

# 2º DAM

*Programación Multimedia y  
Dispositivos Móviles*

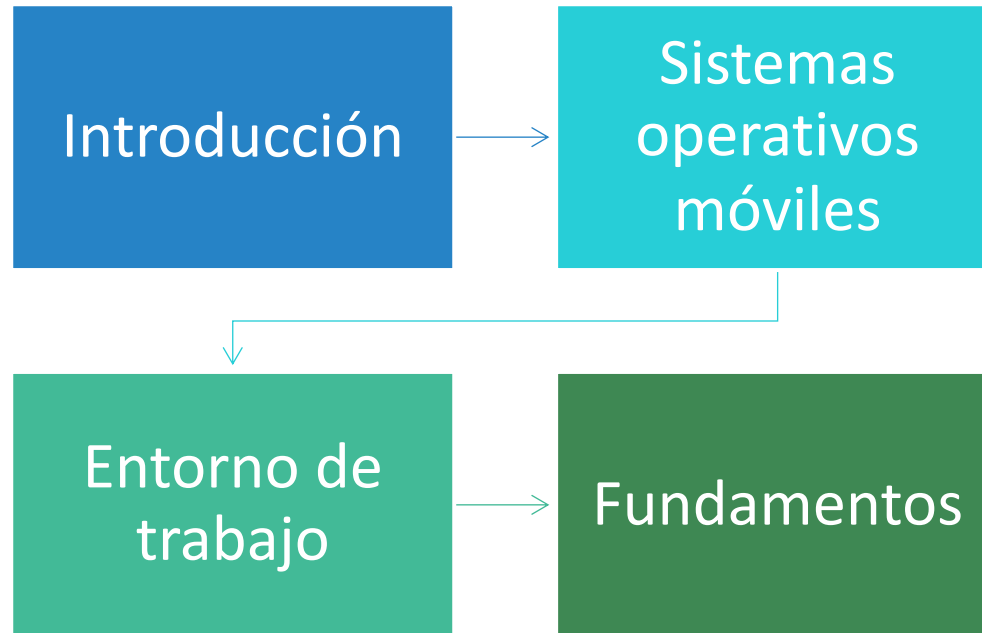
**Programación Android**



**android**

*José A. Lara*

# Contenido



# *Introducción*



# Programación Android

## Introducción

Dispositivos móviles: Un dispositivo móvil es un elemento portable con ciertas capacidades de procesamiento y de comunicación con otros procesos en otros dispositivos o máquinas remotas.

Funcionalidades de un dispositivo móvil

- Llamadas
- SMS
- Navegación WAP (2ª Generación) – Páginas WML
- Tecnología GPRS (Generación 2.5) – Mayor ancho de banda que GSM
- Versión 2 de WAP (basada en XHTML): irrumpen las PDA (Windows Mobile)
- Tarifas planas
- Evolución de móviles a Smartphones (iPhone - conexión a través de operador o WiFi)
- Aparecen las Tablets (iPad y demás tablets)



# *Sistemas operativos móviles*



# Programación Android

## ***Sistemas operativos móviles***

### ***iOS – Apple***

- Fue desarrollado originalmente para el iPhone, siendo después usado en otros dispositivos de Apple como iPod Touch e iPad.
- Se trata de un derivado de Mac OS X, denominado iPhone OS en sus inicios y orientado a la arquitectura ARM.
- En la actualidad está a punto de publicarse iOS 16
- Para iPad hoy día tenemos iPadOS
- Para ordenadores Mac OS está a punto de cambiar a la versión MacOS 13 – Ventura.
- Después de casi 20 años con MacOS X (versiones 0 a 15). La versión 10.0 se publicó en marzo de 2001, en 2021 se publicó MacOS 11 (Big Sur)



# Programación Android

## *Sistemas operativos móviles*

### *iOS – Apple*

- Existe un SDK integrado en el entorno de desarrollo gratuito de Apple: XCode
- Permite el desarrollo de aplicaciones utilizando el lenguaje de programación Objective-C y Swift así como la realización de pruebas en el iPhone Simulator.
- La arquitectura iOS está basada en capas de abstracción, donde las capas más altas contienen los servicios y tecnologías más importantes para el desarrollo de aplicaciones, y las capas más bajas controlan los servicios básicos.

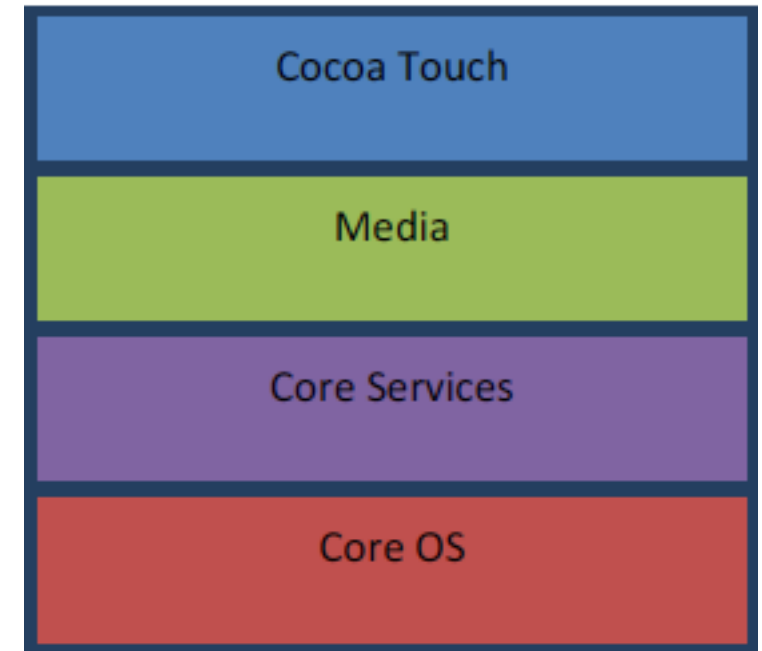


# Programación Android

## Sistemas operativos móviles

### iOS – Apple

- La capa de más alto nivel es Cocoa Touch, que contiene los frameworks para desarrollo de aplicaciones: UIKit con las clases para interfaz de usuario y Foundation Framework con las clases básicas y de acceso a servicios del sistema operativo.
- Por debajo de ella se encuentran la capa Media de servicios gráficos.
- La capa Core Services de servicios comunes a las aplicaciones.
- Por último la capa Core OS contiene todas las características de bajo nivel como acceso a ficheros y memoria.





# Programación Android

## *Sistemas operativos móviles*

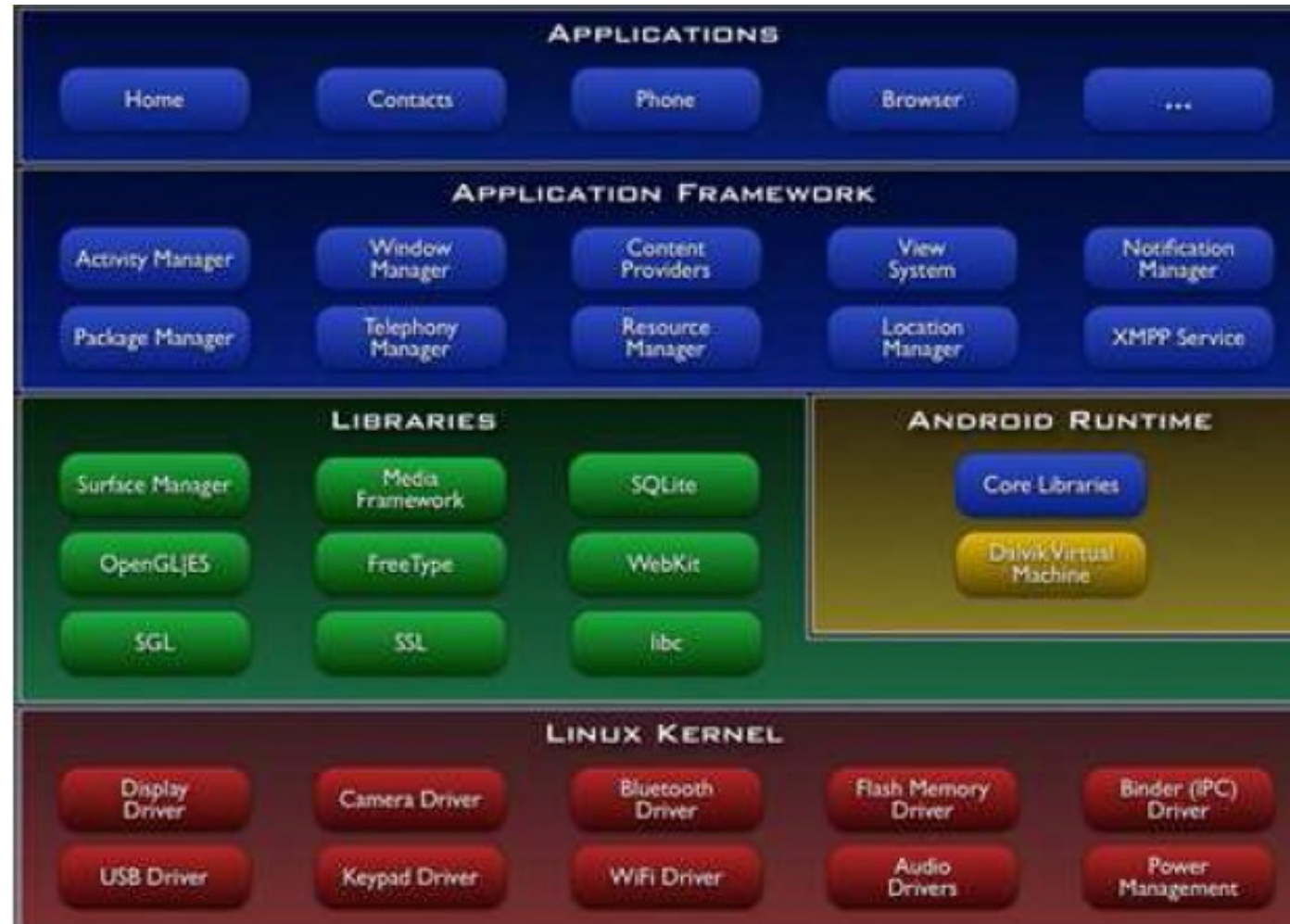
### *Android*

- Es un sistema operativo completo, libre y abierto basado en Linux y orientado a dispositivos móviles.
- Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005, y su desarrollo fue continuado por Google y la Open Handset Alliance.
- Utiliza los lenguajes de programación Java y Kotlin.
- Dispone de una gran comunidad de desarrolladores creando aplicaciones que permiten extender la funcionalidad de los dispositivos.
- Las aplicaciones Android se desarrollan con las librerías Java SE, y se ejecutan sobre una máquina virtual Dalvik.
- No se trata de una máquina virtual Java, ya que no opera con bytecodes sino con archivos dex propios.



# Programación Android

## Sistemas operativos móviles Android



# Programación Android

## *Sistemas operativos móviles*

### *Android*

Las versiones de Android recibían hasta la versión 10 el nombre de diferentes postres en inglés. En cada versión el postre elegido empieza por una letra distinta siguiendo el orden alfabético.

A: Apple Pie (v1.0): Tarta de manzana  
B: Banana Bread (v1.1): Pan de plátano  
C: Cupcake (v1.5): Magdalena  
D: Donut (v1.6): Rosquilla  
E: Éclair (v2.0/v2.1): Pastel francés  
F: Froyo (v2.2) (abreviatura de «frozen yogurt»): Yogur helado  
G: Gingerbread (v2.3): Pan de jengibre  
H: Honeycomb (v3.0/v3.1/v3.2): Panal de miel  
I: Ice Cream Sandwich (v4.0): Sándwich de helado

J: Jelly Bean (v4.1/v4.2/v4.3): Gominola  
K: KitKat (v4.4): Kit Kat  
L: Lollipop (v5.0): Chupa-chups  
M: Marshmallow (v6.0): Nube (esponja)  
N: Nougat (v7.0): Nougat  
O: Oreo (v8.0): Oreo  
P: Pie (v9.0): Tarta  
Android 10 (Q): Septiembre de 2019  
Android 11 (R): Septiembre de 2020  
Android 12 (Snow Cone): Octubre de 2021  
Android 13 (Tiramisu): Versión actual (agosto 2022)



# Programación Android

## Sistemas operativos móviles Android

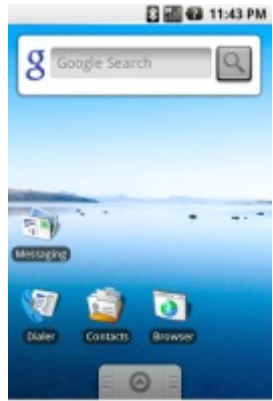
	Versión preliminar
	Última versión
	Versión antigua pero vigente
	Versión descontinuada

Nombre código ↕	Número de versión ↕	Fecha de lanzamiento ↕	Nivel de API ↕
Apple Pie <sup>56</sup>	1.0	23 de septiembre de 2008	1
Banana Bread <sup>56</sup>	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb <sup>57</sup>	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1.2	15 de junio de 2016	24 – 25
Oreo	8.0 – 8.1	21 de agosto de 2017	26 – 27
Pie	9.0	6 de agosto de 2018	28
10	10.0	3 de septiembre de 2019	29
11	11.0	8 de septiembre de 2020	30
12	12.0 - 12L	4 de octubre de 2021	31 - 32
13	13.0	15 de agosto de 2022	33

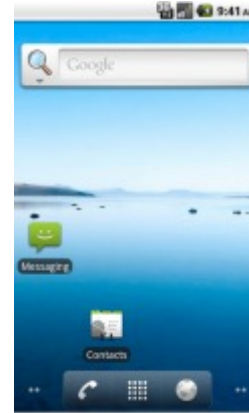


# Programación Android

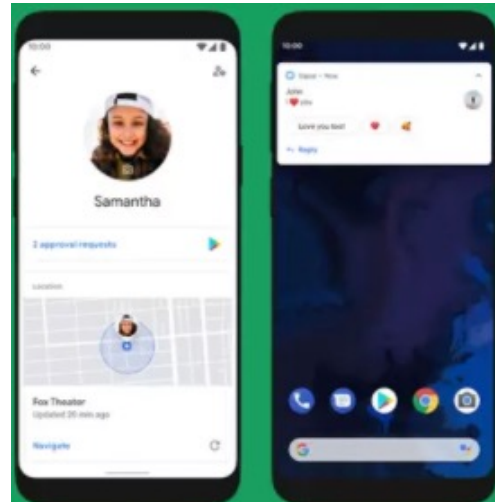
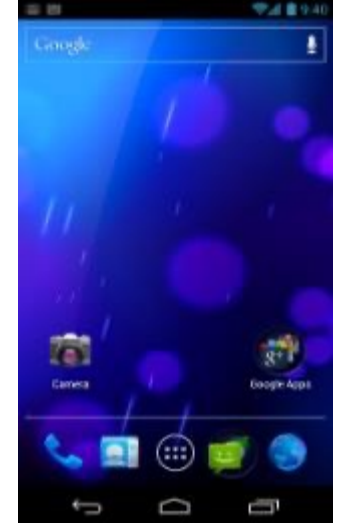
## Sistemas operativos móviles Android



ANDROID 1.1



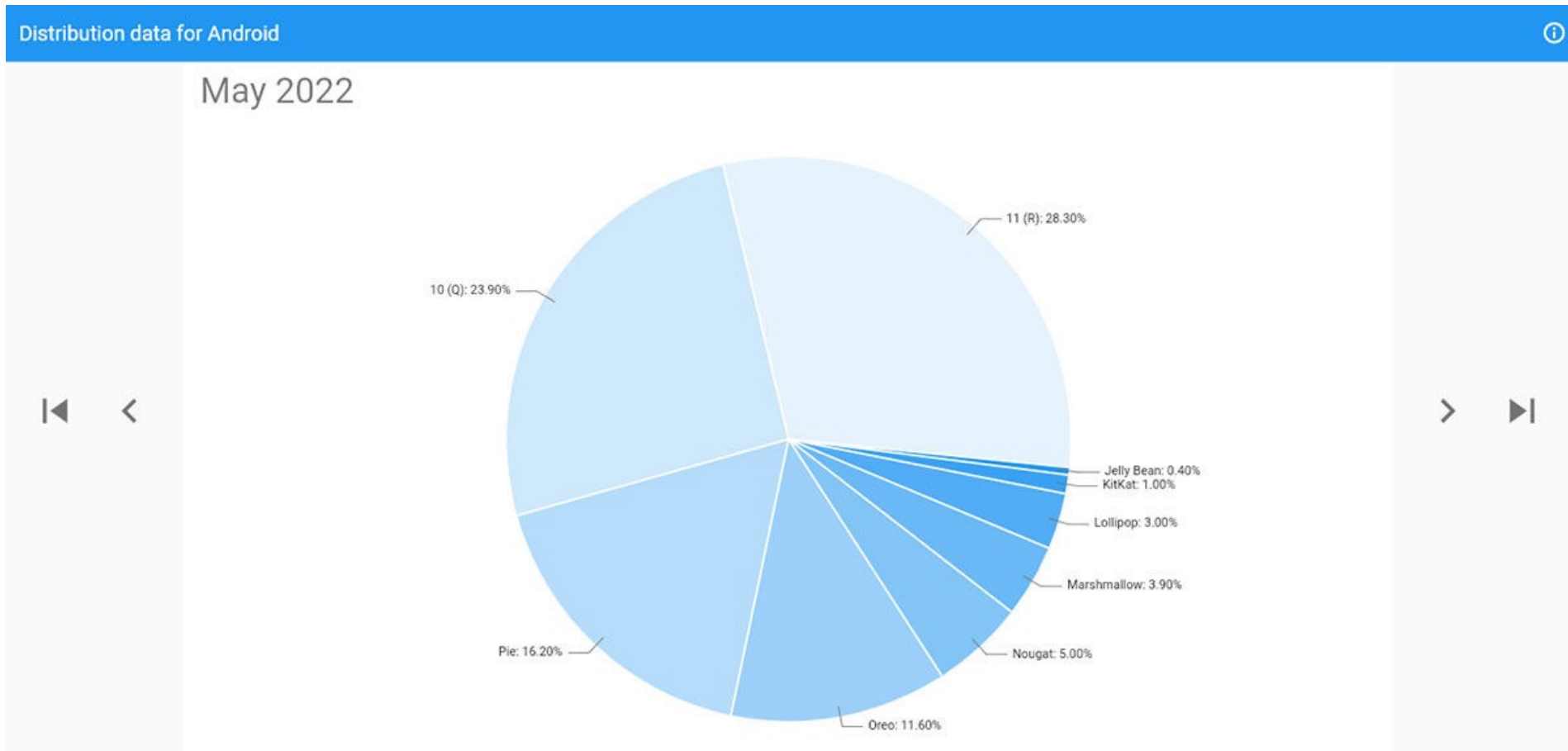
ANDROID 2.2





# Programación Android

## Sistemas operativos móviles Android



# Programación Android

## Sistemas operativos móviles

### Android

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,9%
4.3 Jelly Bean	18	99,7%
4.4 KitKat	19	99,7%
5.0 Lollipop	21	98,8%
5.1 Lollipop	22	98,4%
6.0 Marshmallow	23	96,2%
7.0 Nougat	24	92,7%
7.1 Nougat	25	90,4%
8.0 Oreo	26	88,2%
8.1 Oreo	27	85,2%
9.0 Pie	28	77,3%
10. Q	29	62,8%
11. R	30	40,5%
12. S	31	13,5%



*Entorno de trabajo*



**android**





# Programación Android

## Entorno de trabajo – Android Studio

- Es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.
- La versión actual es Android Studio Chipmunk 2021.2.1 Patch 2 (mayo de 2022). Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, MacOS y GNU/Linux.

Version	Release date
Arctic Fox (2020.3.1)	July 2021 <sup>[26]</sup>
4.2	May 2021 <sup>[27]</sup>
4.1	Oct 2020 <sup>[28]</sup>
4.0	May 2020
3.6	February 2020
3.5	August 2019
3.4	April 2019 <sup>[29]</sup>
3.3	January 2019
3.2	September 2018
3.1	March 2018
3.0	October 2017
2.3	March 2017
2.2	September 2016
2.1	April 2016
2.0	April 2016
1.5	November 2015
1.4	September 2015
1.3	July 2015
1.2	April 2015
1.1	February 2015
1.0	December 2014

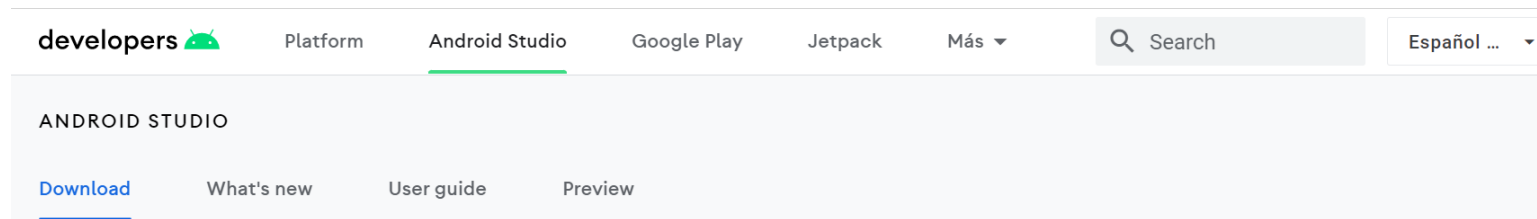


# Programación Android

## Entorno de trabajo – Android Studio

### Descarga de Android Studio

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



Android Studio provides the fastest tools for building apps on every type of Android device.

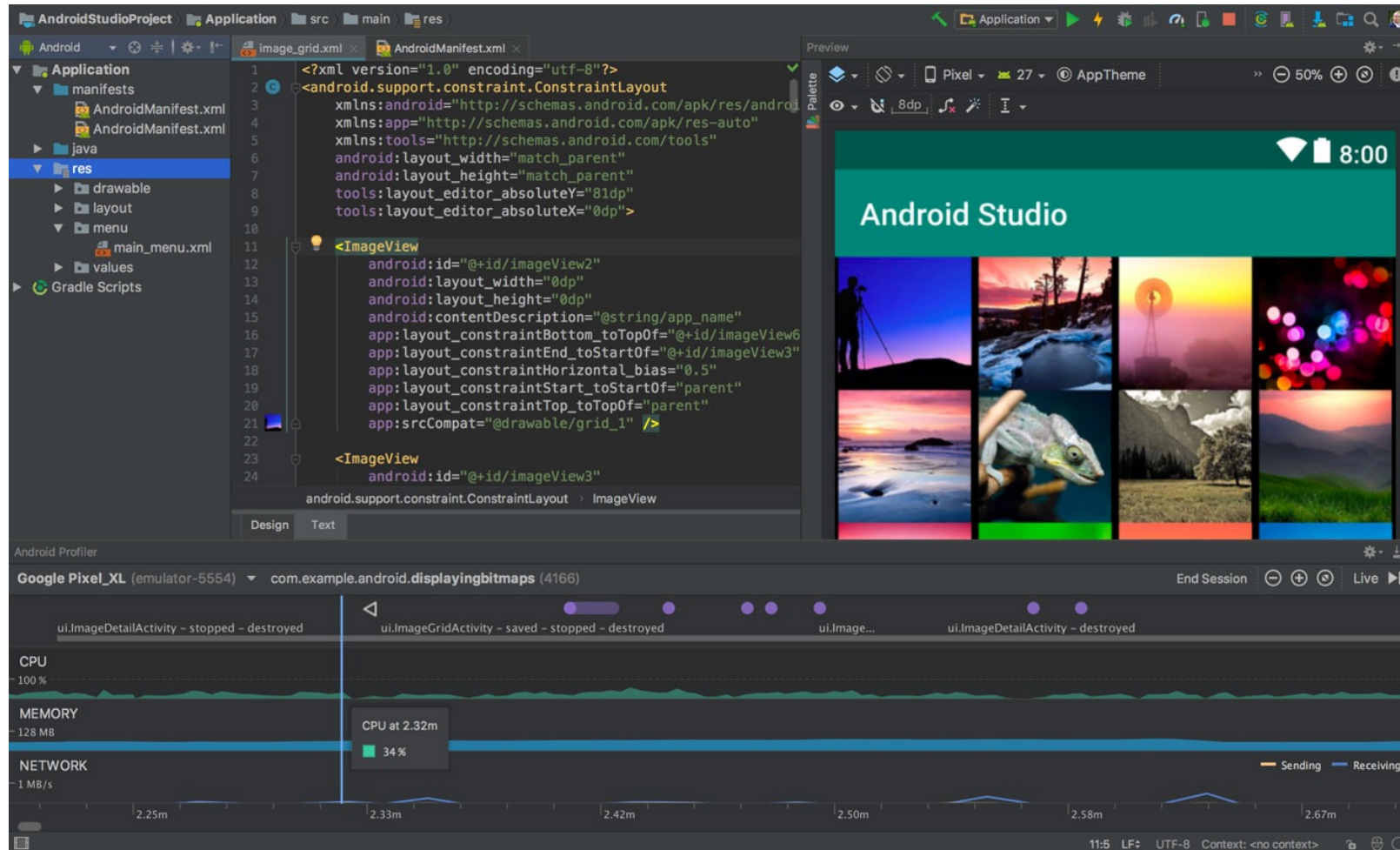
Download Android Studio

2020.3.1 for Windows 64-bit (912 MiB)



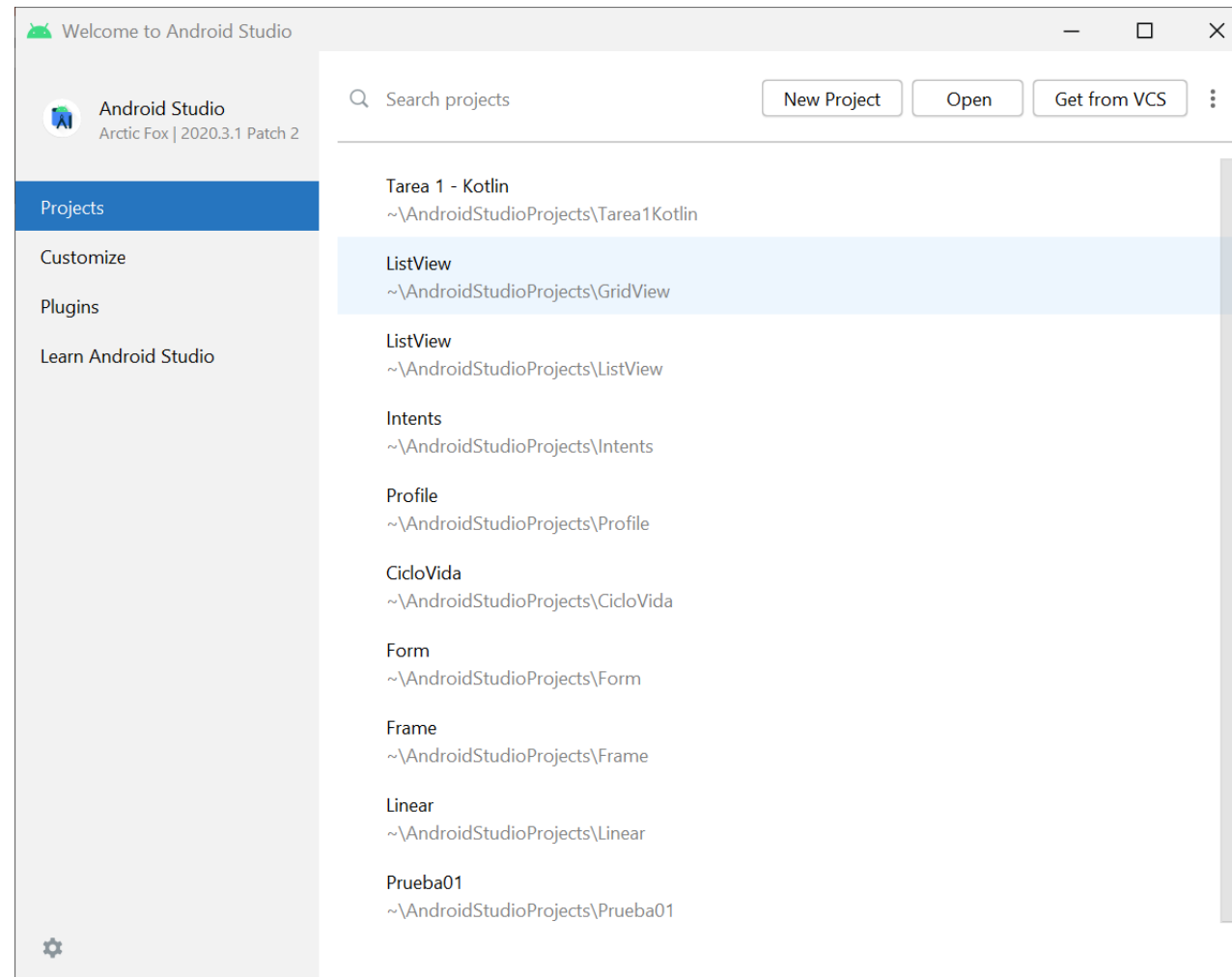
# Programación Android

## Entorno de trabajo – Android Studio



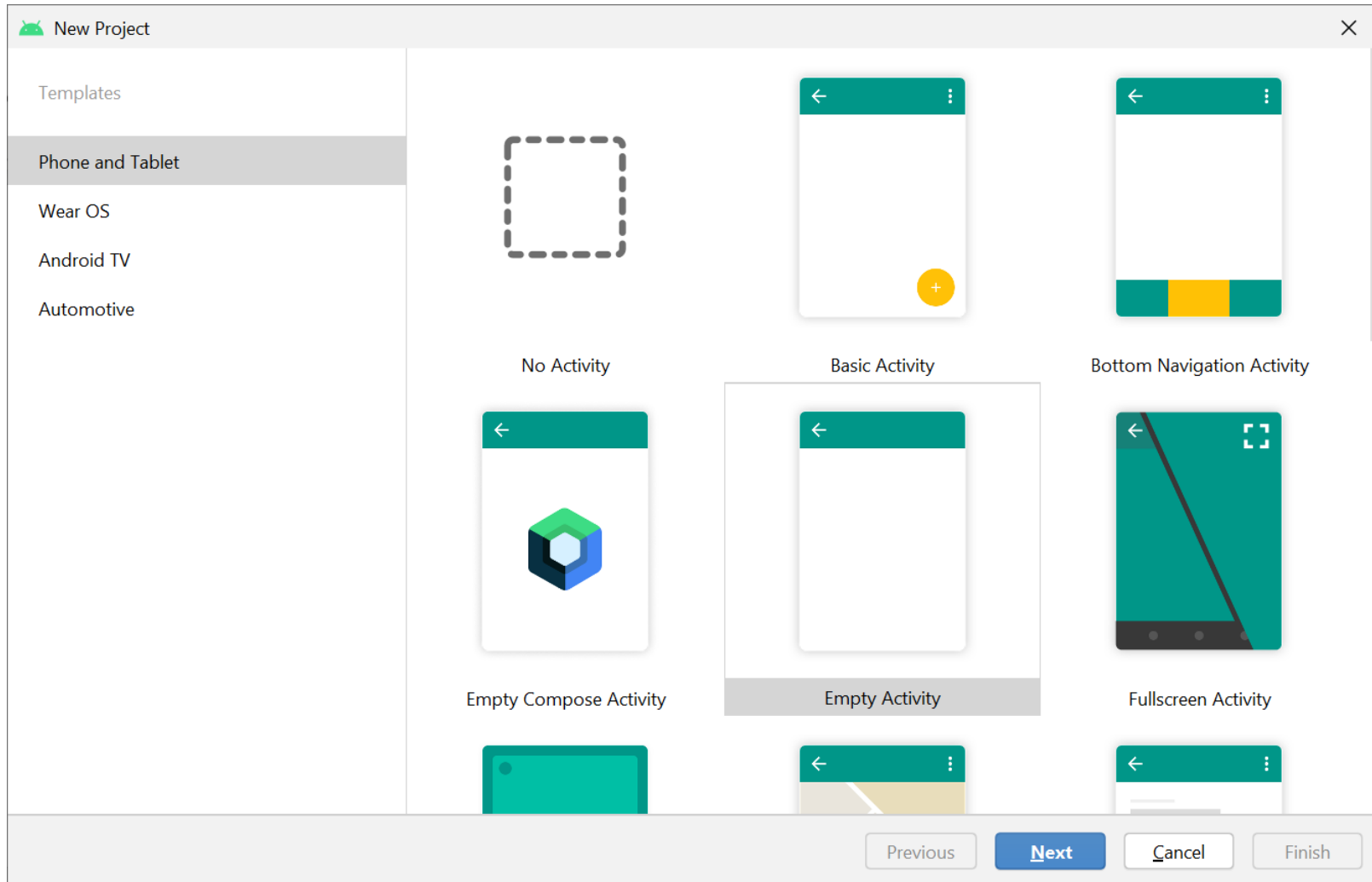
# Programación Android

## Entorno de trabajo – Android Studio




# Programación Android

## Entorno de trabajo – Android Studio



# Programación Android

## Entorno de trabajo – Android Studio


 New Project ✕

**Empty Activity**

Creates a new empty activity


Name


Package name

Save location  

Language  ▼

Minimum SDK  ▼

 Your app will run on approximately **94,1%** of devices.  
[Help me choose](#)

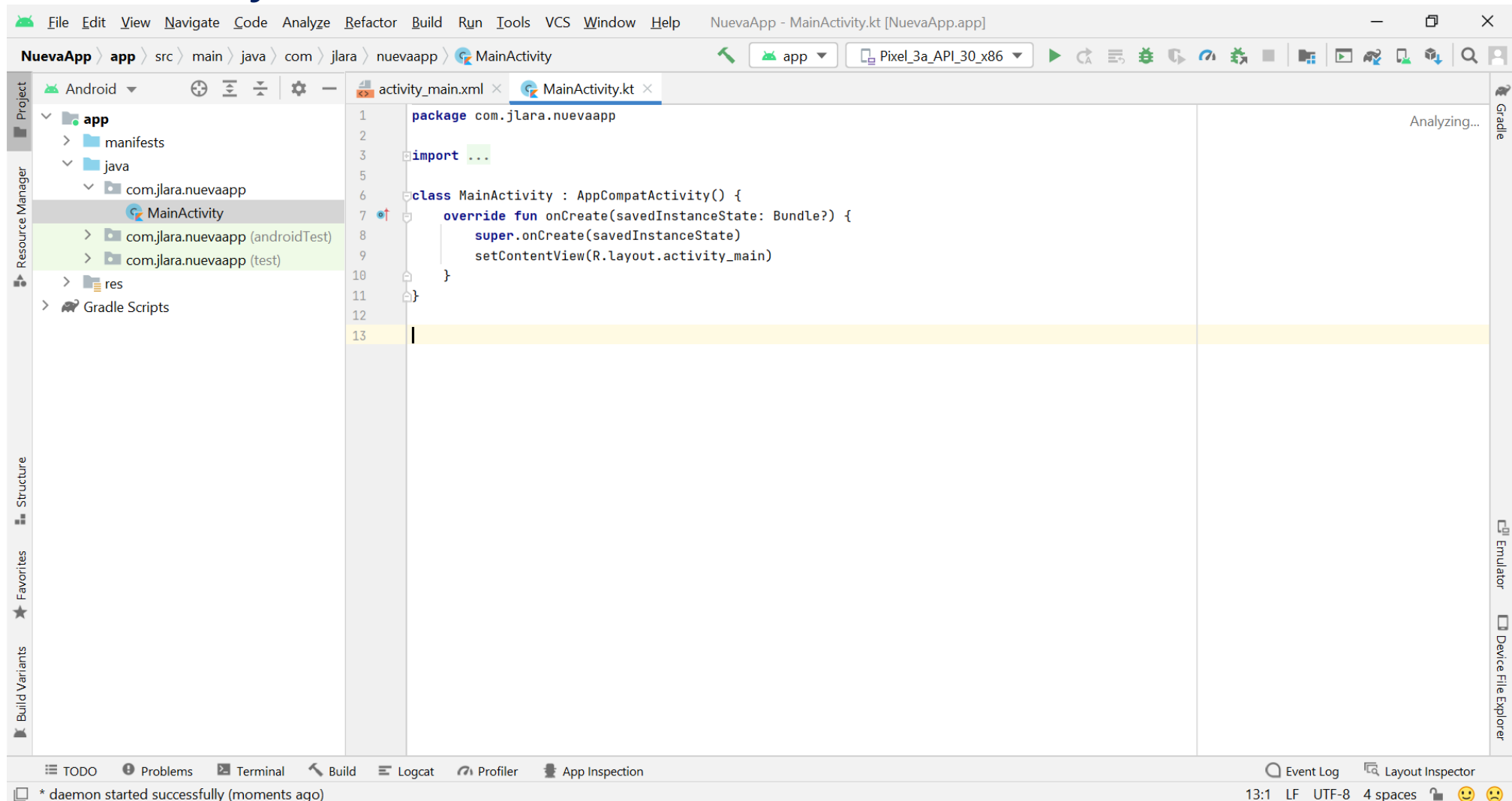
☐ Use legacy android.support libraries 

Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries



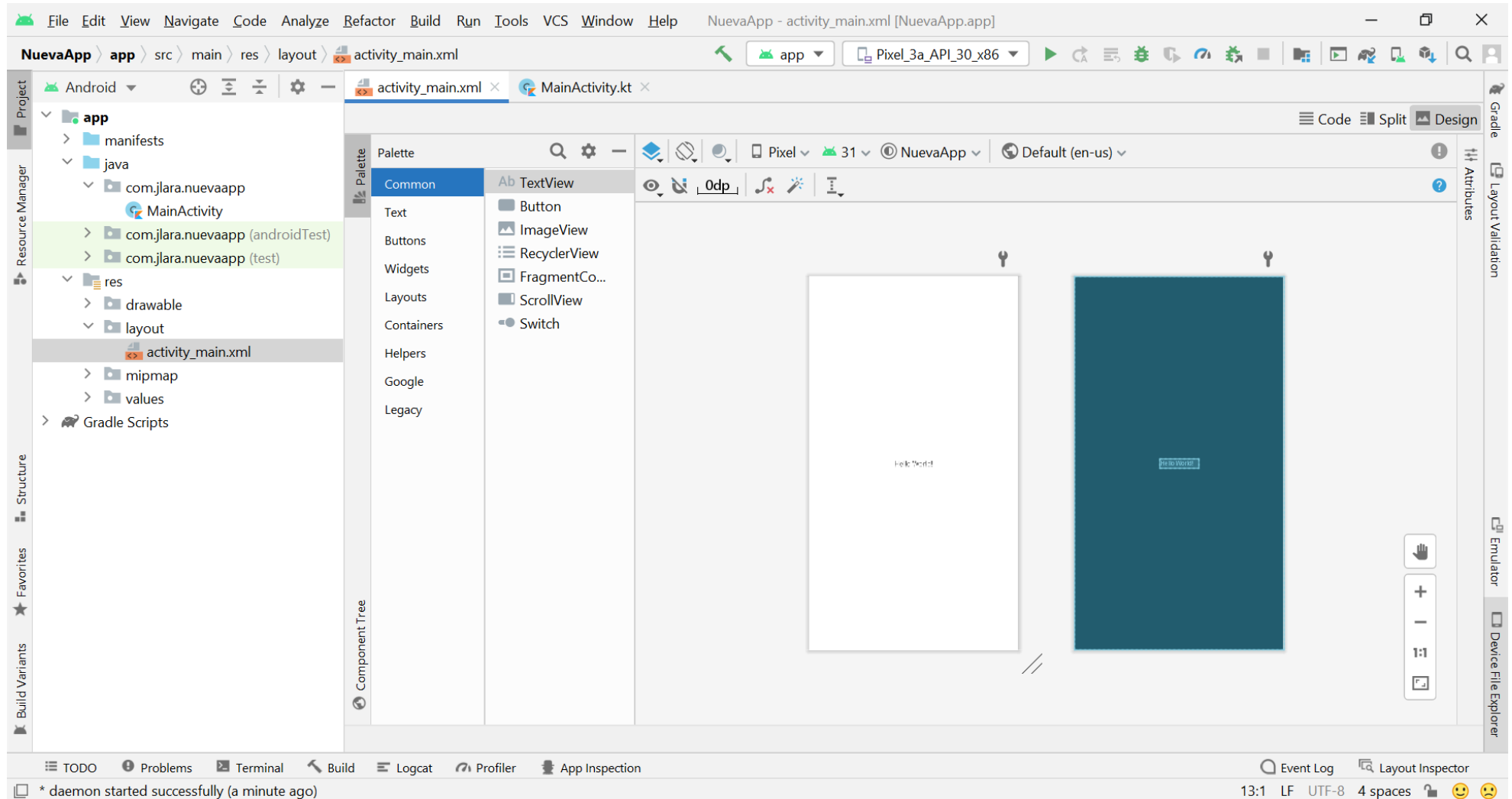
# Programación Android

## Entorno de trabajo – Android Studio



# Programación Android

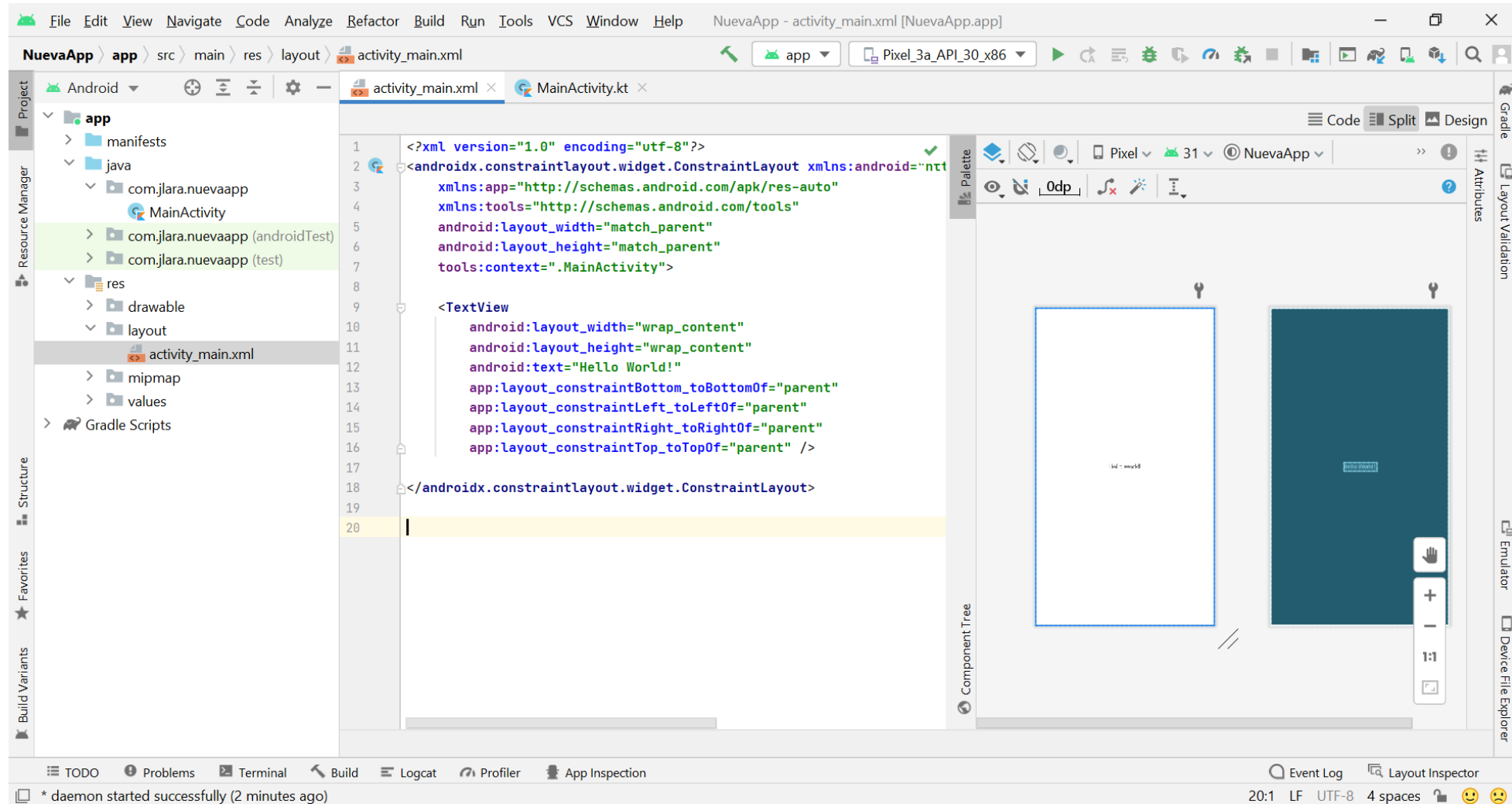
## Entorno de trabajo – Android Studio





# Programación Android

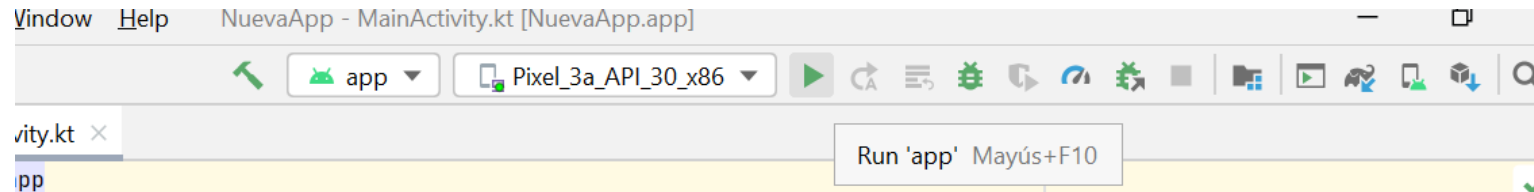
## Entorno de trabajo – Android Studio



# Programación Android

## Entorno de trabajo – Android Studio

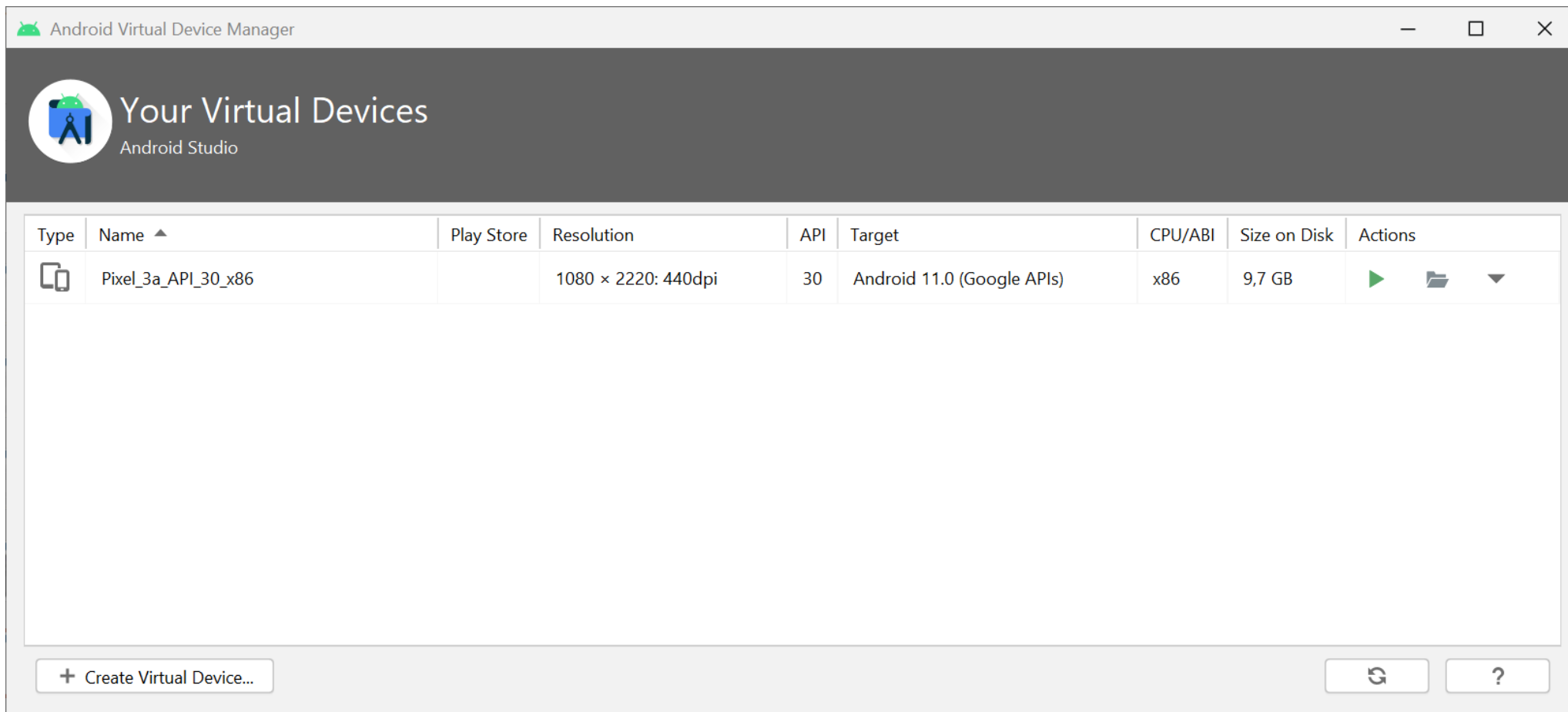
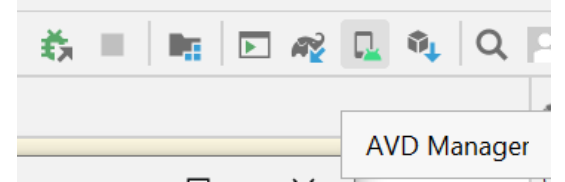
Ejecución de una app



# Programación Android

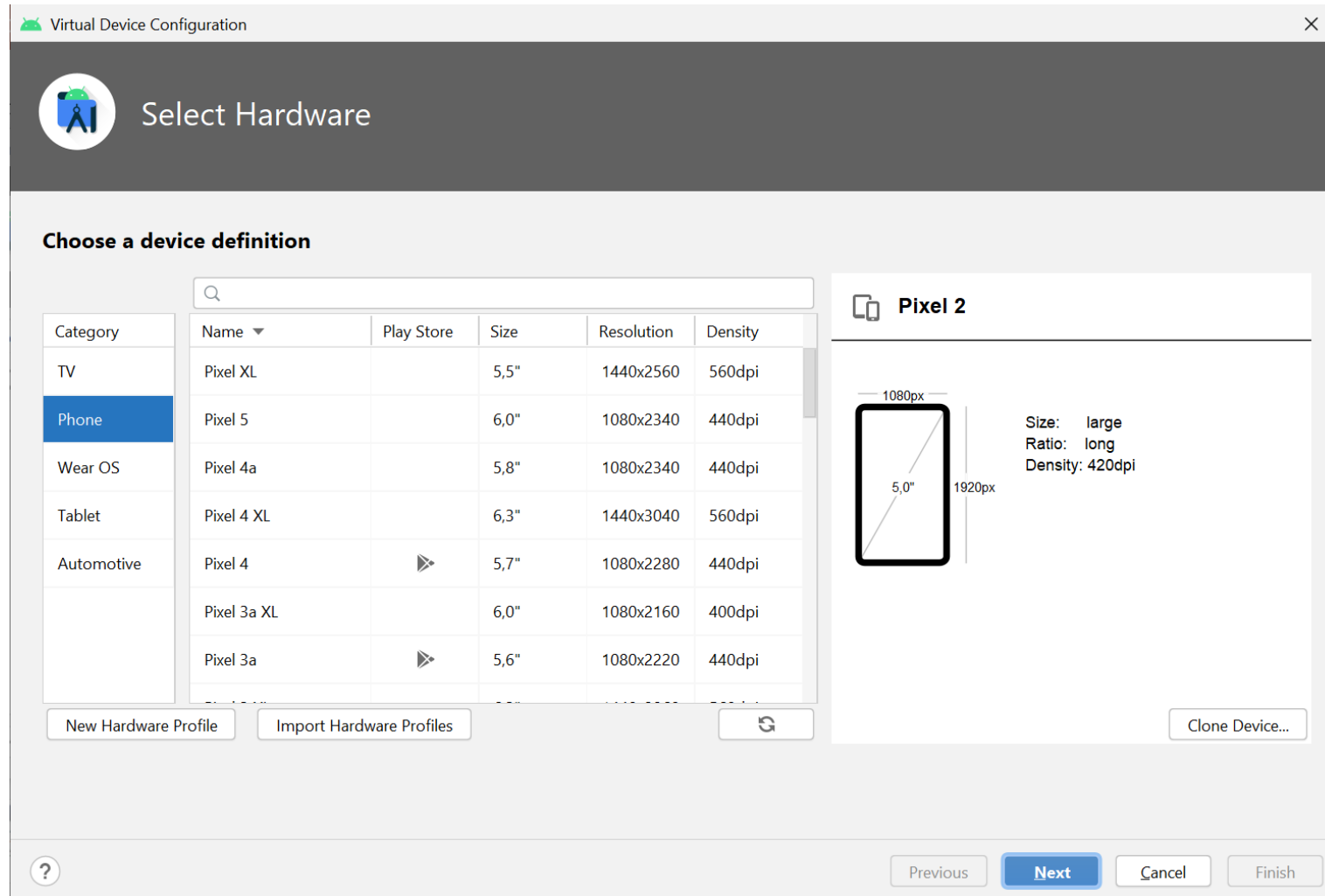
## Entorno de trabajo – Android Studio

Crear un dispositivo virtual



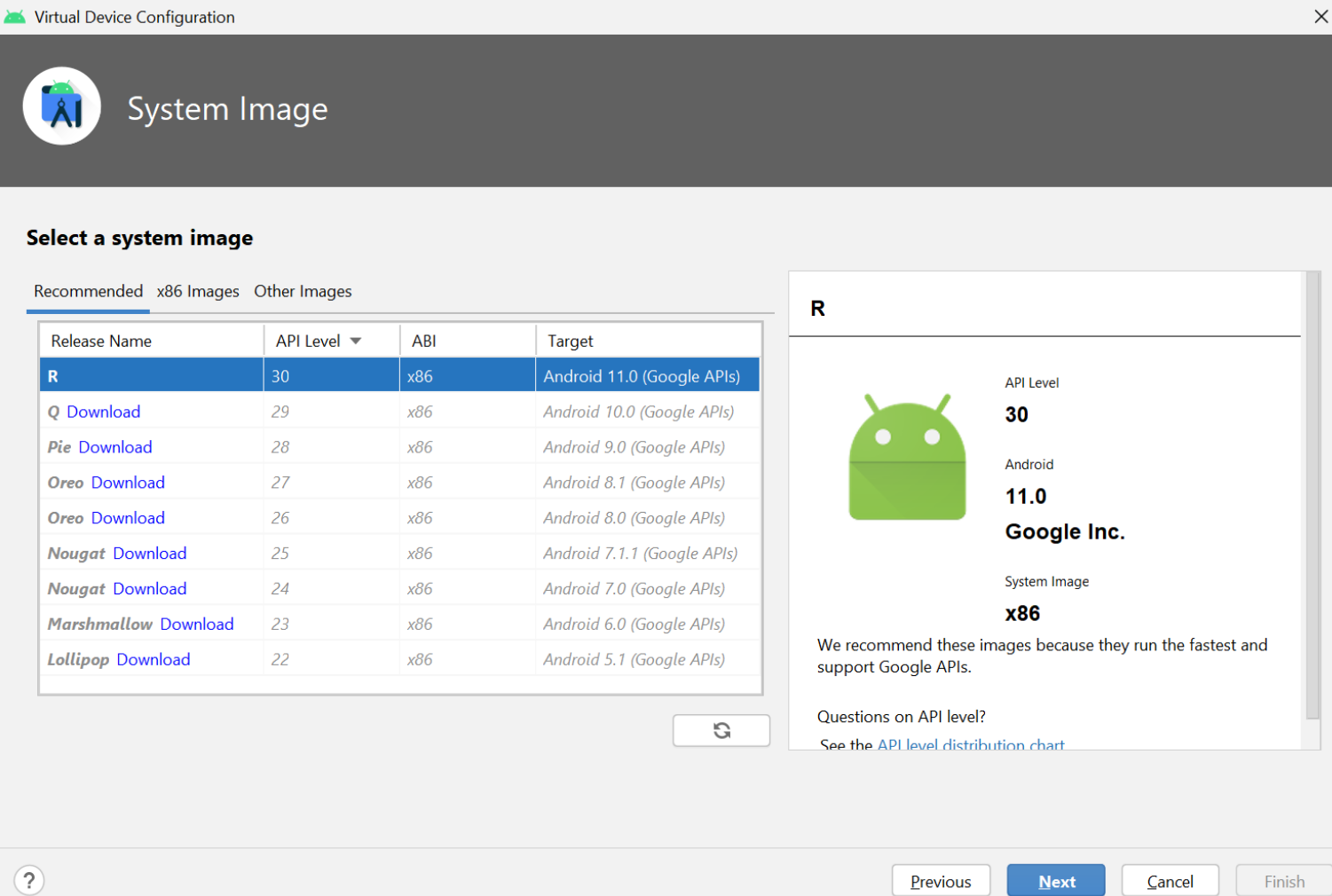
# Programación Android

## Entorno de trabajo – Android Studio



# Programación Android

## Entorno de trabajo – Android Studio



Virtual Device Configuration

System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API Level ▼	ABI	Target
<b>R</b>	30	x86	Android 11.0 (Google APIs)
<a href="#">Q Download</a>	29	x86	Android 10.0 (Google APIs)
<a href="#">Pie Download</a>	28	x86	Android 9.0 (Google APIs)
<a href="#">Oreo Download</a>	27	x86	Android 8.1 (Google APIs)
<a href="#">Oreo Download</a>	26	x86	Android 8.0 (Google APIs)
<a href="#">Nougat Download</a>	25	x86	Android 7.1.1 (Google APIs)
<a href="#">Nougat Download</a>	24	x86	Android 7.0 (Google APIs)
<a href="#">Marshmallow Download</a>	23	x86	Android 6.0 (Google APIs)
<a href="#">Lollipop Download</a>	22	x86	Android 5.1 (Google APIs)

API Level  
**30**

Android  
**11.0**

Google Inc.

System Image  
**x86**

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?  
See the [API level distribution chart](#).

Previous Next Cancel Finish



# Programación Android



## Entorno de trabajo – Android Studio

Virtual Device Configuration


Android Virtual Device (AVD)


### Verify Configuration

AVD Name: Pixel 5 API 30

 Pixel 5	6.0 1080x2340 xxhdpi	<a href="#">Change...</a>
 R	Android 11.0 x86	<a href="#">Change...</a>


Startup orientation

  
Portrait

  
Landscape

Emulated Performance: Graphics: Automatic

[Show Advanced Settings](#)

 The selected skin is not large enough to view the entire screen.

### Graphics Rendering

Choose how the graphics should be rendered in the emulator.

**Hardware**  
Use your computer's graphics card for faster rendering.

**Software**  
Emulate the graphics in software, use this to work around issues with your computer's graphics card.


[Previous](#) [Next](#) [Cancel](#) [Finish](#)









# Programación Android



## Entorno de trabajo – Android Studio

Android Virtual Device Manager

 Your Virtual Devices  
Android Studio

Type	Name ▲	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel_3a_API_30_x86		1080 × 2220: 440dpi	30	Android 11.0 (Google APIs)	x86	9,7 GB	  ▼
	Pixel 5 API 30		1080 × 2340: 440dpi	30	Android 11.0 (Google APIs)	x86	513 MB	  ▼

+ Create Virtual Device...



*Fundamentos*





# Programación Android

## Fundamentos

### Activity

- Es un componente que representa una **pantalla** para interactuar con el usuario.
- Una aplicación Android normalmente se compone de varias Activity, de forma que en cada momento solamente existe una de ellas activa.
- La Activity principal (main) se muestra cuando el usuario inicia la aplicación, y ésta se encargará de iniciar otras Activity en función de las necesidades de la aplicación.
- Una actividad proporciona la ventana en la que la app dibuja su IU. Por lo general, esta ventana llena la pantalla, pero puede ser más pequeña y flotar sobre otras ventanas. Generalmente, una actividad implementa una pantalla en una app.
- Para usar actividades en una app, se debe registrar información sobre estas en el manifiesto de la app y administrar los ciclos de vida de las actividades de manera apropiada.



# Programación Android

## Fundamentos

### Activity

- Existe un mecanismo en Android denominado **Back Stack** utilizado para almacenar la secuencia de llamadas a Activity, de forma que cada vez que una nueva Activity comienza, la anterior es parada y almacenada en una estructura de pila. Si el usuario pulsa el botón Atrás, la Activity anterior se extrae de la pila y se muestra.



# Programación Android

## Fundamentos

### Intent

- Es un objeto de mensajería que se puede usar para solicitar una acción de otro componente de una app. Si bien las intents facilitan la comunicación entre componentes de varias formas, existen tres casos de uso principales:
  - Iniciar una Activity
  - Iniciar un servicio (componente que realiza operaciones en segundo plano sin una interfaz de usuario)
  - Transmitir una emisión (aviso que cualquier aplicación puede recibir)



# Programación Android

## Fundamentos

### Intent

- Existen dos tipos de Intents:
  - **Explícitas:** especifican qué aplicación las administrará, ya sea incluyendo el nombre del paquete de la app de destino o el nombre de clase del componente completamente calificado. Normalmente, el usuario usa una intent explícita para iniciar un componente en su propia aplicación porque conoce el nombre de clase de la actividad o el servicio que desea iniciar.
  - **Implícitas:** no nombran el componente específico, pero, en cambio, declaran una acción general para realizar, lo cual permite que un componente de otra aplicación la maneje. Por ejemplo, si se desea mostrar al usuario una ubicación en un mapa, se puede usar una intent implícita para solicitar que otra aplicación apta muestre una ubicación específica en un mapa.



# Programación Android

## Fundamentos

### Configuración del manifiesto

- Para que una app pueda usar actividades, deben declararse, y también declarar algunos de sus atributos, en el manifiesto.
- La configuración de todos estos componentes se realiza en el fichero principal de la aplicación Android denominado **AndroidManifest.xml**. Toda aplicación tiene que poseer un fichero llamado de esta forma en su carpeta raíz, ya que contiene información básica para informar al sistema sobre la aplicación.



# Programación Android

## Fundamentos

### Configuración del manifiesto

- Para declarar tu actividad, abre tu archivo de manifiesto y agrega el elemento `<activity>` como objeto secundario del elemento `<application>`.

```
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" />
    ...
  </application ... >
  ...
</manifest >
```

- El único atributo obligatorio para este elemento es `android:name`, que especifica el nombre de la clase de la actividad. También se pueden agregar atributos que definan las características de la actividad, como una etiqueta, un icono o un tema de IU.



# Programación Android

## Fundamentos

### Configuración del manifiesto

- Las funciones principales de este fichero son las siguientes:
  - Referenciar el **paquete Java (o Kotlin)** base de la aplicación.
  - Describir los **componentes de la aplicación** y las **clases** que los implementan.
  - Determinar qué **procesos** albergarán componentes de la aplicación.
  - Declarar qué **permisos** necesita la aplicación para acceder a partes protegidas de la API e interactuar con otras aplicaciones, y permisos de otros elementos para interactuar con ella.
  - Declarar el **nivel mínimo de la API Android** que requiere la aplicación.
  - Listar las **librerías necesarias** para la aplicación.

