**Que lenguaje escoger: ¿Java o Kotlin?**

En primer lugar, agradecerte que hayas escogido el curso “Android: Introducción a la programación”. Esperamos que toda la documentación que hemos preparado te sea de utilidad y en unas pocas semanas seas capaz de desarrollar tus propias aplicaciones para Android.  
  
Kotlin ha sido nombrado, junto con Java, lenguaje oficial para el desarrollo en Android. Todo parece indicar que Google va a mantener ambos lenguajes.

Este curso ha sido preparado para poder seguirse en cualquiera de los dos lenguajes. La decisión va a ser tuya. Si no tienes muy claro por cual decidirte, en este artículo se exponen algunos argumentos para saques tus conclusiones. Empezamos analizando las ventajas de cada uno:

**Ventajas en Java**

* Java es más sencillo y rápido de aprender que Kotlin. El requisito fundamental cando se diseñó Java era que fuera lo más simple posible.
* Java es el lenguaje de programación más popular del mundo. Aprender Java te abrirá muchas puertas en el mundo laboral.
* La mayoría de los ejemplos, o consultas en StackOverflow están en Java. Mucha de la documentación oficial de Google está disponible solo en Java.

**Ventajas de Kotlin**

* Es conciso y expresivo: vas a tener que escribir mucho menos para hacer lo mismo. Además, el código resulta mucho más fácil de entender.
* Es un lenguaje moderno: incorpora novedades como acceso directo a propiedades, lambdas, funciones de extensión, clases de datos, funciones inline, soporte nativo de delegados, y un largo etcétera.

*Para finalizar un par de consejos:*

Si eres un **programador novato**: Te recomiendo que comiences por Java. Cuando lo domines puedes plantearte pasarte a Kotlin. No te preocupes que no habrá sido tiempo perdido.  
  
Si eres un **programador experimentado o ya conoces Java**: Te recomiendo que te enfrentes a este nuevo lenguaje. Tiene una curva de aprendizaje rápida y ya verás que acaba enganchando.

**Instalación del entorno de desarrollo**

Google ha preparado el paquete de software **Android SDK**, que incorpora todas las herramientas necesarias para el desarrollo de aplicaciones en Android. En él se incluye: conversor de código, depurador, librerías, emuladores, documentación, etc. Todas estas herramientas son accesibles desde la línea de comandos.

No obstante, la mayoría de los desarrolladores prefieren utilizar un IDE (entorno de desarrollo integrado). Un IDE agrupa, en un entorno visual, un editor de código con todas las herramientas de desarrollo. Google recomienda utilizar Android Studio (basado en el IDE IntelliJ IDEA).

## Instalación de la máquina virtual Java

Las aplicaciones Android están basadas en Java, por lo que necesitas instalar un software para ejecutar código Java en tu equipo. Este software se conoce como máquina virtual Java, entorno de ejecución Java, Java Runtime Environment (JRE) o Java Virtual Machine (JVM).

Es muy posible que ya tengas instalada la máquina virtual Java en tu equipo. Si es así, puedes pasar directamente a uno de los apartados siguientes. En caso de dudas, puedes pasar también al punto siguiente. Al concluirlo te indicará si la versión de la máquina virtual Java es incorrecta. En caso necesario, regresa a este punto para instalar una que sea adecuada.

Para instalar la máquina virtual Java accede a <http://www.java.com/es/download/>, descarga e instala el fichero correspondiente a tu sistema operativo.

## Instalación de Android Studio

En la edición de Google I/O 2014 se lanzó la primera versión estable de Android Studio. Se trata de un nuevo entorno de desarrollo para Android basado en el IDE IntelliJ IDEA. Entre las novedades introducidas destacamos:

* Construcción de proyectos usando la herramienta Gradle.
* Previsualización simultánea de un layout en varios tipos de dispositivos.
* Facilidades para el testeo de código basado en JUnit.
* Integración con herramientas de gestión de versiones (como GitHub).
* Desarrollo en un mismo proyecto de diferentes versiones (como Android Wear, Android TV y Android Auto)

## Ejercicio: Instalación de Android Studio

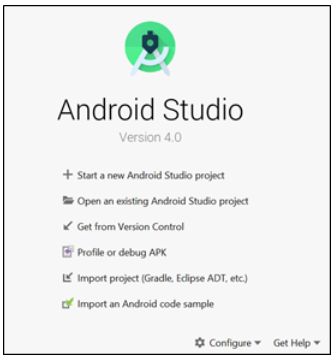
***NOTA:***Puedes encontrar una descripción más detallada de la instalación en:

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

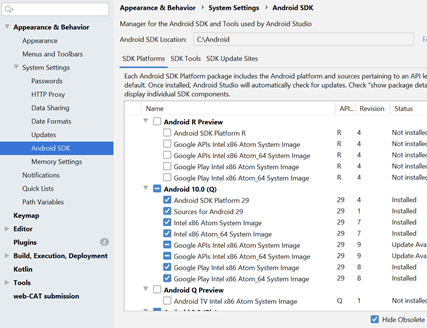
1. Descarga el paquete correspondiente a tu versión de la siguiente dirección:

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

1. Ejecuta el fichero obtenido en el paso anterior:
2. Selecciona todos los componentes a instalar y pulsa *Next*.
3. Acepta el contrato de licencia y selecciona las carpetas donde quieres instalar el IDE Android Studio y el SDK. En el resto de ventanas puedes utilizar las opciones por defecto. En la última ventana indica que quieres arrancar Android Studio.
4. Primero te preguntará si quieres importar la configuración desde una instalación anterior. Luego verificará si hay actualizaciones del SDK.
5. Tras pulsar en *Finish* pasamos a la ventana de bienvenida:



1. Comienza pulsando en *Configure*. Aparecerán varias opciones, selecciona *SDK Manager*. Esta herramienta es de gran utilidad para verificar si existen actualizaciones del SDK o nuevas versiones de la plataforma. Podrás acceder a ella desde la ventana principal de Android Studio pulsando en el botón *SDK Manager*:
2. Al entrar en el SDK Manager te muestra los paquetes instalados y los que puedes instalar o actualizar:



En la lengüeta *SDK Plataforms* se muestran los paquetes de plataforma. Pulsa en *Show Package Details* para ver los diferentes paquetes. Siempre es conveniente que tengas instalados los siguientes paquetes de la última plataforma disponible:

* *Android SDK Platform X (donde X es la última versión disponible)*
* *Sources for Android X (no es imprescindible)*
* *Google APIs … System Image (para crear emuladores con Google APIs)*
* *Google Play … System Image (para crear emuladores con Google APIs + Google Play)*

En la lengüeta *SDK Tools* se muestran paquetes con herramientas de la plataforma. Siempre es conveniente que tengas actualizados los siguientes paquetes:

* *Android SDK Build-Tools*
* *Android SDK Platform-tools*
* *Android SDK Tools*
* *Google Play services*
* *Support Repository*

**Recursos adicionales:** *Teclas de acceso rápido en Android Studio*

**Alt-Intro**: Solución rápida (Ej. añade *imports* de las clases no resueltas).

**Shift-F10**: (**Ctrl-R** en Mac): Ejecuta el proyecto.

**Shift-F9**: (**Ctrl-D** en Mac): Depura el proyecto.

**Shift-F6**: Cambia el nombre de un identificador.

**Ctrl-Alt-L**(**Option-Cmd-L**en Mac): Formatea automáticamente el código.

**Ctrl-Q**(**F1**en Mac): Muestra documentación del código.

**Ctrl-P**: Muestra parámetros del método seleccionado.

**F4**(**Cmd-fecha abajo**en Mac): Salta a declaración.

**Ctrl-Y**(**Cmd-Espacio**en Mac): Borra línea.

**Alt-Insert**(**Cmd-N**en Mac): Inserta método

**Enlaces de interés: *Conoce Android Studio***

<https://developer.android.com/studio/intro/index.html?hl=es-419>

**Preguntas de repaso:**[Instalación y entorno de desarrollo](http://mmoviles.upv.es/test_fundamentos/examen.php?s=0_2%20Instalacion%20y%20entorno%20de%20desarrollo.txt)