

U1.1. Introducción a Android y entorno de desarrollo Android Studio

- **Plataforma abierta**, basada en Linux y de código abierto. Se puede usar sin pagar licencias.
- **Adaptable a hardware diverso**, como teléfonos, tablets, relojes, TV, coches, electrodomésticos, wearables, IoT...
- **Portable**, pues al desarrollar las aplicaciones en Java o Kotlin mediante máquina virtual, estas se pueden ejecutar en cualquier CPU
- **Servicios integrados y API**, como localización basada en GPS y redes, bases de datos con SQL, reconocimiento y generación de voz, navegador, capacidades multimedia, etc.

- **Seguridad** basada en Linux y permisos para los diferentes servicios
- **Optimizado para bajo consumo de CPU y memoria** gracias a la máquina virtual Java ART de Google.
- **Gráficos y sonido**, con animaciones, gráficos 3D, códecs de audio y vídeo (H264, MP3, AAC, etc.)

android 

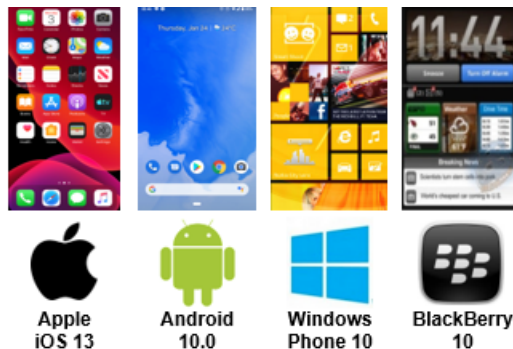
- **En 2005**, Google compra a la pequeña empresa Android Inc.
- **En 2007**, se crea el consorcio Handset Alliance con numerosos fabricantes hardware, software, proveedores de internet, etc.
- **Ese mismo año**, se lanza la primera versión del Android SDK y aparece el primer móvil con Android (T-Mobile G1). Se libera el código fuente.
- **En 2009**, versiones 1.5 y 2.0 del Android SDK con nuevas características como el teclado en pantalla.

- **En 2010**, versiones 2.1, 2.2 y 2.3 del Android SDK. Uno de los sistemas operativos móviles más utilizados, con un uso similar a iOS o mayor.
- **En 2011**, versión 3.0 (Honeycomb) para tablets y 4.0 (Ice Cream Sandwich) para móviles y tablets. Cuota de mercado del 50%.
- **En 2012**, unificación de Play Store tanto para aplicaciones como para contenidos. Versiones 4.1 y 4.2 (Jelly Bean). Cuota de mercado del 70%.

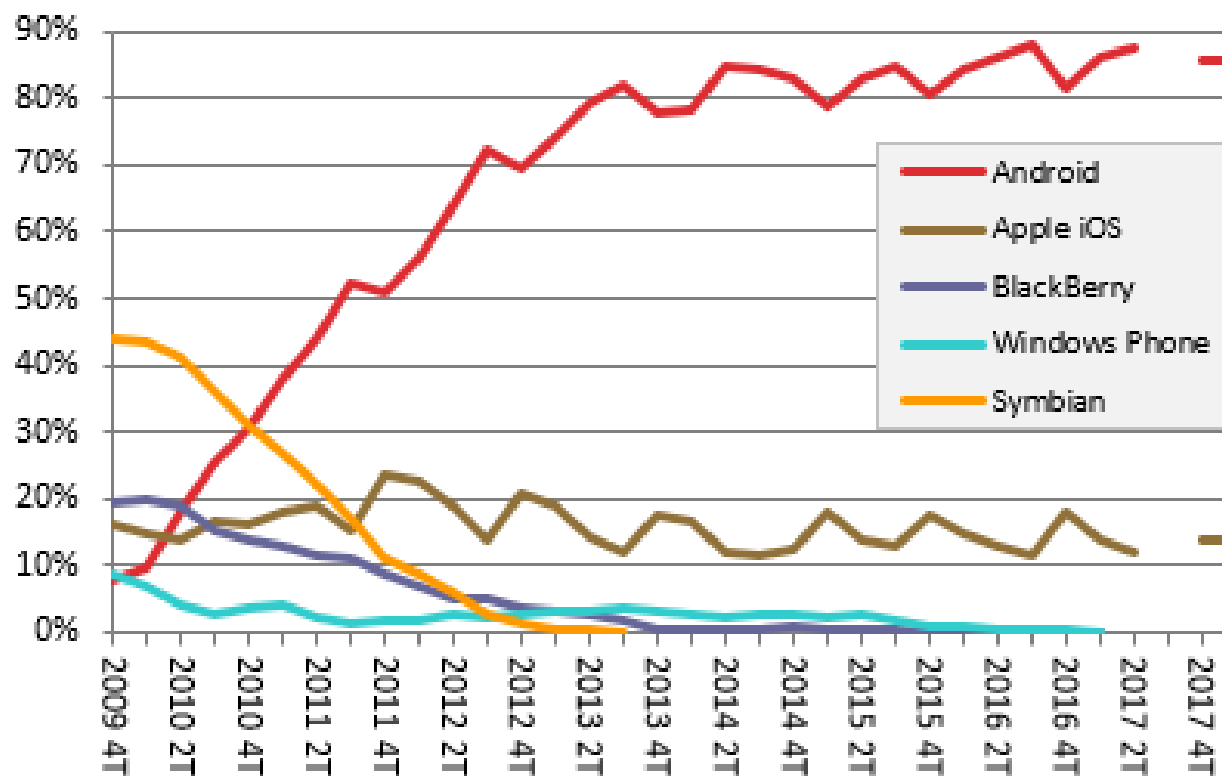


- En **2013**, versiones 4.3 y 4.4 (KitKat)
- En **2014**, versión 5.0 (Lollipop). Cuota de mercado del 85%.
- En **2015**, versión 6.0 (Marshmallow)
- En **2016**, versión 7.0 (Nougat); en 2017, versión 8.0 (Oreo)
- En **2018**, versión 9.0 (Pie); en **2019**, versión Android 10
- En **2020**, versión Android 11 del Android SDK





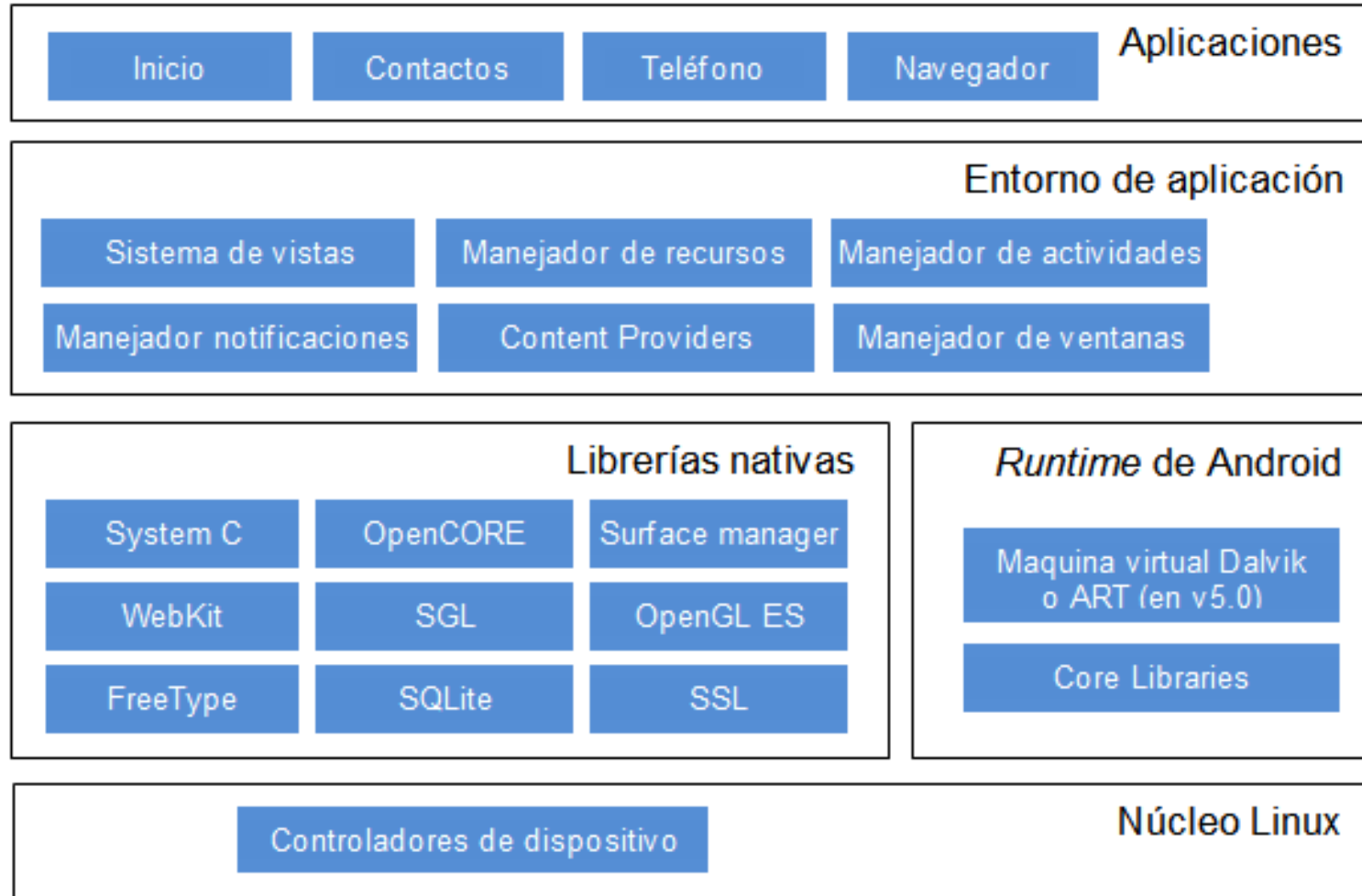
Compañía	Apple	Open Handset Alliance	Microsoft	BlackBerry
Núcleo del SO	Mac OS X	Linux	Windows NT	QNX
Licencia de software	Propietaria	Libre y abierto	Propietaria	Propietaria
Año de lanzamiento	2007	2008	2010	1999
Fabricante único	Sí	No	No	Sí
Variedad de dispositivos	Modelo único	Muy alta	Media	Baja
Soporte memoria externa	No	Sí	Sí	Sí
Motor del navegador web	WebKit	WebKit/Chromium (Blink)	Trident	WebKit
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry World
Número de aplicaciones*	2.400.000 (sept. 2016)	2.000.000 (jun. 2016)	700.000 (oct. 2016)	270.000 (2016)
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	Sin coste
Otras tiendas sin supervisión	No	Si	No	Si
Familia CPU soportada	ARM	ARM, MIPS, x86	ARM	ARM
Máquina virtual	No	Dalvik / ART	.net	No
Lenguaje de programación	Objective-C, Swift	Java, C++, Kotlin	C#, Visual Basic, C++	C, C++, Java
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac
Varios usuarios	No	Si	No	No
Modo invitado	Si	Si	No	No



*Porcentaje de smartphones vendidos a nivel mundial hasta 2018 1T.
Fuente: Gartner Group*



- | | |
|--|---|
| <ul style="list-style-type: none">• Todo supervisado Apple | <ul style="list-style-type: none">• Libre y abierto |
| <ul style="list-style-type: none">• Terminal único | <ul style="list-style-type: none">• Miles de terminales |
| <ul style="list-style-type: none">• Orientado al usuario final | <ul style="list-style-type: none">• Adaptable |



- **Como todo sistema operativo**, proporciona funciones como la seguridad, la gestión de la memoria, la capacidad multiproceso, la pila de protocolos (de comunicaciones, etc.) y soporte a hardware mediante drivers.
- **Esta capa actúa como intermediaria** entre el hardware y el resto de capas. Por tanto, es dependiente del hardware.

- Basado en el concepto de **máquina virtual**.
- **Se crea una nueva máquina virtual Java (Dalvik)** orientada a uso limitado de memoria y procesador. Cada aplicación ejecuta su propio proceso Linux con su propia instancia de la máquina virtual Dalvik.
- **Se reemplaza Dalvik por ART a partir de Android 5.0.** Reducción del tiempo de ejecución de código Java hasta en un 33%.
- **Se incluyen las Core Libraries**, con librerías típicas de Java.

- Conjunto de librerías C/C++ compiladas en código nativo del procesador:
 - **System C library** para dispositivos embebidos basados en Linux
 - **Media Framework**, basada en OpenCore (códecs de reproducción y grabación audio y vídeo, imágenes, etc.)
 - **Surface Manager**, para la representación gráfica
 - **WebKit/Chromium**, navegador web integrado en la vista WebView.

- Conjunto de librerías C/C++ compiladas en código nativo del procesador:
 - **SGL** como motor de gráficos 2D
 - **OpenGL** como motor de gráficos 3D
 - **FreeType**, soporte a fuentes (tipografías) vectoriales y de mapa de bits (raster o bitmap).
 - **SQLite**, un potente y ligero motor de bases de datos relacionales
 - **SSL**, capa de conexión segura con cifrado de las comunicaciones (Secure Socket Layer).

- Proporciona un **framework de desarrollo de aplicaciones enriquecidas** con sensores, localización, servicios, notificaciones...
- Permite la **reutilización de componentes**
- Aunque Android SDK no ofrece todo lo disponible en el entorno de ejecución de Java (JRE), sí es compatible con buena parte del mismo.

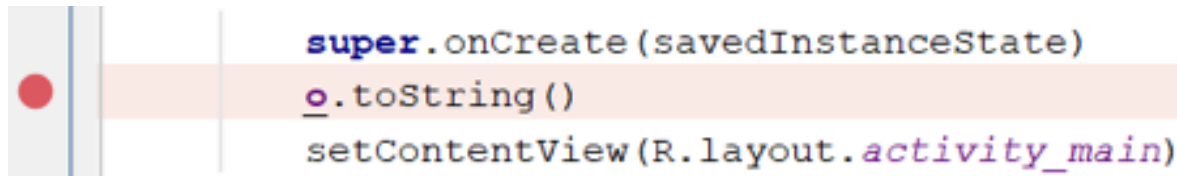
Los servicios más importantes que incluye son:

- **Views:** conjunto de vistas (parte visual de los componentes).
- **Resource Manager:** proporciona acceso a recursos que no son en código Java / Kotlin (imágenes, vídeos...)
- **Activity Manager:** gestiona el ciclo de vida de las aplicaciones y proporciona la navegación entre ellas.
- **Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

- Nivel formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la **máquina virtual Dalvik o ART** para garantizar la seguridad del sistema.
- Normalmente las aplicaciones Android están escritas en **Java / Kotlin** mediante el **Android SDK** (Software Development Kit).
- Existe la opción de desarrollar las aplicaciones utilizando **C/C++** mediante el **Android NDK** (Native Development Kit).



- Como todo IDE, Android Studio nos permite **depurar (*debugging*)** las aplicaciones de forma completa.
- Una forma es mediante la inserción de **puntos de ruptura o breakpoints** (el breakpoint se introduce haciendo clic en la barra de la izquierda de una línea del código fuente).



- Tras ello, ejecutamos la aplicación mediante **Run > Debug app (Mayús + F9)** o pulsamos en el icono equivalente de la barra de menú.
- La aplicación parará en el punto de ruptura. Podremos continuar, línea a línea, con **Run > Step Over (F8)**

- El sistema Android utiliza el **fichero *LogCat*** para registrar todos los problemas y eventos principales que ocurren en el sistema.
- La **clase *Log*** proporciona un mecanismo para **introducir mensajes desde nuestro código en este fichero**. Disponemos de varios métodos para generar distintos tipos de mensajes:

Log.e(): Errores

Log.w(): Warnings

Log.i(): Información

Log.d(): Depuración

Log.v(): Verbose

```
@Override
public void onCreate(Bundle savedInstanceState) {
    Log.d("HolaMundo", "Entramos en onCreate");
    super.onCreate(savedInstanceState);
    Object o = null;
    o.toString();
    setContentView(R.layout.activity_main);
}
```

- La documentación principal de Android se encuentra en:

<http://developer.android.com/>

- Se incluye la descripción de todas las clases (*Docs > Reference*), conceptos clave y otro tipo de recursos.

developers 

Platform

Android Studio

Google Play

Jetpack

Kotlin

Docs

News

Documentation

OVERVIEW

GUIDES

REFERENCE

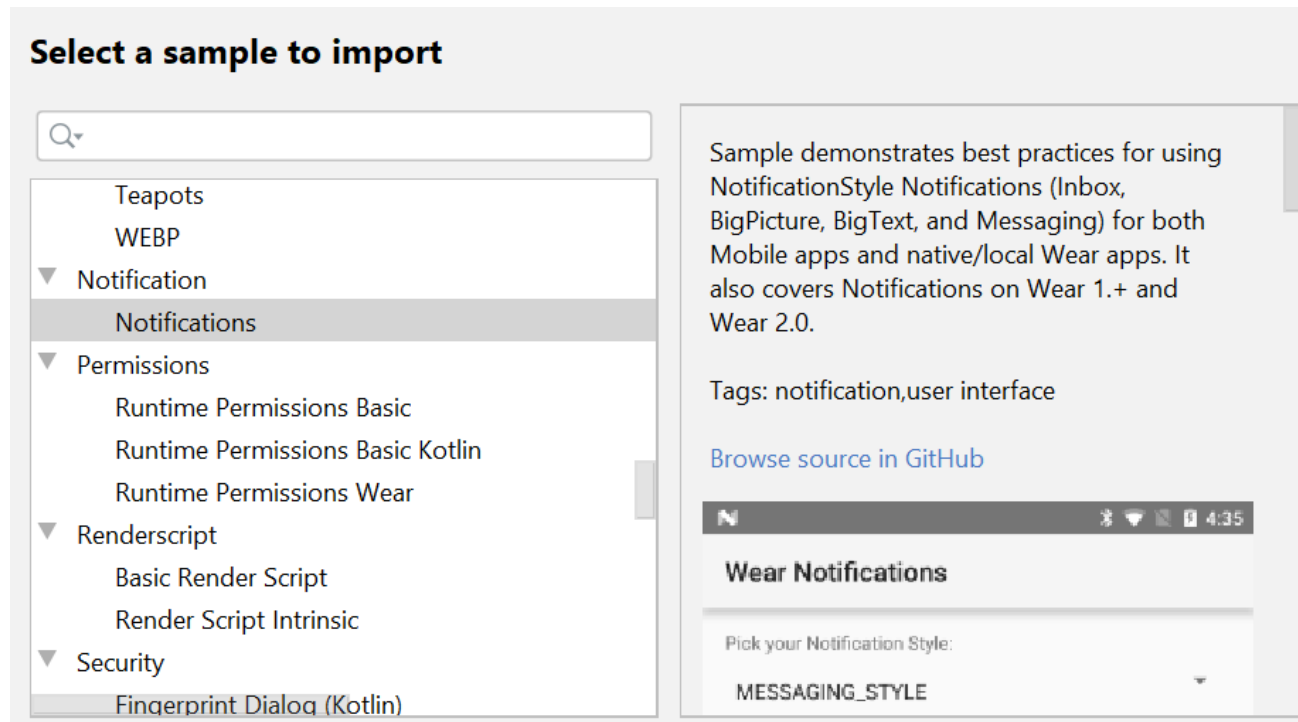
SAMPLES

DESIGN & QUALITY

- Google ha preparado un repositorio de ejemplos en GitHub que pueden ser instalados **desde Android Studio**.

- Selecciona **File > New > Import Sample...**

Aparecerá la siguiente ventana:



Los proyectos se encuentran clasificados en categorías: *Admin, Background, Connectivity, Content, Input, Media, Notification, ...* Selecciona un proyecto de alguna de estas categorías. A la derecha podrás leer una breve descripción o ver una vista previa.