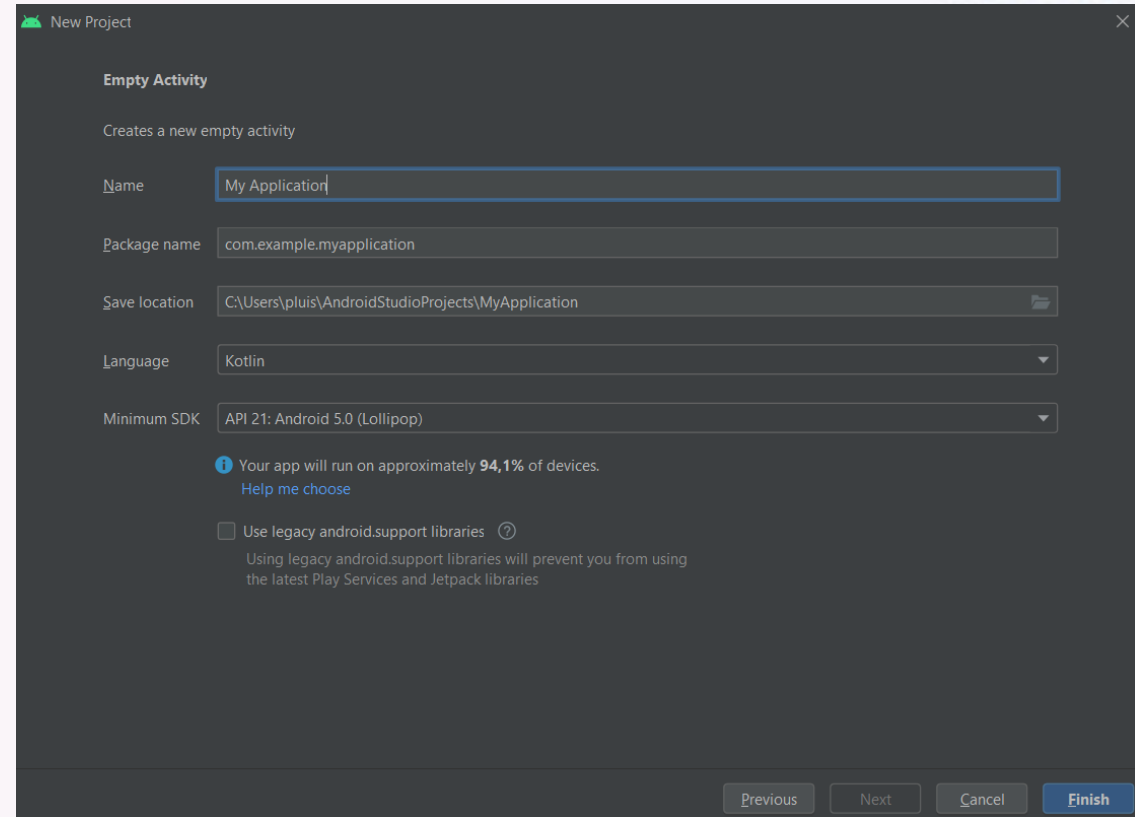
Nombre del paquete: usa.sesion1.miprimerprograma

Lenguaje: Java

El resto de la configuración queda igual.

Luego dar click en Finish.

Tarda un poco en construir el proyecto....





El futuro digital
es de todos

MinTIC

```
1 package usa.sesion1.miprimerprograma;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
```

MiPrimerPrograma – MainActivity.java
Archivo generado: MiPrimerPrograma.apk



UNIVERSIDAD
SERGIO ARBOLEDA

**Mision
TIC2022**



5. MiPrimerPrograma

Revisar que la ventana **Project** esté abierta (selecciona **View > Tool Windows > Project**) y que la vista Android esté seleccionada en la lista desplegable de la parte superior de la ventana.

Podrás ver los siguientes archivos:

app > java > usa.sesion1.miprimerprograma > MainActivity

Esta es la actividad principal. Es el punto de entrada de tu app. Cuando compilas y ejecutas la app, el sistema inicia una instancia del elemento Activity y carga su diseño.

app > res > layout > activity_main.xml

Este archivo en formato XML define el diseño de la interfaz de usuario (IU) de la actividad. Contiene un elemento **TextView** con el texto "**Hello, World!**".

app > manifests > AndroidManifest.xml

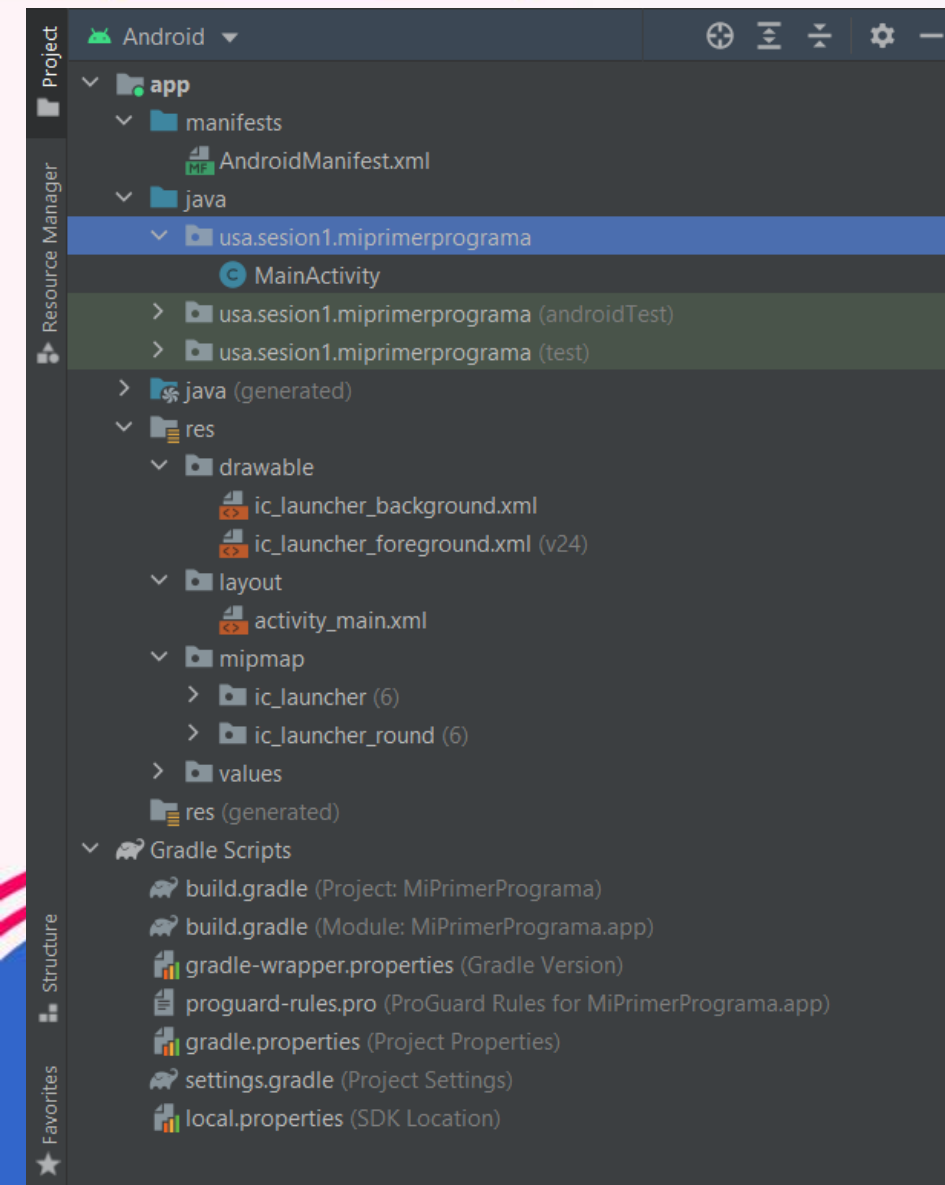
En el archivo de manifiesto, se describen las características fundamentales de la app, y se define cada uno de sus componentes.

Gradle Scripts > build.gradle

Hay dos archivos con este nombre: uno para el proyecto ("Project: MiPrimerPrograma") y otro para el módulo de la app ("Module: MiPrimerPrograma"). Cada módulo tiene su propio archivo build.gradle, pero este proyecto, por el momento, tiene un solo módulo. Usa el archivo build.gradle de cada módulo para controlar cómo el complemento de Gradle compila tu app. consulta [Cómo configurar tu compilación](https://developer.android.com/training/basics/firstapp/creating-project?hl=es).

Para ejecutar la app, continúa con la sesión 2. Cómo ejecutar tu app.

Tomado de: <https://developer.android.com/training/basics/firstapp/creating-project?hl=es>





Preguntas

