



El futuro digital  
es de todos

MinTIC

Universidad  
Industrial de  
Santander



‘Misión  
TIC 2022’

# EL ECOSISTEMA ANDROID Y SU HERRAMIENTA DE DESARROLLO

# 1. Android

## 1.1. Desarrollo de aplicaciones móviles

El desarrollo de aplicaciones móviles es el conjunto de procesos y procedimientos involucrados en la escritura de software para pequeños dispositivos inalámbricos de cómputo como teléfonos inteligentes o tabletas.

El desarrollo de aplicaciones móviles es muy parecido al desarrollo de aplicaciones web, ya que ambos nacen en el desarrollo de software más tradicional. La diferencia más pronunciada; sin embargo, es que las aplicaciones (apps) móviles a menudo se escriben específicamente para aprovechar las características únicas que ofrece un dispositivo móvil en especial. Por ejemplo, una aplicación podría escribirse para que el dispositivo pueda ser aprovechado al 100% de su hardware.



Una forma de garantizar que una aplicación muestre el mejor rendimiento en un dispositivo determinado, es desarrollar la aplicación de forma nativa en ese dispositivo. Esto significa que, a un nivel muy bajo, el código está escrito específicamente para el procesador de un dispositivo específico. Sin embargo, cuando una aplicación necesita ejecutarse en varios sistemas operativos, casi no hay código (si lo hay) que se pueda reutilizar desde el desarrollo inicial. La aplicación debe reescribirse para cada dispositivo específico.

De cara al futuro, se espera que la mayor parte del trabajo de desarrollo de aplicaciones móviles, se centre en la creación de aplicaciones basadas en navegador que sean independientes del dispositivo. Las aplicaciones basadas en navegador son solo sitios web creados para navegadores móviles. Estos sitios están diseñados para cargarse rápidamente a través de redes móviles y tienen una navegación con los dedos, fácil de usar.

Más información en: [¿Qué es desarrollo de aplicaciones móviles?](#)

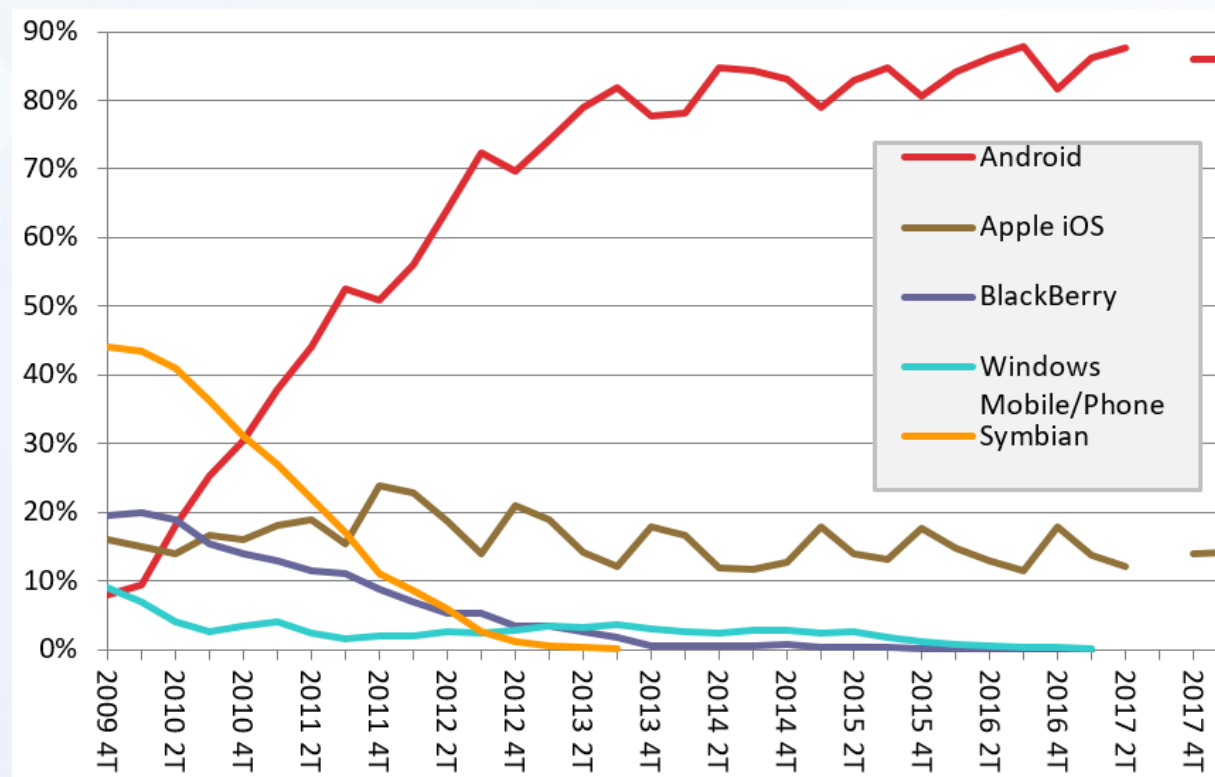
## 1.2. Sistemas operativos móviles

Un sistema operativo móvil o (SO) móvil, es un sistema que controla un dispositivo móvil al igual que los PCs que utilizan Windows o Linux, los dispositivos móviles tienen sus sistemas operativos como Android, IOS, entre otros. Los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica. A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan, adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el primer trimestre de 2016 era el siguiente:

Android	84.1 %
iOS	14.8 %
Windows Phone	0.7 %
BlackBerry OS	0.2 %
Otros	0.2 %



Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en solo dos años (2009 a comienzos de 2011) pasó a ser el SO móvil más utilizado.



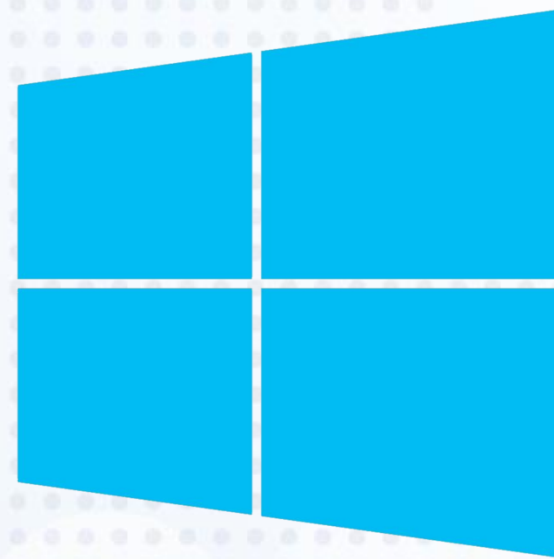
Cuota de mercado de sistemas operativos móviles en el mundo hasta el primer trimestre de 2018 (Tomado de: <http://www.androidcurso.com/index.php/98> )

Los 5 sistemas operativos para móviles más importantes dentro del mercado son:

# symbian

• **Symbian OS:** que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia como la más importante, Sony Ericsson, Samsung, Siemens, Benq, Fujitsu, Lenovo, LG, Motorola. Esta alianza le permitió en un momento dado ser uno de los pioneros y más usados, pero actualmente está perdiendo vertiginosamente cuota de usuarios aceleradamente; está por la versión 3. Está discontinuado y sin soporte.

• **Windows Phone:** anteriormente llamado Windows Mobile es un S.O. móvil compacto desarrollado por Microsoft, se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas, actualmente va por la versión 7. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente y existe una gran oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de la tienda en línea Windows Marketplace for Mobile.



• **Blackberry OS:** desarrollado por la empresa canadiense RIM (Research In Motion) para sus dispositivos. El sistema permite multitarea y tiene soporte para diferentes métodos exclusivos de RIM.





# ios

• **IOS de los iPhones:** anteriormente denominado iPhone OS creado por Apple originalmente para el iPhone, siendo después usado en el iPod Touch e iPad. Es un derivado de Mac OS X, se lanzó en el año 2007, aumentó el interés con el iPod Touch e iPad que son dispositivos con las capacidades multimedia del iPhone pero sin la capacidad de hacer llamadas telefónicas, en sí su principal revolución es una combinación casi perfecta entre hardware y software, el manejo de la pantalla multitáctil.



# android

• **Android:** es sin duda el líder del mercado móvil en S.O, está basado en Linux diseñado originalmente para dispositivos móviles como los teléfonos inteligentes pero después tuvo modificación para ser usado en tablets como es el caso del Galaxy Tab de Samsung , actualmente se encuentra en desarrollo para usarse en netbooks y PCs, el desarrollador de este S.O. es Google, fue anunciado en el 2007 y liberado en el 2008; además de la creación de la Open Handset Alliance, compuesto por 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para celulares, esto le ha ayudado mucho a Google a masificar el S.O, hasta el punto de ser usado por empresas como HTC, LG, Samsung, Motorola entre otros.

Más información en: [Sistemas operativos móviles](#)



Apple  
iOS 12



Android  
9.0



Windows  
Phone 10



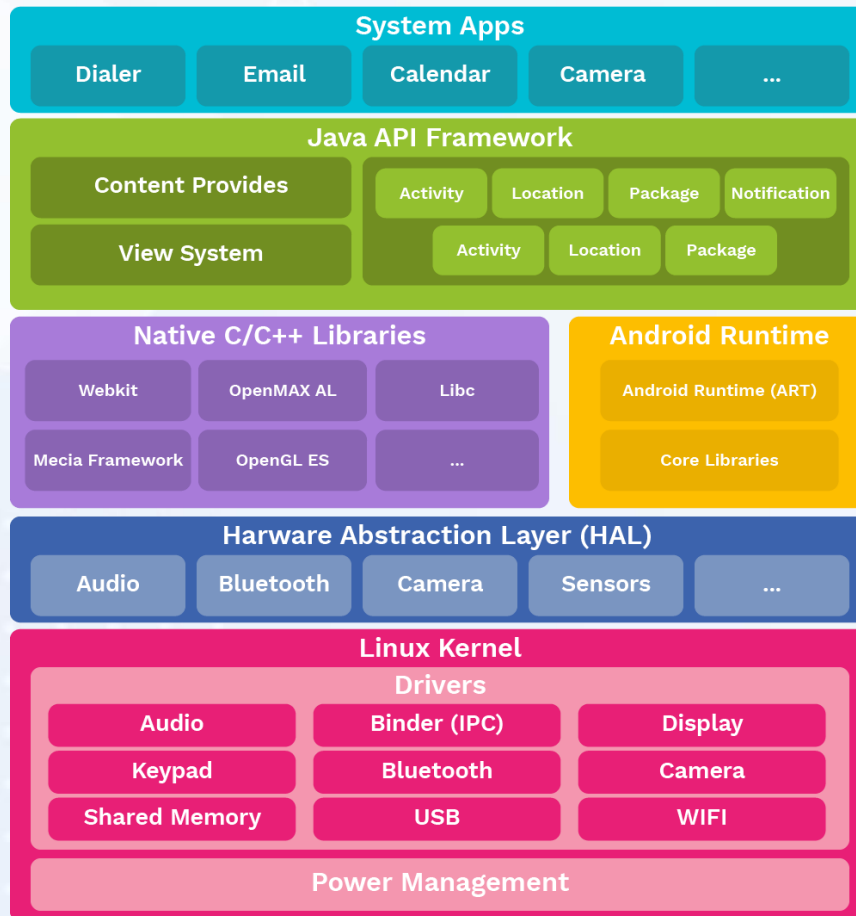
BlackBerry  
10

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	Sí	No	Sí
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	Sí	No	No
Modo invitado	Sí	Sí	No	No

Comparativa de las principales plataformas móviles

Tomado de: <http://www.androidcurso.com/index.php/98>

## 1.3. Arquitectura de Android

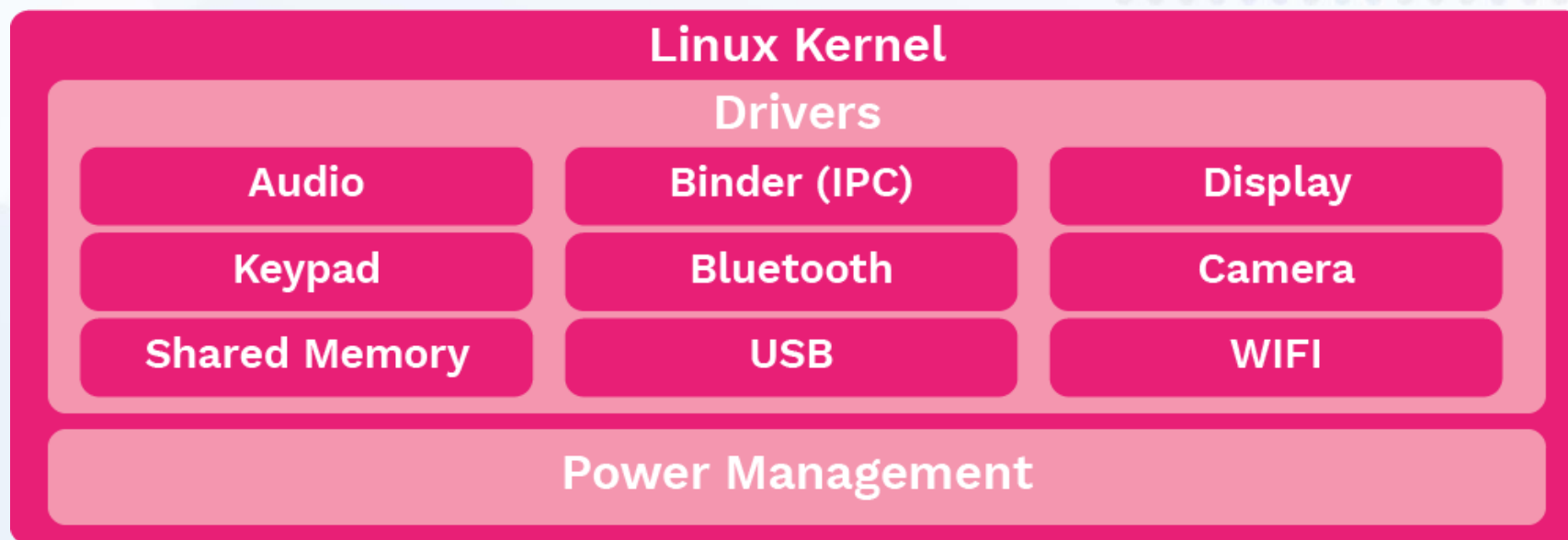


Android está construido con una arquitectura de 4 capas o niveles relacionados entre sí.

Tomada de: [Android Developer](#)

# Kernel de Linux

Android está construido sobre el núcleo de Linux en la versión 2.6, pero se ha modificado para adaptarse a dispositivos móviles.

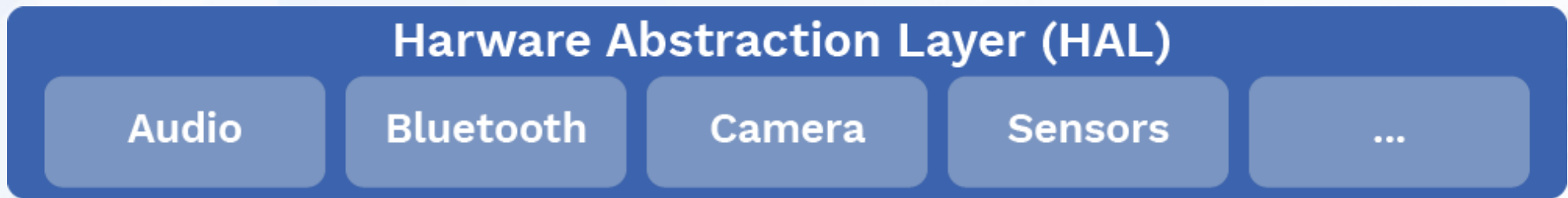


Tomada de: [Android Developer](#)



## Capa de abstracción de hardware (HAL)

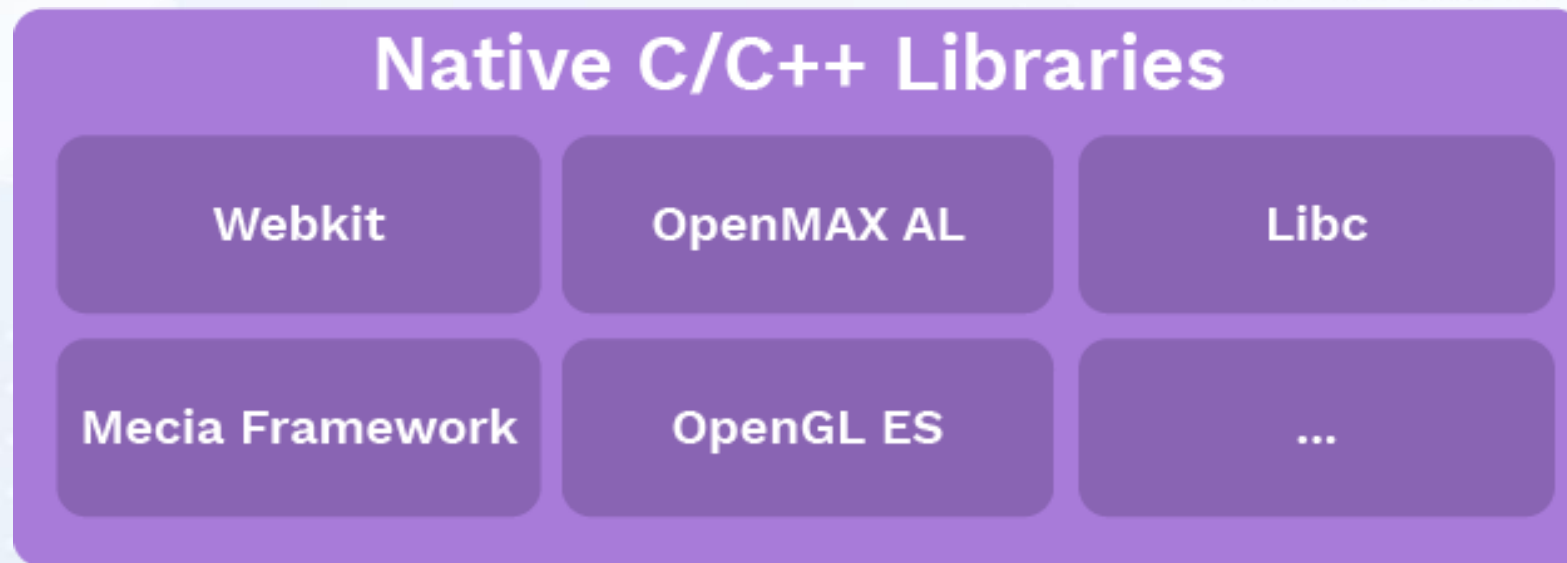
Este componente es aquel que permite la independencia del hardware. Quiere decir que Android está construido para ejecutarse en cualquier dispositivo móvil sin importar su arquitectura física.



Tomada de: [Android Developer](#)

## Librerías nativas

Aquí encontramos interfaces de código abierto como OpenGL para el renderizado de gráficos 3D, SQLite para la gestión de bases de datos, WebKit para el renderizado de los browsers, etc. También librerías para soportar los servicios del sistema como Wifi, GPS, telefonía, y muchos más.



Tomada de: [Android Developer](#)

# Android Runtime

Es un entorno de ejecución de aplicaciones utilizado por el sistema operativo móvil Android. ART reemplaza a Dalvik, que es la máquina virtual utilizada originalmente por Android, y lleva a cabo la transformación de la aplicación en instrucciones de máquina, que luego son ejecutadas por el entorno de ejecución nativo del dispositivo.

## Android Runtime

Android Runtime (ART)

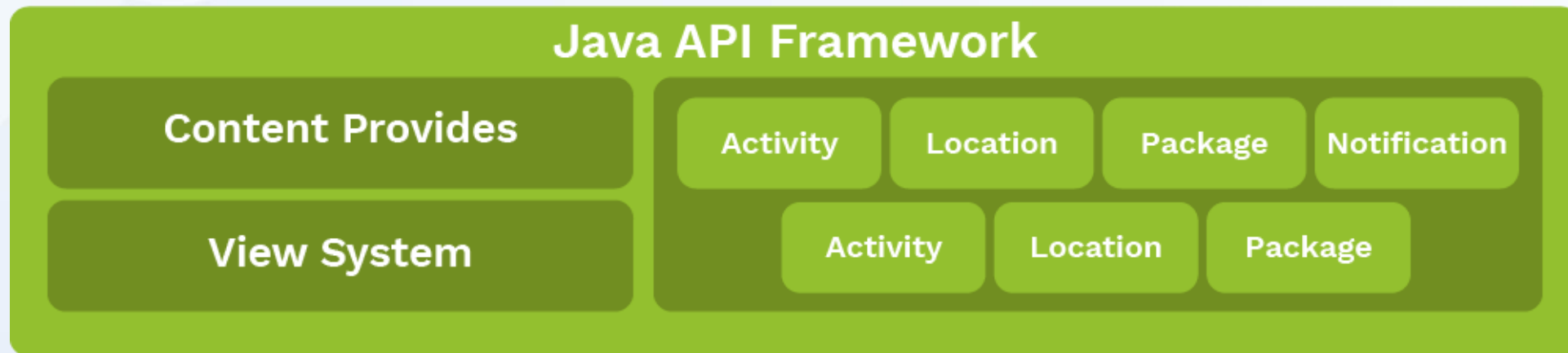
Core Libraries

Tomada de: [Android Developer](#)

# Application Framework de JAVA

Esta es la capa que nos interesa a los desarrolladores, ya que en ella encontramos todas las librerías Java que necesitamos para programar nuestras aplicaciones.

En esta capa también encontraremos manejadores, servicios y proveedores de contenido que soportan la comunicación de nuestra aplicación con el ecosistema de Android.



Tomada de: [Android Developer](#)

# Capa de Aplicaciones

Esta es la capa que permite la ejecución, comunicación y estabilidad de aplicaciones del sistema o nuestras aplicaciones.



Tomada de: [Android Developer](#)



## 1.4. El ecosistema de Android

En el 2014, Android trajo una fragmentación nueva: la fragmentación de sistemas operativos Android. Y esta fragmentación es la que ha venido de la mano de todos los nuevos productos que Google anunció: Android Wear, Android Auto, Android TV o, incluso, Android One.

Cada dispositivo cuenta con un sistema operativo que cumple las necesidades de lo que es requerido por el usuario.

•**Android Wear:** cada minuto vale con Wear OS by Google. Relojes inteligentes que te mantienen conectado con tus datos de salud, tus contactos, la información que te interesa y el Asistente de Google, todo desde la muñeca, en una interfaz versátil que permite tener un control amplio sobre todos los servicios ofrecidos.

**Android Auto:** Crea apps que ayuden a los usuarios a conectarse en la ruta mediante el SO Android Automotive y Android Auto. Los usuarios que tienen un vehículo con el SO Android Automotive integrado pueden instalar su app en el sistema de infoentretenimiento de su vehículo. Android Auto permite a los usuarios conectar su teléfono (con Android 5.0 o posterior) a un vehículo compatible a fin de mostrar una versión optimizada para conductores directamente en la consola.

**Android TV:** Crea apps que permiten disfrutar de contenido envolvente en la pantalla grande. Los usuarios pueden descubrir tus recomendaciones de contenido en la pantalla de inicio y la biblioteca Leanback proporciona API que te ayudan a generar una gran experiencia de uso para un control remoto.

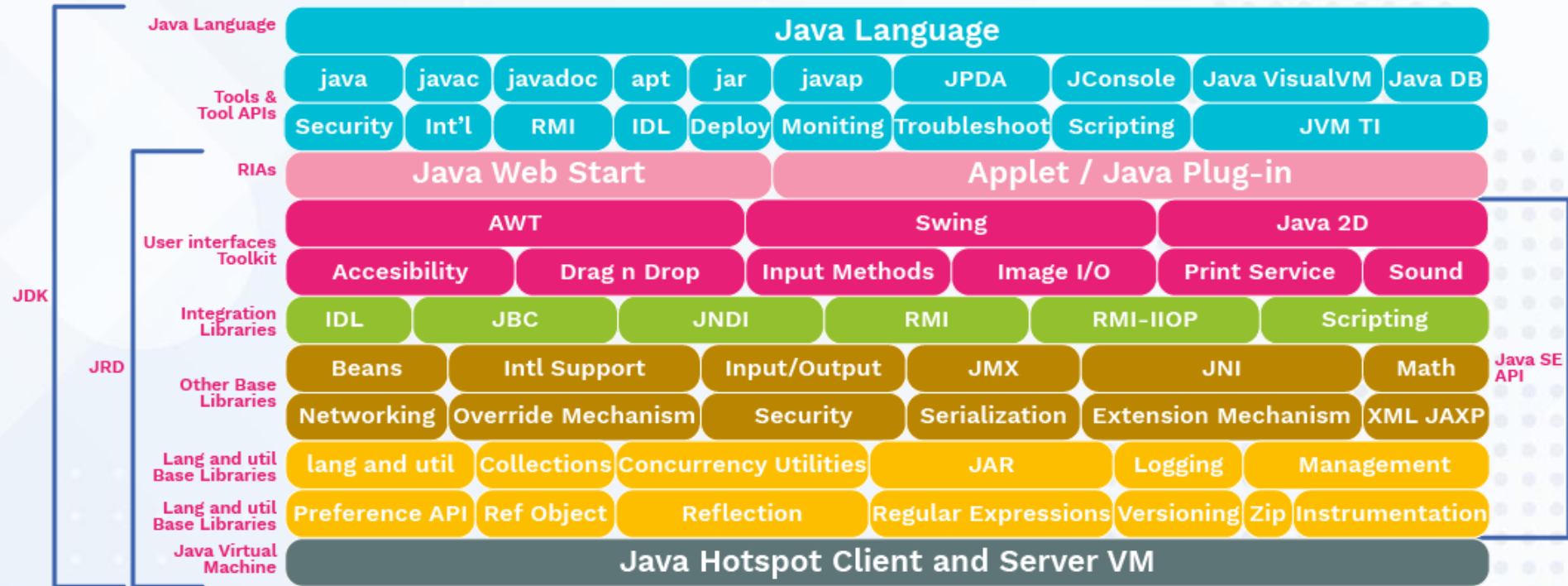
A esto se añade, la solución planteada por google con su chromecast para televisores antiguos que no cuentan con las opciones de conectarse a internet, con el fin de que no queden obsoletos, sino que puedan pertenecer a este ecosistema inteligente que hoy en día se está volviendo necesario.

## 2. Entorno de desarrollo integrado

### 2.1. Java Development Kit

**Java Development Kit (JDK)** es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

## Estructura de la plataforma JAVA



A continuación, se presenta información de utilidad. Haga clic en cada uno de los enlaces que se han dispuesto:

[Enlace de la Documentación](#)

[Enlace de descarga de JDK.](#)

[Material de Instalación de Java Development Kit \(JDK\).](#)

## 2.2. Requerimientos mínimos para Android Studio

**Android Studio** está disponible para Windows 2003, Vista, 7, 8, y 10, tanto plataformas de 32 como de 64 bits, GNU/Linux, Linux con GNOME o KDE y 2 GB de memoria RAM mínimo y macOS, desde 10.8.5 en adelante.

### Usuario del sistema

Para los sistemas operativos Windows, se recomienda que el usuario donde se realizará la instalación de Android Studio no debe contener caracteres especiales como !, @, \$, %, &, - y espacios en blanco:

- C:\Users\Maicol 2020\  
•C:\Users\José\

Se recomienda la creación o el uso de usuarios de Windows como:

- C:\Users\Jose\  
•C:\Users\Maicol\_2020\

[Leer más acerca de cómo agregar o quitar cuentas en tu PC.](#)

[Material de Requerimientos para Aplicaciones Android.](#)



## 2.3. Instalación de Android Studio

Para descargar el IDE de Android ingrese a la página de [developer de Android](#).

[Material de Instalación de Android Studio](#).

[Material de configuración de Android Studio](#)

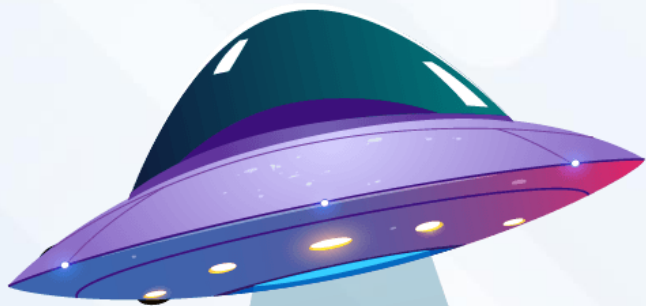


## 3. Reconocimiento del IDE de Android Studio

### 3.1. Configuración del SDK de Android

El SDK (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen GNU/Linux, Mac OS X 10.5.8 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Android Studio junto con el complemento ADT (Android Development Tools plugin). Además, los programadores pueden usar un editor de texto para escribir ficheros Java y XML y utilizar comandos en un terminal (se necesitan los paquetes JDK, Java Development Kit y Apache Ant) para crear y depurar aplicaciones, así como controlar dispositivos Android que estén conectados ( es decir, reiniciarlos, instalar aplicaciones en remoto, etc.). [Leer más acerca del SDK de Android.](#)

Una vez instalado Android Studio, se debe administrar el SDK de Android seleccionando las plataformas y las herramientas necesarias que posteriormente usaremos en el desarrollo de un proyecto Android.



## SDK Platform

Cada paquete de la plataforma Android SDK, incluye la plataforma Android y las fuentes que pertenecen a un nivel de API de forma predeterminada.

Se deben seleccionar las plataformas que se usará para depurar y también se deberá incluir la plataforma en la cual se compilara (La plataforma por defecto será la más reciente o última), las plataformas se pueden descargar posteriormente (File > Settings > Appearance & Behavior > System Settings > Android SDK), también se puede modificar el SDK en el cual se compilara la aplicación (Vea Vista > Android > Gradle Scripts > build.gradle : Module ).

## SDK Tools

Las herramientas de desarrollo del SDK, se deberán descargar las siguientes herramientas:

- Android SDK Build-Tools (Versión más reciente).
- Android SDK Platform-Tools.
- Google Play services.
- Google USB Driver.
- Android Emulator (Opcional)
- Intel x86 Emulator Accelerator (HAXM installer) (Para procesadores INTEL)
- Android Emulator Hypervisor Driver for AMD Processors (installer) (Para procesadores AMD).

[Material de Configuración del SDK de Android Studio.](#)





## 3.2. Creación y configuración de un Android Virtual Device

Un dispositivo virtual de Android (AVD) es una configuración que define las características de un teléfono o una tablet Android, o de un dispositivo Wear OS, Android TV o Automotive OS, que deseas simular en Android Emulator. El Administrador de AVD es una interfaz que puedes iniciar desde Android Studio y te permite crear y administrar los AVD. [Leer más acerca de AVD.](#)

### Acerca de los AVD

Un AVD contiene un perfil de hardware, una imagen del sistema, un área de almacenamiento y una máscara, entre otras propiedades. Te recomendamos que crees un AVD para cada imagen del sistema que tu app pueda llegar a admitir según la configuración de [<uses-sdk>](#) de tu manifiesto.

### Perfil de hardware

El perfil de hardware define las características de un dispositivo que se envía desde la fábrica. El Administrador de AVD incluye algunos perfiles de hardware previamente cargados, como los de los dispositivos Pixel, que puedes definir o personalizar cuando sea necesario.



# Imágenes del sistema

Una imagen del sistema etiquetada con las API de Google, incluye acceso a los Servicios de Google Play. Una imagen del sistema etiquetada con el logotipo de Google Play en la columna Play Store incluye la app de Google Play Store y acceso a los Servicios de Google Play.

## Requerimientos mínimos del AVD

Además de los requisitos del entorno de desarrollo, el procesador de tu computadora debe admitir una de las siguientes tecnologías de extensión de virtualización:

- Extensiones de tecnología de virtualización Intel (VT, VT-x, vmx)
- Extensiones de virtualización AMD (AMD-V, SVM)

Para el uso de AVD se sugiere tener un equipo con las siguientes características recomendadas:

- Windows versión 7 o superior( Se recomienda el uso de la versión de Windows 10).
- Procesador Intel o AMD compatible con tecnología de virtualización.
- 8GB de memoria RAM ( 1GB para el AVD).

[Leer más acerca de cómo configurar la aceleración de hardware para Android Emulator.](#)

[Leer más acerca de cómo crear un AVD.](#)

[Material de Configuración del AVD de Android Studio.](#)



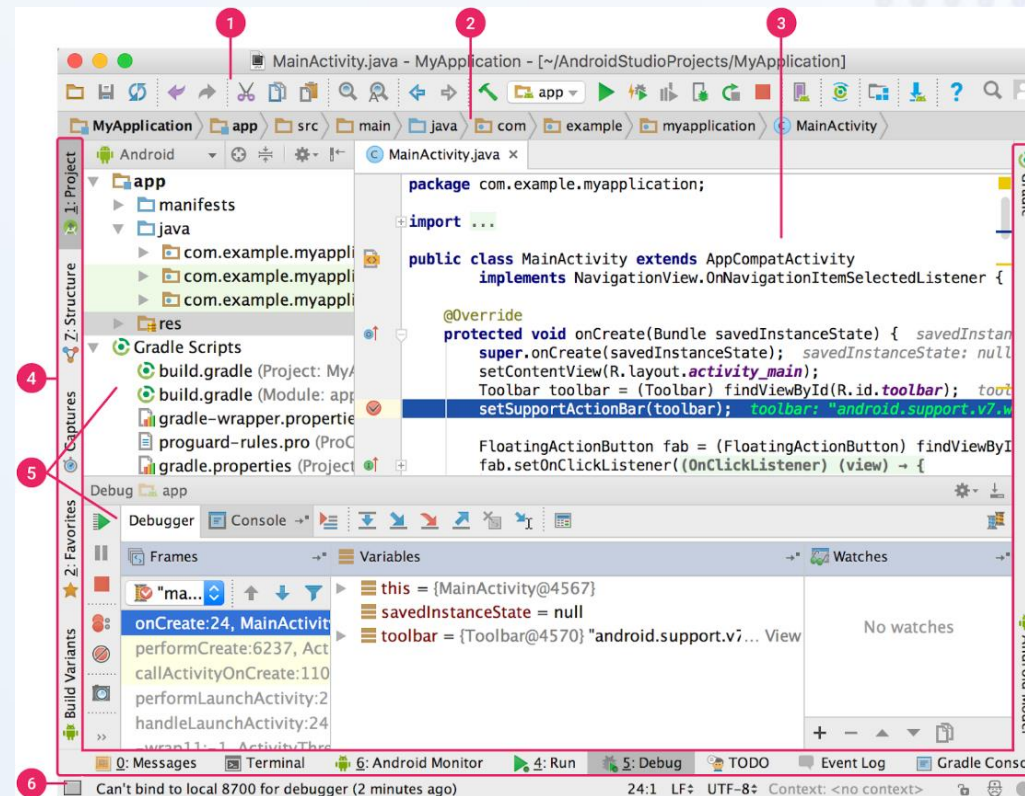
### 3.3. Conexión a un dispositivo móvil externo

Como una alternativa para depurar las aplicaciones Android se puede usar un dispositivo físico, realizando una serie de pasos para activar el modo desarrollador (Para tabletas o Smartphone) y activar la opción de depuración USB (USB debugging). [Leer más acerca de cómo habilitar la depuración USB en un dispositivo con Android.](#)

## 3.4. Reconocimiento del IDE Android Studio

La ventana principal de Android Studio consta de varias áreas lógicas.

### Interfaz de Android Studio



Tomada de Google Developers:  
[Introducción a Android Studio](#) | [Desarrolladores de Android](#)

- 1.La barra de herramientas te permite realizar una gran variedad de acciones, como ejecutar tu app e iniciar las herramientas de Android.
- 2.La barra de navegación te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana Project.
- 3.La ventana del editor es el área en la que puedes crear y modificar código. Según el tipo de actividad actual, el editor puede cambiar. Por ejemplo, cuando ves un archivo de diseño, el editor muestra el Editor de diseño.
- 4.La barra de la ventana de herramientas se encuentra afuera de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
- 5.Las ventanas de herramientas te brindan acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras. Puedes expandirlas y contraerlas.
- 6.En la barra de estado, se muestra el estado de tu proyecto y el IDE, además de advertencias o mensajes.

Se puede organizar la ventana principal para tener más espacio en pantalla ocultando o desplazando las barras y ventanas de herramientas. También puedes usar combinaciones de teclas para acceder a la mayoría de las funciones del IDE. [Leer más acerca de Android Studio.](#)



## Ventanas de herramientas

En lugar de usar perspectivas predeterminadas, Android Studio sigue tu contexto y te ofrece automáticamente ventanas de herramientas relevantes mientras trabajas. De manera predeterminada, las ventanas de herramientas más usadas se fijan a la barra de la ventana de herramientas en los bordes de la ventana de la app.

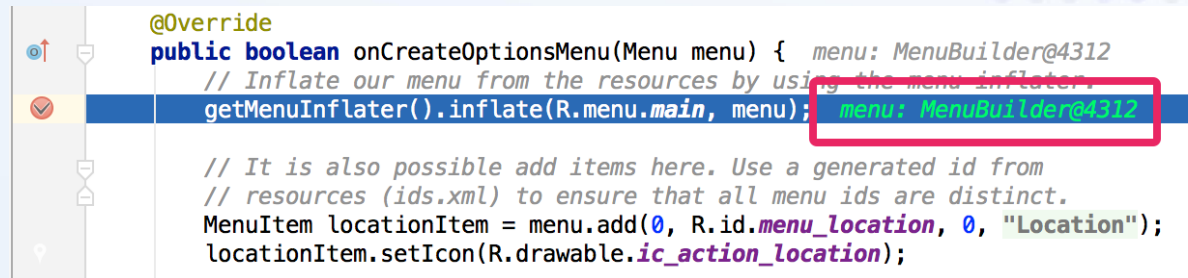
- Para expandir o contraer una ventana de herramientas, haz clic en el nombre de la herramienta, en la barra de la ventana de herramientas. También puedes arrastrar, fijar, no fijar, adjuntar y ocultar ventanas de herramientas.
- Para volver al diseño predeterminado actual de la ventana de herramientas, haz clic en **Window > Restore Default Layout** o personaliza tu diseño predeterminado haciendo clic en **Window > Store Current Layout as Default**.
- Para mostrar u ocultar la barra de la ventana de herramientas completa, haz clic en el ícono de ventana, en la esquina inferior izquierda de la ventana de Android Studio.



# Depuración integrada

Usa la depuración integrada para mejorar las revisiones de código en la vista del depurador con verificación integrada de referencias, expresiones y valores de variables. La información de depuración integrada incluye:

- Valores de variables integradas
- Referencia a objetos que hacen referencia a un objeto seleccionado
- Valores de retorno de métodos
- Expresiones lambda y de operador
- Valores de información sobre herramientas



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate our menu from the resources by using the menu inflater.
    getMenuInflater().inflate(R.menu.main, menu);
    // It is also possible add items here. Use a generated id from
    // resources (ids.xml) to ensure that all menu ids are distinct.
    MenuItem locationItem = menu.add(0, R.id.menu_location, 0, "Location");
    locationItem.setIcon(R.drawable.ic_action_location);
}
```

**Tomada de Google Developers:**  
[Introducción a Android Studio](#) | [Desarrolladores de Android](#)

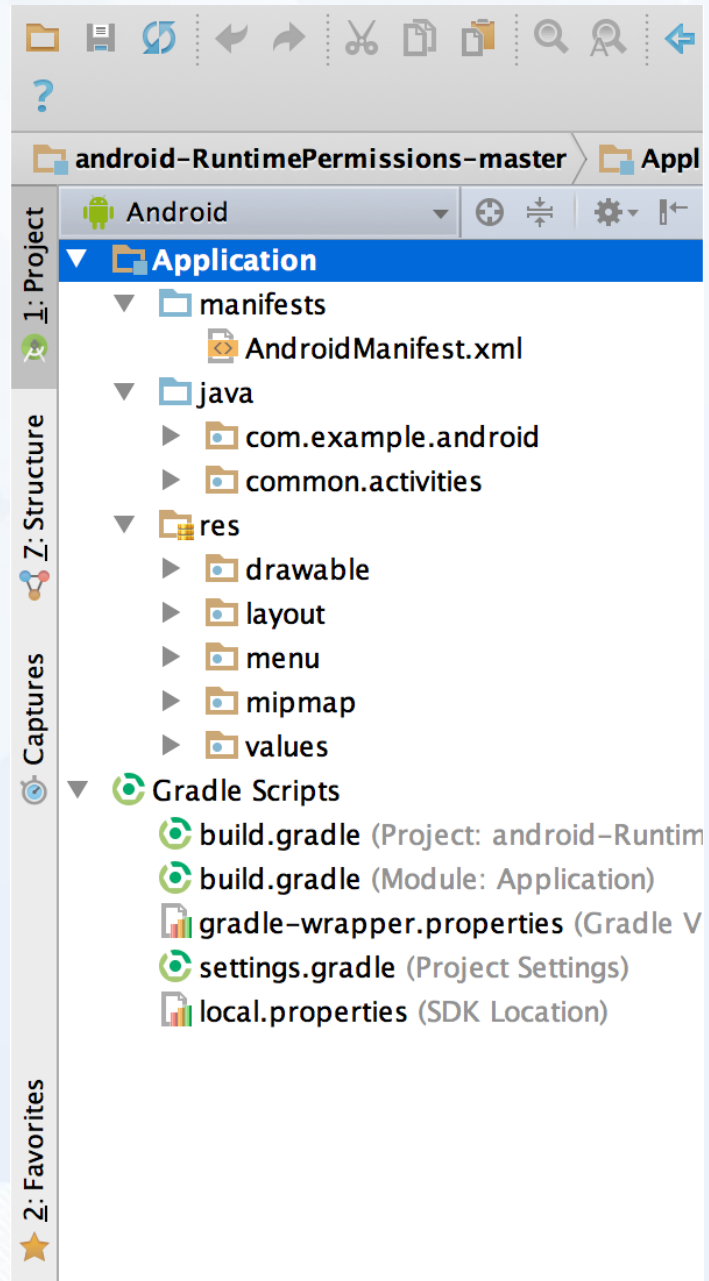
A fin de habilitar la depuración integrada, en la ventana **Debug**, haz clic en **Settings** y selecciona la casilla de verificación para activar la opción **Show Values Inline**.



## 3.5. Estructura de los proyectos en Android Studio

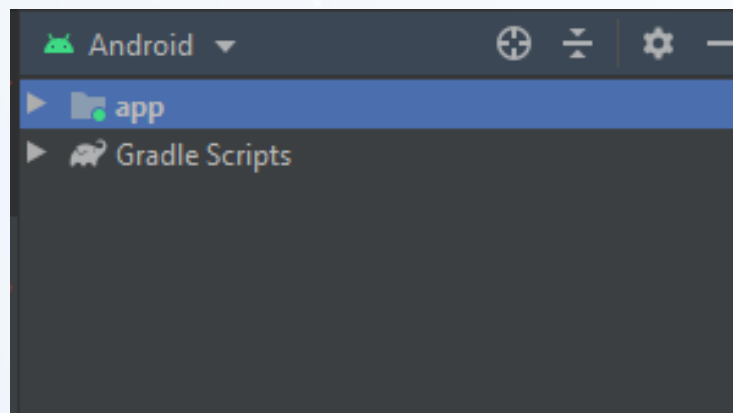
La estructura de los proyectos Android está definida por una jerarquía donde se encuentran todos los archivos de código fuente Java o Kotlin, los recursos, archivos, dependencias y los scripts del Gradle.

De manera predeterminada, Android Studio muestra los archivos de tu proyecto en la vista de proyecto de Android. Esta vista está organizada en módulos para que puedas acceder rápidamente a los archivos fuente clave de tu proyecto.

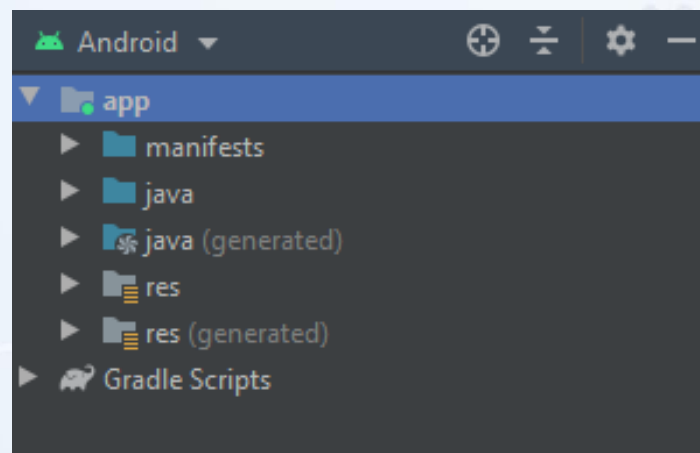


**Tomada de Google Developers:**  
[Introducción a Android Studio](#) | [Desarrolladores de Android](#)

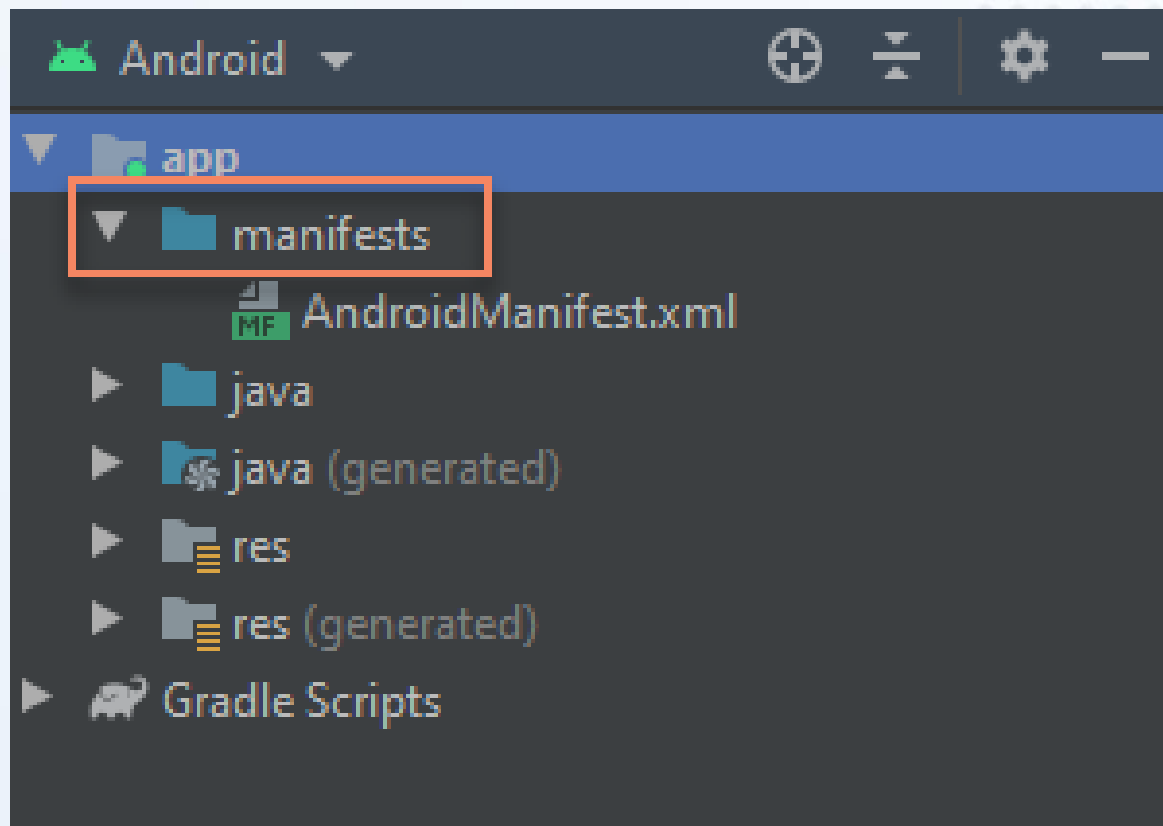
En la vista de Android podemos encontrar la carpeta app y los Gradle Scripts.



La carpeta app es la encargada de contener todos los archivos relacionados con el proyecto.

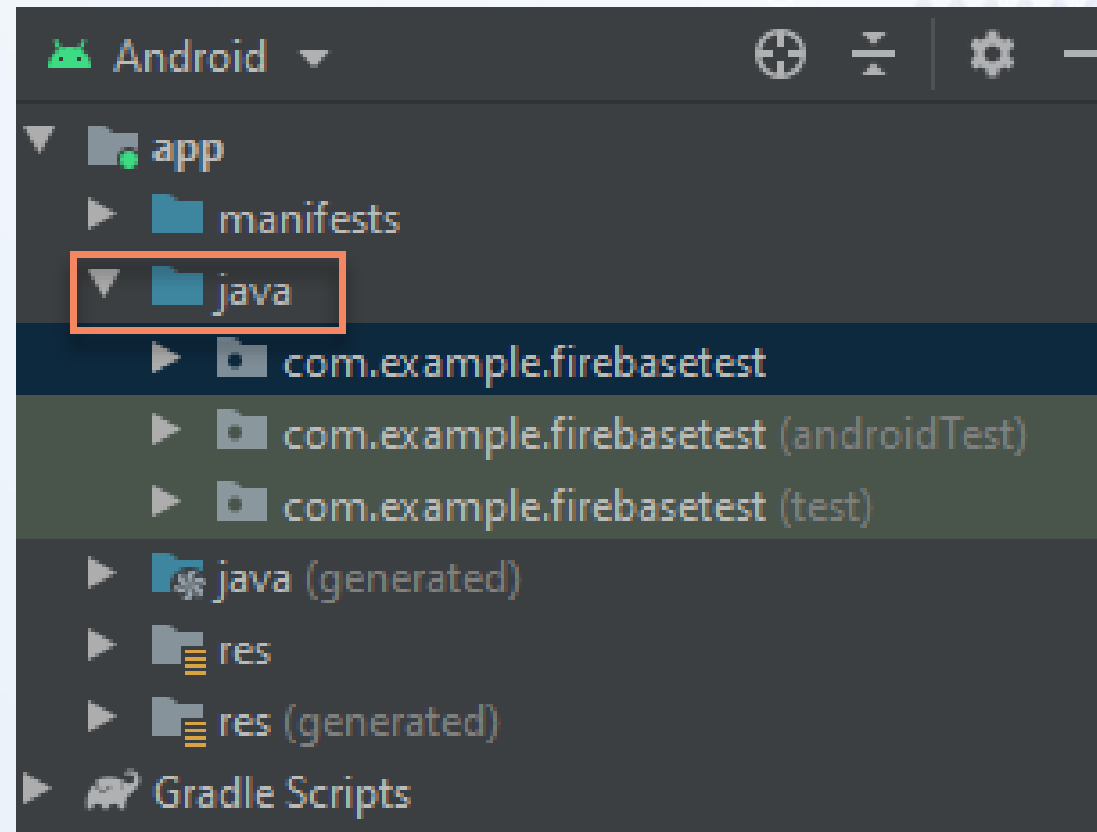


La carpeta **manifests** contiene el archivo XML que contiene todas las características de la aplicación Android. Características como icono, estilos, nombre de la aplicación, permisos, servicio y las actividades.

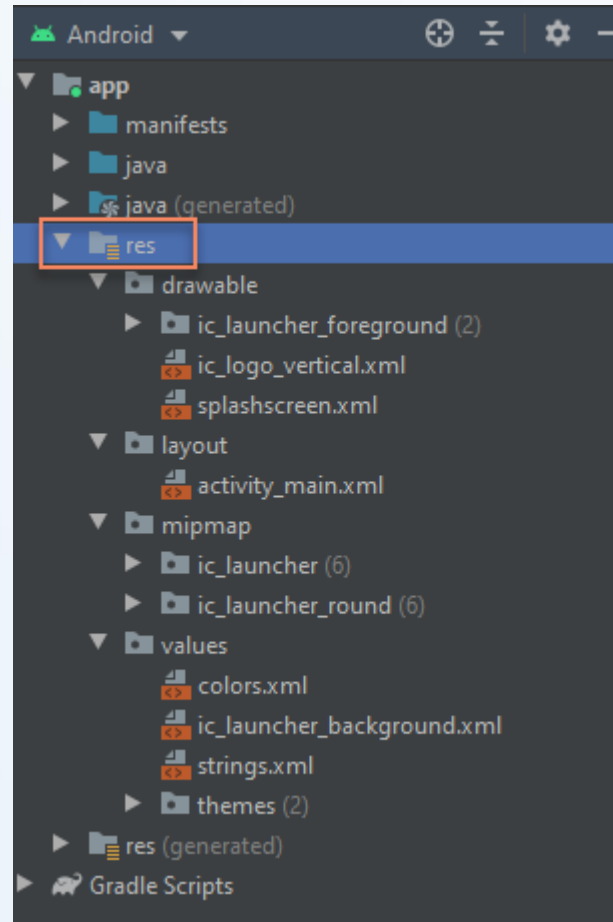




La carpeta **java** es la encargada de contener el paquete o las carpetas (bajo la estructura nombre del dominio o TLD, dominio y nombre del proyecto), donde se encuentra todo lo relacionado con nuestro código como las actividades, clases, fragmentos, interfaces y etc., también podemos encontrar las carpetas para realizar test o pruebas en nuestro proyecto.



La carpeta **res** es la encargada de contener todos los recursos de diseño como: iconos de la aplicación, imágenes, layouts o diseños de las interfaces, colores, temas, strings de IU y etc.



## 3.6. Creación de un proyecto Android

### ¿Cómo crear un proyecto?

Android Studio facilita la creación de apps para Android en varios factores, como teléfonos celulares, tablets, TVs y dispositivos para Wear.

Si no tienes un proyecto abierto, Android Studio muestra la pantalla de bienvenida, donde puedes hacer clic en Start a new Android Studio project para crear un proyecto.

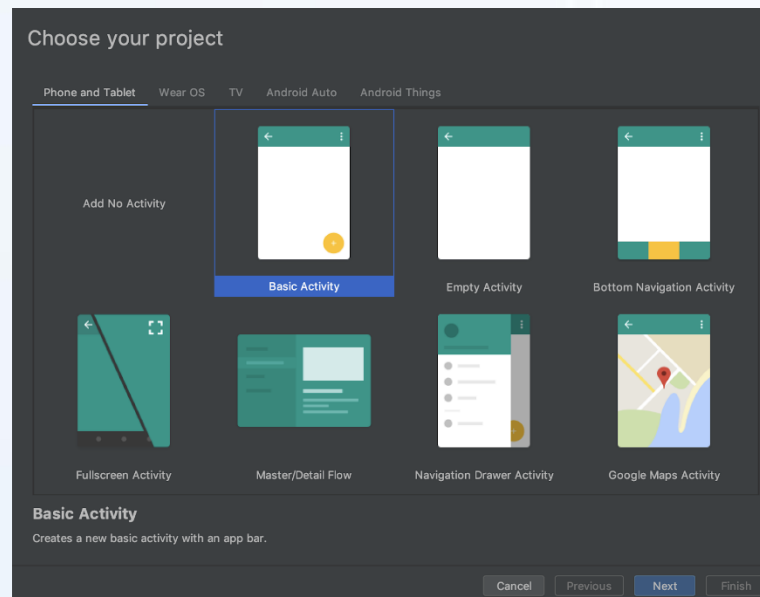
Si tienes un proyecto abierto, seleccione **File > New > New Project** en el menú principal para comenzar a crear un proyecto nuevo.

Con el asistente **Create New Project**, que te permite elegir el tipo de proyecto que deseas crear y completar con código y recursos para comenzar.



## ¿Cómo elegir tu proyecto?

En la pantalla Choose your project que aparece, puedes seleccionar el tipo de proyecto que deseas crear entre las categorías de factores de forma del dispositivo, que se muestran como pestañas cerca de la parte superior del asistente.



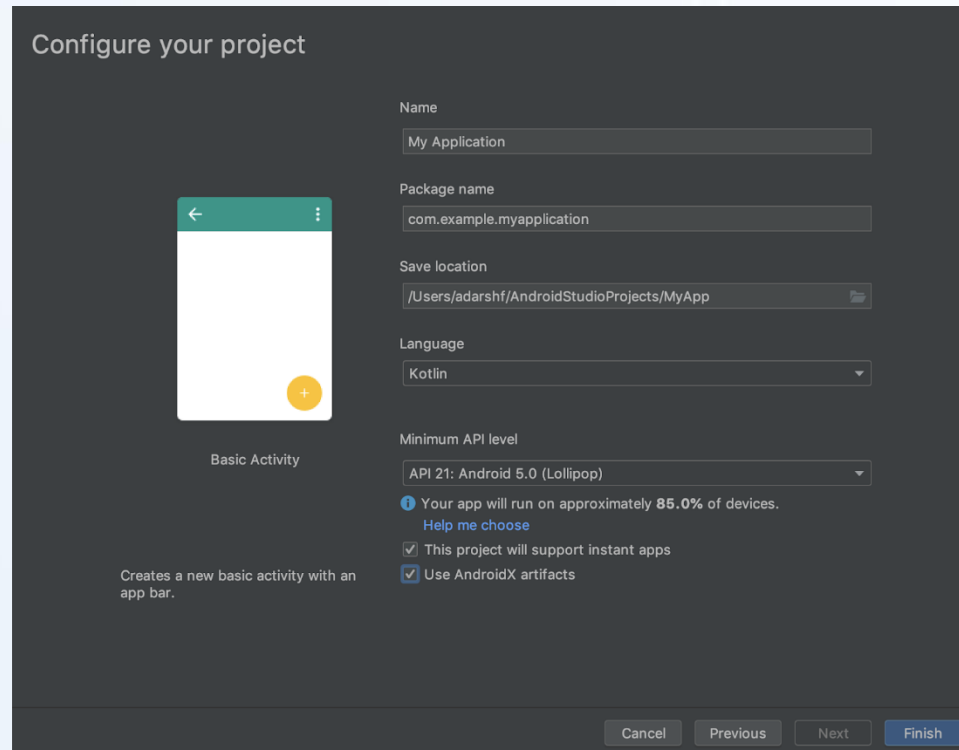
**Tomada de Google Developers:**  
**Cómo crear un proyecto | Desarrolladores de Android**

Al seleccionar el tipo de proyecto que deseas crear, Android Studio puede incluir código de muestra y recursos como ayuda para comenzar.

Después de elegir, haz clic en **Next**.

# ¿Cómo configurar tu proyecto?

El siguiente paso consiste en configurar algunos ajustes y crear un proyecto nuevo.



The screenshot shows the 'Configure your project' dialog in Android Studio. On the left, there is a preview of a 'Basic Activity' with a green header bar, a white body, and a yellow floating action button with a plus sign. Below the preview, it says 'Creates a new basic activity with an app bar.' On the right, there are several configuration fields: 'Name' (My Application), 'Package name' (com.example.myapplication), 'Save location' (/Users/adarshf/AndroidStudioProjects/MyApp), 'Language' (Kotlin), and 'Minimum API level' (API 21: Android 5.0 (Lollipop)). Below these fields, there is a warning icon and text: 'Your app will run on approximately 85.0% of devices.' with a 'Help me choose' link. There are also two checked checkboxes: 'This project will support instant apps' and 'Use AndroidX artifacts'. At the bottom, there are four buttons: 'Cancel', 'Previous', 'Next', and 'Finish'.

**Tomada de Google Developers:**  
**Cómo crear un proyecto | Desarrolladores de Android**



1.Especifica el nombre del proyecto en **Name**.

2.Especifica el nombre del paquete en **Package name**. De forma predeterminada, el nombre del paquete también se usa como ID de tu aplicación, aunque puedes cambiar este valor más adelante.

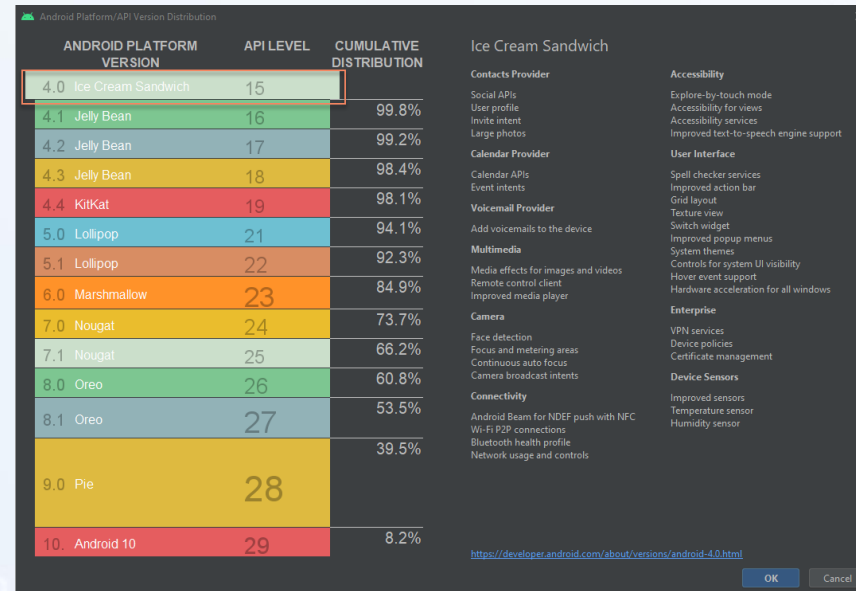
3.Especifica en **Save location** la ruta local donde vas a guardar el proyecto.

4.Selecciona el lenguaje que deseas que use Android Studio al crear código de muestra para el nuevo proyecto en Language. Ten en cuenta que no estás limitado a usar solo ese lenguaje para crear el proyecto.

5.Selecciona el nivel mínimo de API que admitirá la app en **Minimum API level**. Si seleccionas un nivel de API más bajo, es posible que la app confíe en menos API de Android modernas. Sin embargo, también habrá un porcentaje mayor de dispositivos Android que podrán ejecutar tu app. Cuando eliges un nivel de API más alto, ocurre lo contrario. Si quieres obtener información adicional que te ayude a tomar esta decisión, haz clic en **Help me choose**.

# Help me choose

En esta ventana tendremos un listado de las plataformas con su respectivo API Level junto a su porcentaje de dispositivos que se pueden alcanzar con esa versión y sus características o novedades que incluye cada versión el cual proporcionará una ayuda para poder determinar cuál será la versión mínima a trabajar.



6. Cuando esté todo listo para crear tu proyecto, haz clic en **Finish**.

# Referencias Bibliográficas

Pernía, M., 2016. *Sistemas Operativos Móviles*. [online] DTyOC. Available at: <<https://dtioc.com/2016/10/03/sistemas-operativos-moviles/>> [Accessed 20 May 2021].

Es.wikipedia.org. 2021. Android SDK - Wikipedia, la enciclopedia libre. [online] Available at: <[https://es.wikipedia.org/wiki/Android\\_SDK](https://es.wikipedia.org/wiki/Android_SDK)> [Accessed 20 May 2021].

Android Developers. 2021. Cómo crear y administrar dispositivos virtuales. [online] Available at: <<https://developer.android.com/studio/run/managing-avds?hl=es>> [Accessed 21 May 2021].

SearchDataCenter;Español. 2021. ¿Qué es Desarrollo de aplicaciones móviles? - Definición en WhatIs.com. [online] Available at: <<https://searchdatacenter.techtarget.com/es/definicion/Desarrollo-de-aplicaciones-moviles>> [Accessed 20 May 2021].

How-To Geek. 2021. How to Access Developer Options and Enable USB Debugging on Android. [online] Available at: <<https://www.howtogeek.com/129728/how-to-access-the-developer-options-menu-and-enable-usb-debugging-on-android-4.2/>> [Accessed 20 May 2021].

Es.wikipedia.org. 2021. .com - Wikipedia, la enciclopedia libre. [online] Available at: <<https://es.wikipedia.org/wiki/.com>> [Accessed 20 May 2021].

Android Developers. 2021. Introducción a Android Studio | Desarrolladores de Android. [online] Available at: <<https://developer.android.com/studio/intro?hl=es-419>> [Accessed 21 May 2021].