

# ÍNDICE

<b>Prefacio .....</b>	<b>XI</b>
<b>Capítulo 1. Introducción .....</b>	<b>1</b>
Acerca de este libro.....	2
<b>Capítulo 2. ¿Qué es Android? .....</b>	<b>3</b>
Historia de Android .....	3
Versiones de Android .....	3
Estructura del sistema operativo .....	4
Linux Kernel (Capa ROJA) .....	5
Libraries (Capa VERDE).....	5
Entorno de ejecución (Capa AMARILLA) .....	5
Framework de aplicaciones (Capa AZUL) .....	6
Aplicaciones (Capa AZUL).....	6
<b>Capítulo 3. Android Studio .....</b>	<b>7</b>
Netbeans.....	7
Eclipse .....	7
Android Studio .....	8
¿Qué se necesitará para desarrollar en Android? .....	8
Obteniendo Java .....	8
Android Studio .....	10
Configurar Android Studio para programar .....	14
Emulador.....	17
Configurar un emulador .....	19

<b>Capítulo 4. Crear una aplicación .....</b>	<b>25</b>
Crear una aplicación.....	25
<b>Capítulo 5. Estructura de un proyecto .....</b>	<b>31</b>
Perspectivas .....	32
Estructura de un proyecto.....	32
Directorio manifests.....	32
Directorio Java.....	36
Directorio res .....	38
drawable .....	39
layout .....	39
mipmap .....	40
values .....	40
Gradle.....	41
Gradle Scripts .....	41
build.gradle (Project).....	42
build.gradle (Module) .....	43
<b>Capítulo 6. Activity.....</b>	<b>45</b>
Ciclo de vida de una activity.....	47
<b>Capítulo 7. Layouts .....</b>	<b>51</b>
Tipos de layouts .....	53
LinearLayout.....	54
RelativeLayout.....	55
AbsoluteLayout .....	56
TableLayout.....	57
FrameLayout .....	58
<b>Capítulo 8. Elementos gráficos en Android .....</b>	<b>61</b>
Interfaz gráfica .....	63
Pestaña “Design” .....	64
Pestaña “Text” .....	64
TextView .....	67

Probando la orientación LinearLayout .....	68
Propiedades del TextView .....	73
Button .....	76
EditText .....	78
Practicando RelativeLayout.....	80
Margin y Padding .....	88
Pesos y distribución.....	96
<b>Capítulo 9. Densidades y tamaños de pantalla .....</b>	<b>103</b>
Densidades .....	103
Tamaños de pantalla .....	104
Relación densidad y tamaño de pantalla .....	104
<b>Capítulo 10. Medidas en Android .....</b>	<b>107</b>
Píxeles (PX).....	107
Pulgada (inch).....	108
Milímetros (mm) .....	108
Puntos (pt) .....	109
Píxeles de densidad independiente (dp) .....	109
<b>Capítulo 11. Pantallas para múltiples dispositivos .....</b>	<b>111</b>
Orientación de pantalla.....	111
Portrait .....	112
Landscape .....	115
Forzar solamente una opción de orientación.....	124
Tamaños de pantallas .....	126
<b>Capítulo 12. Archivo String .....</b>	<b>133</b>
<b>Capítulo 13. Imágenes .....</b>	<b>139</b>
Características de imágenes.....	139
ImageView .....	140
Directorios drawable.....	142
Escala de imágenes .....	145

<b>Capítulo 14. Interacción en una aplicación .....</b>	<b>149</b>
Java, Programación orientada a objetos .....	150
Clase .....	150
Objeto .....	152
Herencia .....	153
Modificadores de acceso.....	154
Acceder a elementos desde Java .....	155
Toast.....	163
<b>Capítulo 15. Agregar una Activity .....</b>	<b>169</b>
Invocar una Activity.....	173
Intent.....	176
Intent para llamadas .....	179
Permisos para llamar.....	182
Intent para enviar SMS.....	184
<b>Capítulo 16. WebView .....</b>	<b>189</b>
<b>Capítulo 17. Audio .....</b>	<b>195</b>
MediaPlayer .....	195
Directorio raw .....	196
<b>Capítulo 18. Sensores.....</b>	<b>203</b>
<b>Capítulo 19. Fragmentos .....</b>	<b>211</b>
Creando la interfaz gráfica delos fragmentos .....	213
Creando la interfaz de comunicación.....	214
Creando código de los fragmentos.....	215
Registrar los fragmentos en el manifest.....	221
<b>Capítulo 20. Pérdida de información .....</b>	<b>225</b>
Evitar pérdida de datos .....	232
<b>Capítulo 21. ListView .....</b>	<b>237</b>
<b>Capítulo 22. RecyclerView.....</b>	<b>251</b>
LayoutManager .....	259

Adapter .....	260
Comparando RecyclerView y ListView .....	267
<b>Capítulo 23. CardView .....</b>	<b>269</b>
<b>Capítulo 24. FloatingActionButton .....</b>	<b>279</b>
Sentido de usar el FloatingActionButton.....	288
<b>Capítulo 25. SnackBar .....</b>	<b>289</b>
Acciones en el SnackBar .....	294
Diferencias entre SnackBar y Toast.....	301
<b>Capítulo 26. Publicar una aplicación .....</b>	<b>303</b>
Generar APK.....	305
Subir APK a la Play Store .....	309
Ficha Play Store.....	311
Precio y distribución.....	315
Productos de compra en aplicaciones.....	316
Recomendaciones al publicar una aplicación.....	316
<b>Índice Analítico .....</b>	<b>319</b>

# PREFACIO

Desde los inicios de las computadoras se ha tenido la necesidad de interactuar con estos dispositivos tecnológicos, con el paso de los años se logra una mayor comunicación, al grado que actualmente se pueden utilizar gestos, voz, movimientos y casi cualquier acción que nuestra imaginación nos permita pensar, todo esto con el fin de lograr una respuesta o ejecutar una acción en los dispositivos que conocemos como computadoras.

Una de las revoluciones tecnológicas más importantes en los últimos años es la de los dispositivos móviles, ya que han logrado comunicar a una persona con otra en el otro lado del mundo en un instante, esto produce una sensación de “tiempo real” ya que solo basta con el acceso a una línea telefónica, plan de datos o cualquier otro medio compatible con esta tecnología hoy en día.

En la actualidad el sistema operativo más usado dentro de los dispositivos móviles es sin duda Android. Este sistema operativo acapara según la mayoría de las estadísticas el 85%-90% de la cuota del mercado. ¿Te imaginas tener acceso a esos usuarios?

Android es un sistema operativo móvil que utiliza por detrás a un viejo conocido de la industria del software, el lenguaje de programación Java, aunque Android cuenta con sus propias clases, librerías, framework y otras herramientas. Algo recomendable para comenzar en el mundo del desarrollo móvil es informarse y estudiar Java, ya que este lenguaje cuenta con algunas reglas propias en comparación con otros lenguajes de programación.

El último gran avance en el mundo del desarrollo de aplicaciones Android es la herramienta “*Android Studio*”. Fue desarrollada por la misma compañía que está detrás de Android, Google.

Una forma de apoyar y volver más ágil el desarrollo de aplicaciones es utilizar *Android Studio*, esta herramienta vino a ser un punto de apoyo bastante sólido durante el desarrollo de aplicaciones, ya que cubrió algunas de las carencias que se

tenían con las herramientas de programación y, además, cambió ciertas reglas del desarrollo en Android.

## Acerca del autor

José Dimas Luján Castillo nació en 1986 en la ciudad de México, en referencia a sus estudios el último grado que obtuvo es una Maestría en Tecnologías de Información en el ITESM.

En su experiencia como desarrollador tiene ya más de 200 aplicaciones realizadas para clientes, plataformas en línea, cursos y ejemplos en internet. Cuenta con la impartición de clases en más de 18 universidades a nivel presencial en Latinoamérica en los niveles de Licenciatura y Maestría.

Actualmente también es colaborador de las plataformas más importantes de educación en línea: Video2brain/ LinkedIn(Microsoft) y Escuela.it, codigofacilito. Cuenta con más de 170 cursos en línea en los últimos 5 años.

El autor proporciona su sitio web, redes sociales, twitter y correo a continuación:

Sitio web: [www.josedlujan.com](http://www.josedlujan.com)

Facebook: [www.facebook.com/josedlujan](http://www.facebook.com/josedlujan)

Twitter: [www.twitter.com/josedlujan](http://www.twitter.com/josedlujan)

Correo: [josedlujan@gmail.com](mailto:josedlujan@gmail.com)

El código de los ejercicios estará colgado en el repositorio del autor en GitHub: <https://github.com/josedlujan>.

## Agradecimientos

Agradezco a mi pareja Noemí por el camino que estamos iniciando y por su apoyo incondicional en estos últimos años, además de apoyarme con la revisión en los textos de este libro. También a mis padres, Fabiola y José, que espero que sientan que el camino que iniciamos juntos hace 31 años ha valido mucho la pena, les agradezco su apoyo para siempre.

# 1 INTRODUCCIÓN

Android es un sistema operativo, no es lenguaje de programación, esto nos debe hacer conscientes de que desarrollar una aplicación móvil no solo involucra los conceptos que nos encontramos habitualmente en la programación como: variables, funciones, ciclos, condiciones. El desarrollo móvil implica que conozcamos como funciona un sistema operativo, permisos, implementación de algunos sensores, manejo de tareas, manejo de memoria y de procesos, entre otros. Podemos decir que la complejidad es mayor si la desarrollamos en un solo lenguaje de programación.

El reto del desarrollador hoy en día es entender el sistema operativo, interactuar con el usuario, con otras aplicaciones, con servidores y con tecnología que forma parte del mundo móvil.

En el mercado vamos a encontrar muchos sistemas operativos como: iOS, Firefox OS, BlackBerry, Windows Phone, Ubuntu, entre otros. Android cuenta con la mayor cuota de mercado, pero también con la mayor cuota de fabricantes, esto añade complejidad ya que no todo el hardware interactúa de la misma forma. Además, el fabricante es el encargado de proporcionar la actualización al usuario y nos encontramos con el problema de la fragmentación que eleva la complejidad para el desarrollador de aplicación.

Los párrafos anteriores tratan de describir la situación actual de un desarrollador Android, se busca que el lector sea consciente del reto e implicaciones a las que se va a enfrentar en el mundo del desarrollo, pero como todo esfuerzo después de haber trabajado, estudiado y practicado, te darás cuenta de lo interesante y productivo que puede ser desarrollar aplicaciones para el sistema operativo número uno del mercado.



## **Acerca de este libro**

Este libro está dirigido a los desarrolladores novatos interesados en dar los primeros pasos en el mundo del desarrollo móvil Android; por tanto, no son necesarios los conocimientos en el lenguaje de programación Java, pero si se tienen conocimientos básicos de programación, y también de Java, podemos decir que va a ser mucho más fácil asimilar el contenido, y con las horas adecuadas dedicadas a su lectura y estudio resultará una herramienta de gran ayuda para crear aplicaciones Android.

Android como ya comenté en el inicio del capítulo, no es un lenguaje de programación como Java, Python y otros. Se recomienda tener paciencia y perseverancia para lograr el objetivo de desarrollar una aplicación.

El libro contiene impresiones de pantalla, y con ello se busca que el lector pueda entender con el apoyo visual lo que se está buscando con el ejemplo o ejercicio que se esté realizando.

# ¿QUÉ ES ANDROID?

Android es un sistema operativo al igual que Windows, Linux, etc. Por lo tanto, tiene el control total del dispositivo que lo contiene, así que cuando desarrollamos una aplicación, estamos desarrollando para el sistema operativo y podemos tener control de ciertos elementos que generar una experiencia más interesante y agradable para el usuario si lo hacemos de buena forma, en caso contrario podríamos estar generando una de las peores experiencias como usuario y afectando el rendimiento del dispositivo.

## Historia de Android

Android en la actualidad pertenece a una de las compañías pilares del mundo de la tecnología, Google. Se sabe que los creadores fueron una pequeña empresa con el nombre de Android Inc., que fue creada en el año 2003 y un par de años atrás Google la compró con la intención de adquirir el proyecto estrella de la compañía Android, que era un sistema operativo para dispositivos móviles.

Google continuó con el reto de desarrollar un sistema operativo para dispositivos móviles, pero no lo hizo solo, para ello fundó "OHA" (*Handset Alliance*) en el año 2007 teniendo dentro de sus filas compañías tecnológicas gigantes y con renombre como: HTC, Texas Instruments, Motorola y obviamente a Google.

Después de esta colaboración y arduo trabajo se lanza Android en el mes de septiembre del año 2008, fecha oficial de su lanzamiento.

## Versiones de Android

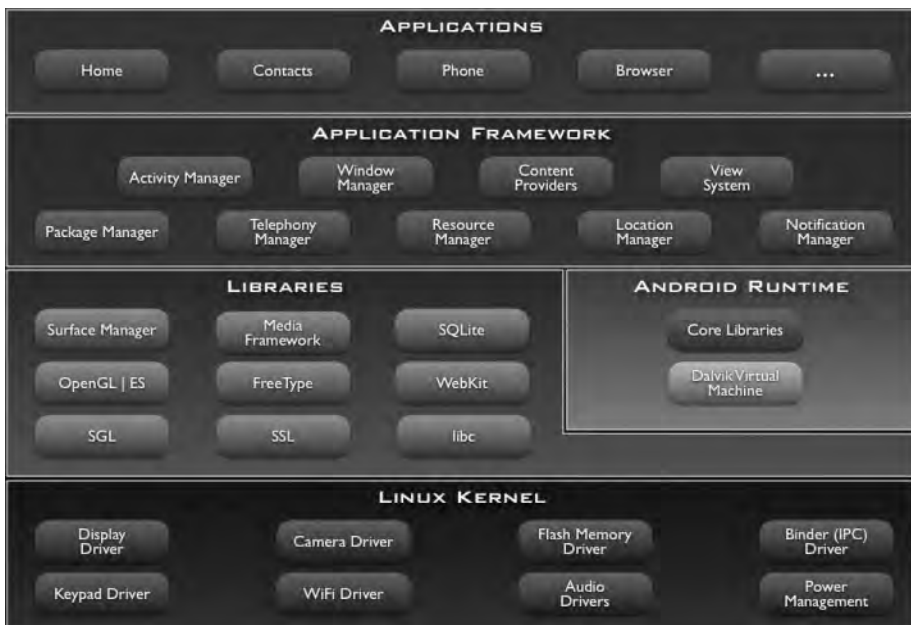
Los nombres de las versiones de Android siempre están en orden alfabético, es decir, que la primera versión se inició con la letra "A", la segunda con la "B" y así

sucesivamente; el nombre siempre hace referencia a un postre en el idioma inglés. Hasta el día de hoy estas son las versiones disponibles:

- A - Apple Pie. Versión 1.0
- B - Banana Bread. Versión V 1.1
- C - Cupcake. Versión V1.5
- D - Donut. Versión V1.6
- E - Éclair. Versión V2.0
- F - Froyo. Versión V2.2
- G - Gingerbread. Versión V2.3
- H - Honeycomb. Versión V3.0
- I - Icecream Sandwich. Versión V4.0
- J - Jelly Bean. Versión V4.1
- K - KitKat. Versión V4.4
- L - Lollipop. Versión V5.0
- M – Marshmallow. Versión V6.0
- N – Nougat. Versión V7.0

## Estructura del sistema operativo

La siguiente imagen nos ayudará a representar las capas del sistema operativo:



## Linux Kernel (Capa ROJA)

El núcleo de Android es Linux, para ser específico hablamos del kernel de Linux versión 2.6, aunque se le agregaron partes necesarias para el hardware de los dispositivos móviles.

El desarrollador tiene muy poco acceso a esta capa, como desarrolladores notaremos la casi nula interacción, el desarrollador no se preocupa por los modelos de cámara, por ejemplo, solamente decimos y damos la orden a la cámara y esta capa nos resuelve la vida. Podemos decir que esta capa contiene los drivers o las instrucciones para la interacción con el hardware de los dispositivos.

## Libraries (Capa VERDE)

Estas librerías están en los lenguajes C y C++, en su mayoría los fabricantes de los dispositivos se hacen cargo de estas capas, cada una de las librerías tiene sus funciones y las describimos a continuación:

- Surface Manager: gestiona el acceso a la pantalla.
- Media Framework imágenes, audio y vídeo.
- SQLite.
- WebKit.
- SGL.
- OpenGL.
- Freetype vectores o imágenes.

## Entorno de ejecución (Capa AMARILLA)

No lo podemos considerar una capa, ya que internamente tiene librerías. Las librerías están divididas en dos: las de Java y las de Android.

En esta capa se encuentra una de las partes más conocidas de Android, la “Máquina Virtual”, durante mucho tiempo la máquina virtual fue Dalvik, pero en las últimas versiones de Android se cambió por ARM. Aunque en la imagen aparece Dalvik en los dispositivos actualizados debes de encontrar ARM como máquina virtual, esto para el usuario final pasa inadvertido.

La máquina virtual de Java es diferente a la máquina virtual de Android, por lo tanto los bytecode no son compatibles.

## Framework de aplicaciones (Capa AZUL)

El desarrollador puede acceder a la mayor parte de esta capa sin problemas, aquí es donde encontramos la mayoría de las librerías que se usan en el día a día del desarrollador Android; describiremos brevemente cada una de las partes:

- Activity Manager: administra las actividades de nuestra aplicación y el ciclo de vida.
- Windows Manager: administra lo que se muestra en la pantalla.
- Content Provider: dependiendo de cómo le indiquemos algunos contenidos, puede ser que necesitemos encapsular para enviar o compartir información.
- View: las vistas de elementos que son parte de la interfaz gráfica, como los mapas, cuadros de texto, etc.
- Notification Manager: administra las notificaciones.
- Package Manager: administra los paquetes y nos permite el uso de archivos en otros paquetes.
- Telephony Manager.
- Resource Manager: gestiona sonidos, imágenes.
- Location Manager: gestiona la posición geográfica.
- Sensor Manager: gestiona los sensores que tenemos en el dispositivo.
- Multimedia: gestiona lo referente a vídeo, audio e imágenes.

## Aplicaciones (Capa AZUL)

La última capa de la imagen si vamos en el orden abajo -> arriba tiene como objetivo interactuar con el usuario ya que cuenta con las aplicaciones nativas del sistema operativo, por ejemplo, la parte de los contactos también cuenta con la administración de los escritorios, accesos directos, widgets, etc.

# ANDROID STUDIO 3

Hagamos un breve repaso sobre la historia de las herramientas del desarrollo en Android.

Durante mucho tiempo no existió una herramienta exclusiva para el desarrollo Android como lo tiene iOS (sistema operativo del iPhone), se usaban herramientas que en base a algunos módulos descargables se adaptaban para Android; algunas de esas herramientas eran Netbeans y Eclipse. Motorola lanzó en sus inicios una herramienta también para desarrollar aplicaciones, pero no tuvo demasiada fuerza en comparación con las primeras mencionadas.

## **Netbeans**

Netbeans era de las herramientas disponibles la que más compleja configuración presentaba, pero cuenta con una comunidad Java que la prefiere por encima de Eclipse, así que la mayoría de los desarrolladores Java con experiencia optaron por este, aunque con el paso de los años Eclipse fue pisando terreno hasta lograr ser la herramienta preferida de los desarrolladores Android.

## **Eclipse**

Eclipse es uno de los IDE más sofisticados que vamos a encontrar en el mundo del desarrollo, además de su gran tamaño, complementos, configuraciones, cuenta con una de las comunidades del desarrollo ya que se puede programar en casi cualquier lenguaje si se configura este IDE de la forma correcta, en los últimos 3 años esta fue la herramienta preferida por los desarrolladores Android hasta la aparición del IDE oficial para Android.

## Android Studio

Este entorno de desarrollo fue presentado en el año 2013 en el Google I/O y pasó a tomar la batuta como la herramienta, este IDE fue desarrollado por Google, compañía propietaria de Android, así que es el IDE oficial de desarrollo para aplicaciones Android.

Está basado en IntelliJ por lo que cuenta con diferencias notables en comparación con Eclipse, esto no hace ni más difícil ni más sencillo el desarrollo de una aplicación, en realidad cambian algunas cosas como la compilación, uso de librerías o paquetes externos, pero el código de una aplicación en su mayoría funciona de la misma forma sin importar el IDE que se esté utilizando.

## ¿Qué se necesitará para desarrollar en Android?

Vamos a responder primero la duda: ¿Qué necesitamos para desarrollar en Android? Para poder continuar con el libro, debemos saber que necesitamos 3 partes fundamentales:

- Java - Lenguaje de programación para programar en Android
- Entorno de desarrollo - Herramienta para desarrollar aplicaciones
- Android – Librerías de Android

## Obteniendo Java

Java es el lenguaje de programación utilizado para desarrollar una aplicación Android, así que lo primero que debemos obtener es Java, para eso vamos a la página de Oracle y buscamos en descargas, te dejo el enlace en el que puedes encontrar Java, pero esto puede cambiar, así que lo más recomendable es ir a un buscador y teclear “Java Download” o “Java JDK”.

Es importante mencionar que no necesitamos el Java habitual que usa el navegador u otra herramienta de tu computadora, lo que se necesita es el Java JDK, esto es porque JDK es el kit de desarrollo y contiene herramientas específicas para la programación. El enlace es el siguiente:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Saldrá una imagen como la siguiente:

Resumen **Descargas** Documentación Comunidad Tecnología Formación

## Java SE Downloads



Java Platform (JDK) 8u65 / 8u66



NetBeans with JDK 8

**Java Platform, Standard Edition**

**Java SE 8u65 / 8u66**  
 Java SE 8u65 includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u66 is a patch-set update, including all of 8u65 plus additional features (described in the release notes).  
[Learn more](#)

<ul style="list-style-type: none"> <li>• <a href="#">Installation Instructions</a></li> <li>• <a href="#">Release Notes</a></li> <li>• <a href="#">Oracle License</a></li> <li>• <a href="#">Java SE Products</a></li> <li>• <a href="#">Third Party Licenses</a></li> <li>• <a href="#">Certified System Configurations</a></li> <li>• <a href="#">Readme Files</a> <ul style="list-style-type: none"> <li>• <a href="#">JDK ReadMe</a></li> <li>• <a href="#">JRE ReadMe</a></li> </ul> </li> </ul>	<p style="text-align: center;"><b>JDK</b></p> <p style="text-align: center;"><a href="#">DOWNLOAD</a></p> <hr/> <p style="text-align: center;"><b>Server JRE</b></p> <p style="text-align: center;"><a href="#">DOWNLOAD</a></p> <hr/> <p style="text-align: center;"><b>JRE</b></p> <p style="text-align: center;"><a href="#">DOWNLOAD</a></p>
---	--

**Which Java package do I need?**

Podemos ver que tenemos dos opciones, la opción de NetBeans y la opción de Java, no olvidemos que vamos a utilizar el IDE oficial, por lo tanto, solo necesitamos Java y el paquete de la izquierda es el que va a funcionar sin ningún problema.

Al seleccionar la opción de Java vamos a ver una ventana como la siguiente:



- Overview
- Downloads**
- Documentation
- Community
- Technologies
- Training

## Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u65 Checksum  
 JDK 8u66 Checksum

### Java SE Development Kit 8u65

**You must accept the Oracle Binary Code License Agreement for Java SE to download this software.**

**Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.**

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	jdk-8u65-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.66 MB	jdk-8u65-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.67 MB	jdk-8u65-linux-i586.rpm
Linux x86	174.84 MB	jdk-8u65-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u65-linux-x64.rpm
Linux x64	172.86 MB	jdk-8u65-linux-x64.tar.gz
Mac OS X x64	227.14 MB	jdk-8u65-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.71 MB	jdk-8u65-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.01 MB	jdk-8u65-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.22 MB	jdk-8u65-solaris-x64.tar.Z
Solaris x64	96.74 MB	jdk-8u65-solaris-x64.tar.gz
Windows x86	181.24 MB	jdk-8u65-windows-i586.exe
Windows x64	186.57 MB	jdk-8u65-windows-x64.exe

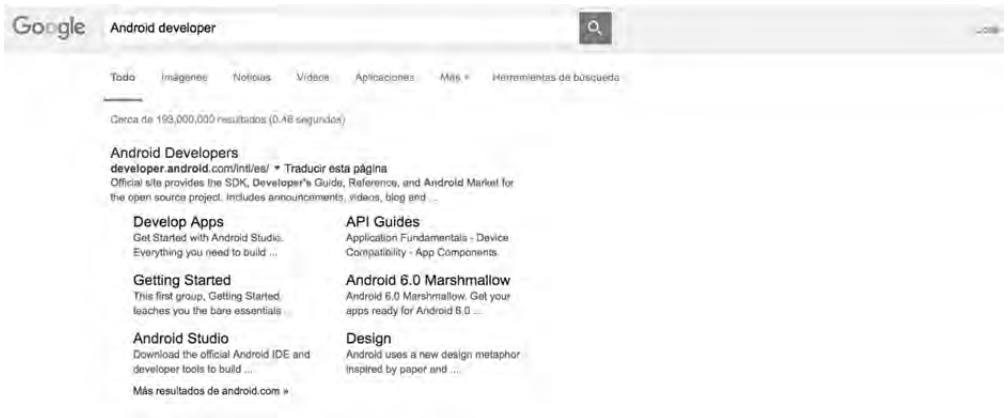
Antes de descargar una opción debemos aceptar la licencia de uso, por otro lado, debemos de seleccionar una descarga dependiendo del equipo y del procesador, por ejemplo, si tu computadora es de 64 bits y tiene Windows, tendrías que descargar el último paquete de la lista que aparece en la imagen, así que con cuidado ya que no siempre será lo mismo: depende de tu sistema operativo y de la versión de tu procesador, si es de 32 o 64 bits.

## Android Studio

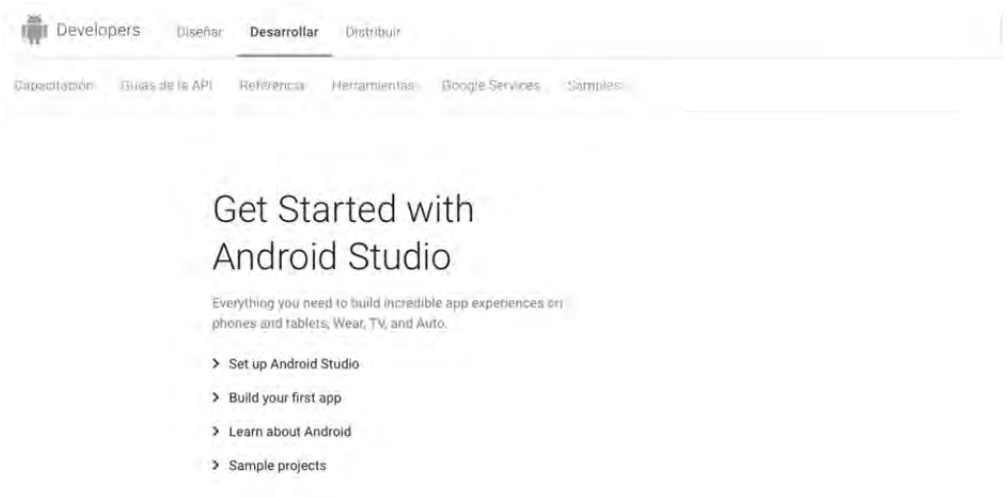
Android Studio es el IDE oficial como ya mencionamos en páginas anteriores, ahora veamos cómo obtenerlo.

Lo primero que vamos a hacer es ir al sitio oficial de desarrollo para Android:

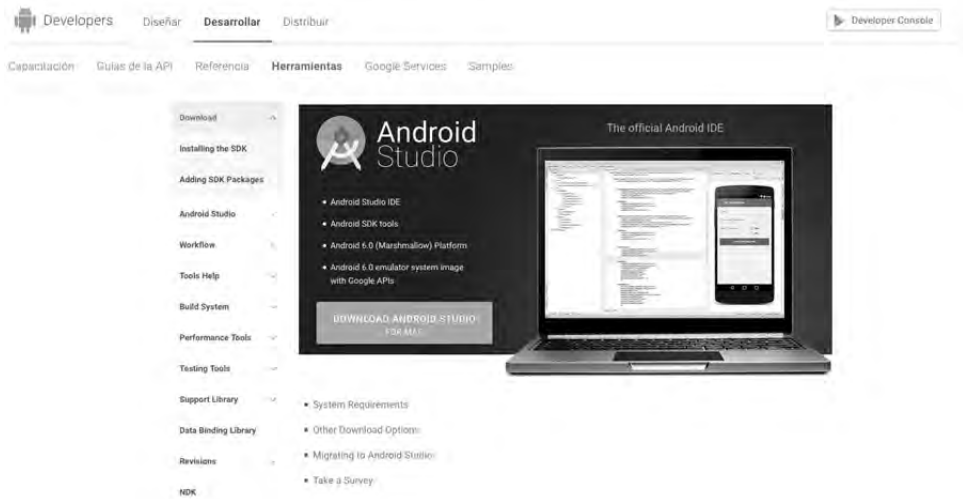
<http://developer.android.com>



Al entrar en el sitio web, seleccionamos la opción de Desarrollo para llegar a la parte de las descargas:

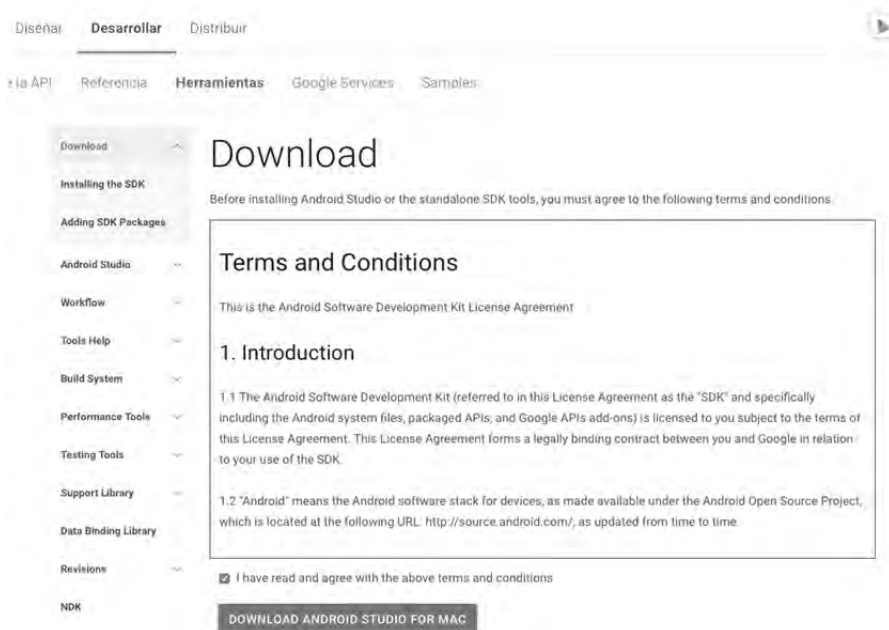


Ahora tenemos un pequeño menú con opciones en la parte central de la pantalla, vamos a la primera opción "Set up Android Studio" y vemos una ventana como la siguiente imagen:



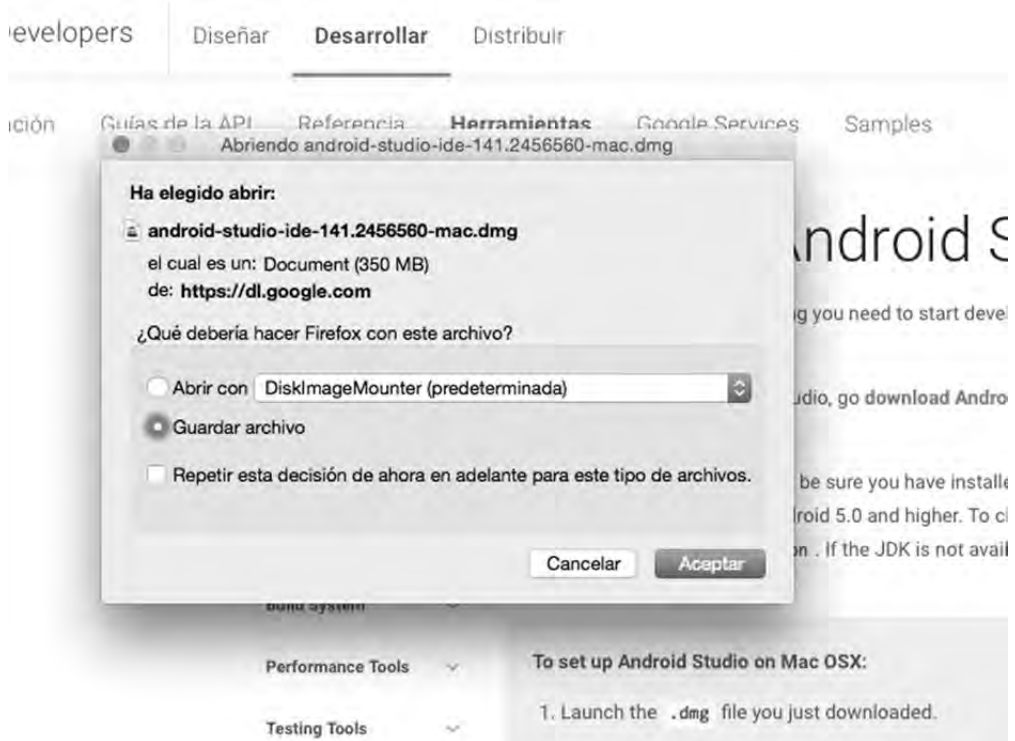
Vemos que en la pantalla nos resalta con un botón que podemos descargar el IDE Android Studio, por defecto detecta el sistema operativo y te muestra la opción adecuada, esto quiere decir que si tienes Windows te muestra la opción para este, eso mismo hace con los otros sistemas operativos para los que existe Android Studio.

Al hacer clic en el botón veremos una pantalla como la siguiente:



Lo importante de la imagen anterior es observar que si queremos descargar Android Studio primero debemos de aceptar los términos y las condiciones para poder habilitar el botón de la descarga, ya que el botón de color azul significa que tenemos el botón habilitado y funcionando.

Al hacer clic veremos una ventana de descarga como la de cualquier programa, lo importante es que en el momento de instalar Android Studio ya tengamos Java en nuestro equipo para que el paso de instalación sea sencillo, si lo hacemos de forma correcta solo tendríamos que dar doble clic al archivo y pulsar “Siguiente” en casi todas las opciones de instalación para que se instale solo; se verá como la siguiente imagen:



Al terminar la instalación tendremos Android Studio en nuestro equipo ya casi listo para programar.