



Unidad 1 / Escenario 2

Lectura fundamental

Estructura de un programa en Android Studio

Contenido

- 1 Introducción a Android Studio
- 2 Estructura de un programa Android Studio

Palabras clave: AVD, Java, JDK, componente.

1. Introducción a Android Studio

Android Studio es un IDE (entorno integrado de desarrollo) para aplicaciones móviles de Android en un entorno de lenguaje Java, que permite simplificar las tareas de desarrollo. Proporciona un conjunto de herramientas para la elaboración de aplicaciones móviles en un medio de programación amigable.

Android Studio ofrece varias características, algunas de las cuales son:

- a. Contiene un editor de diseño que administra los componentes de la interfaz del usuario mediante arrastrar y soltar.
- b. Presenta un conjunto de plantillas para crear diseños y componentes frecuentes en Android.
- c. Sugiere autocompletado de sentencias e importación de paquetes.
- d. Utiliza simulador para ejecución de aplicaciones.
- e. Permite detectar errores en el momento de escribir las sentencias y presenta la posible corrección.

A continuación, se describe la instalación y configuración del IDE Android Studio para el desarrollo de aplicaciones móviles en Android.

1.1. Instalación de Java Development Kit (JDK)

Antes de poder ejecutar el entorno de desarrollo, se requiere tener instaladas las librerías de desarrollo de Java. Para esto, se debe acceder a la página de Oracle y seleccionar la versión de acuerdo con la arquitectura del equipo y del sistema operativo en donde se desea hacer la instalación.

La Figura 1 muestra las opciones para descargar el JDK, una vez se ha entrado a la página de descargas de Oracle; luego de seleccionar la correspondiente versión, se descarga e instala en el equipo. Se debe tener presente la ruta de instalación para que posteriormente sea asignada a las variables del entorno con el objetivo de que, cuando se ejecute el programa, encuentre dónde está instalado Java, como se describe continuación:

- a. En la variable del sistema, dar a la variable JAVA_HOME el valor de la ruta donde quedó instalado el JDK, es decir, C:\Program Files\Java\jdk1.8.0_171, dependiendo de la versión instalada (Figura 2).
- b. Asignar a la variable de entorno %JAVA_HOME%\bin el PATH (Figura 2).

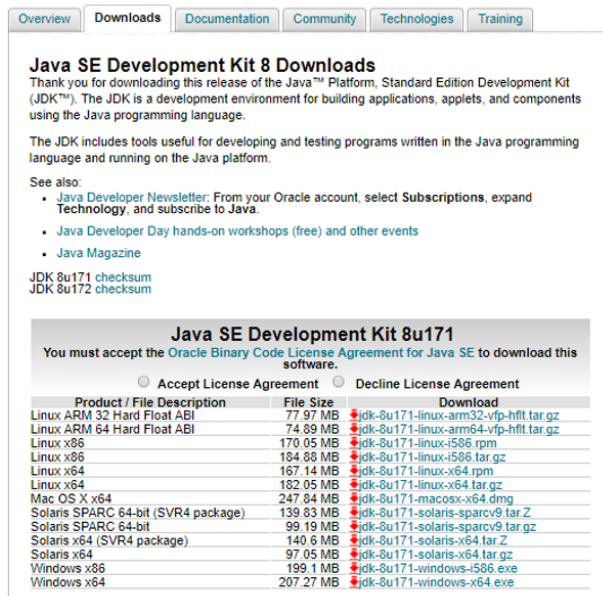


Figura 1. Captura de pantalla de Opciones de descarga JDK en su uso bajo licencia

Fuente: elaboración propia

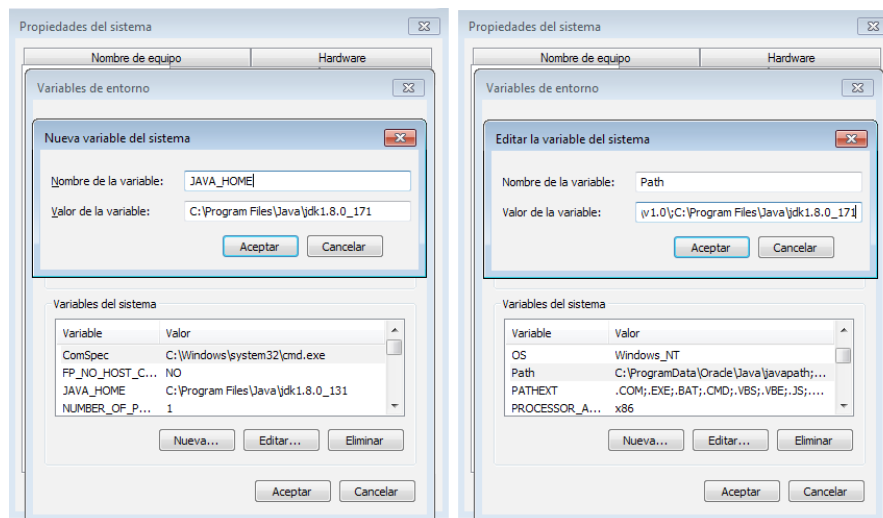


Figura 2. Captura de pantalla de variable de entorno Windows en su uso bajo licencia

Fuente: elaboración propia

1.2. Instalación Android Studio

Inicialmente, se descarga el programa de Android Developers, en donde se selecciona el paquete dependiendo de la arquitectura y la plataforma del equipo a instalar (Figura 3); después, se ejecuta y se siguen los pasos del asistente.

Android Studio downloads

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-ide-173.4720617-windows.exe Recommended	758 MB	e2695b73300ec398325cc5f242c6ecfd6e84db190b7d48e6e78a8b0115d49b0d
	android-studio-ide-173.4720617-windows.zip No .exe installer	854 MB	e8903b443dd73ec120c5a967b2c7d9db82d8ffb4735a39d3b979d22c61e882ad
Windows (32-bit)	android-studio-ide-173.4720617-windows32.zip No .exe installer	854 MB	c238f54f795db03f9d4a4077464bd9303113504327d5878b27c9e965676c6473
Mac	android-studio-ide-173.4720617-mac.dmg	848 MB	4665cb18c838a3695a417cebc7751cbe658a297a9d6c01cbd9e9a1979b8b167e
Linux	android-studio-ide-173.4720617-linux.zip	853 MB	13f290279790df570bb6592f72a979a495f7591960a378abea7876ece7252ec1

Figura 3. Captura de pantalla de Opciones de descarga Android Studio en uso de software gratuito.

Fuente: elaboración propia

1.2.1. Configuración Android Studio

Después de instalar Android Studio y antes de empezar a programar, se deben tener en cuenta las siguientes consideraciones:

- Establecer la versión del JDK: en la versión de Android Studio 2.2 y posteriores, se incluye el paquete JDK para compilar las aplicaciones; si se desea utilizar el paquete incluido, en la opción de **File -> Project Structure**, en la categoría de **SDK Location** se selecciona la casilla **Use embedded JDK** (Figura 4).
- Configurar SDK de Android: para instalar cualquier versión de Android disponible y poder utilizarla en el desarrollo de aplicaciones para las diferentes versiones que contienen los dispositivos móviles, se selecciona la opción **Tools -> SDK Manager** y en la categoría **Android SDK**, en la pestaña **SDK Platforms**, se muestran las versiones instaladas y las disponibles para instalar; para esto se activa o desactiva la casilla de la versión que se pretenda instalar o desinstalar (Figura 5).

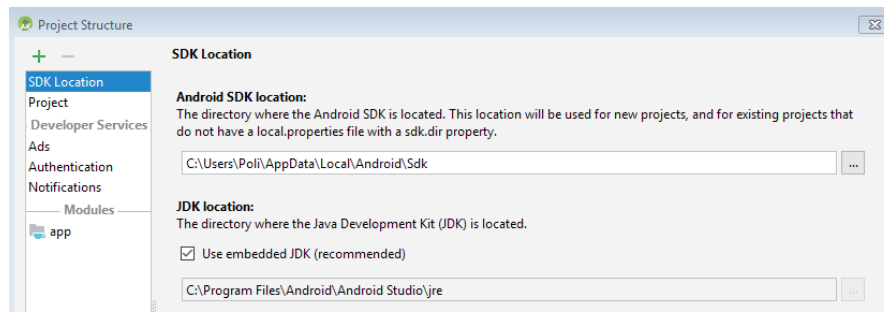


Figura 4. Captura de pantalla SDK location desde Android Studio en uso de software gratuito

Fuente: elaboración propia

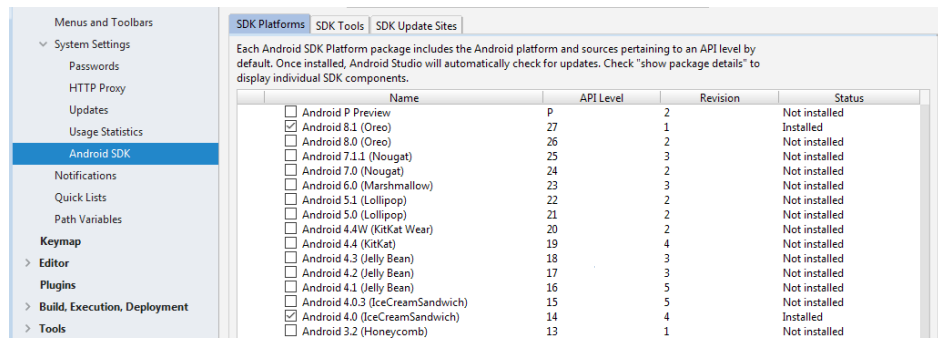


Figura 5. Captura de pantalla Android SDK Android Studio en uso de software gratuito

Fuente: elaboración propia

1.2.2. Creación del emulador Android Virtual Device (AVD)

La creación de un emulador permite definir las características de un teléfono Android, una tableta, un reloj o un televisor para simular las aplicaciones a desarrollar y ejecutarlas en este emulador sin necesidad de tener un dispositivo Android.

Para crear el emulador se selecciona la opción **Tools -> AVD Manager + Create Virtual Device New Hardware Profile** o se elige el dispositivo ya creado, señalando la correspondiente categoría (Category: TV, Phone, Wear, Tablet). Posteriormente, se selecciona la API a utilizar y, para finalizar, se verifica la configuración (Figura 6).

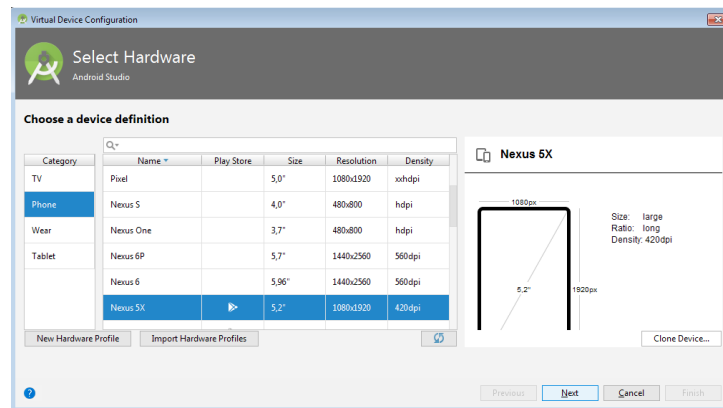


Figura 6. Captura de pantalla de proceso para crear el emulador desde Android Studio en uso de *software* gratuito

Fuente: elaboración propia

1.2.3. Configurar un dispositivo físico por USB

En el momento de ejecutar una aplicación se puede dirigir la salida hacia el emulador o hacia el dispositivo físico conectado por USB. En este último caso, para configurarlo se requiere tener instalada la herramienta Google USB Driver, como se muestra en la Figura 7; se selecciona la opción del menú **Tools -> SDK Manager -> Android SDK -> SDK Tools** y se debe tener habilitada la opción de desarrollo en el dispositivo.

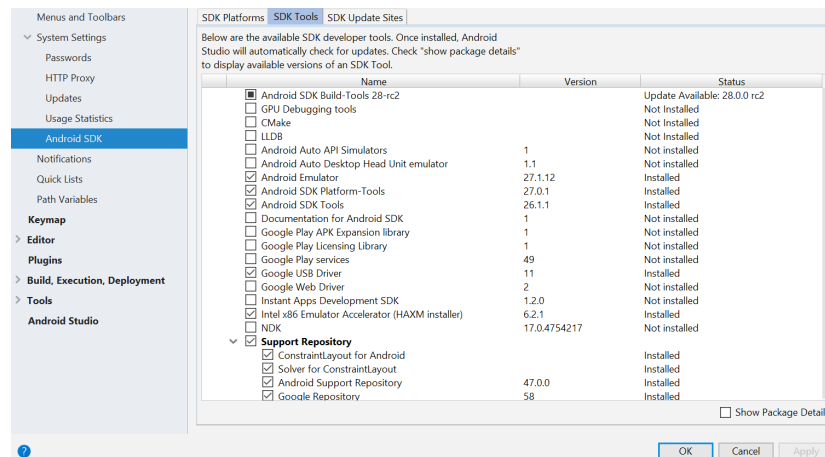


Figura 7. Captura de pantalla SDK Tools desde Android Studio en uso de *software* gratuito

Fuente: elaboración propia

2. Estructura de un programa Android Studio

Android Studio permite la creación de aplicaciones de Android para teléfonos, tabletas, televisores y relojes, entre otros. A continuación, se describe la creación de un proyecto y se estudia su estructura y sus componentes.

2.1. Creación de un proyecto Android Studio

Una vez iniciado Android Studio, para crear un nuevo proyecto, si es la primera vez que se ejecuta, se elige **Start a new Android Studio project** o se selecciona desde el menú **File -> New Project**; luego, el asistente presenta una serie de pantallas que se describen a continuación:

- a. *Create Android Project*: nombre de la aplicación; el nombre que recibe la aplicación a desarrollar.
Dominio de la compañía: un dominio que se constituye en la base para crear el paquete.
Package name: paquete de Java donde se guardan las clases de la aplicación.
Project Location: ruta en donde va a quedar almacenado el proyecto.
- b. *Target Android Devices*: se selecciona el dispositivo (*Phone and Tablet*, *Wear*, *TV*, *Android Auto*, *Android Things*), el destino de la aplicación y se elige la versión (API) a considerar. Dependiendo de la API seleccionada, estarán disponibles las características y será compatible con dispositivos de versiones anteriores.
- c. *Add an Activity to Mobile*: se elige la apariencia que tendrá la aplicación con base en una serie de plantillas que se presentan.
- d. *Configure Activity*: en esta sección se digita el nombre que va a recibir tanto la actividad como el *layout* dentro de la aplicación, los cuales se explican más adelante.

2.2. Estructura de un programa

Luego de la creación del proyecto, Android Studio muestra su interfaz de desarrollo. A continuación, se describe en forma general la estructura del proyecto (Figura 8):

- a. *manifests* - *AndroidManifest.xml*: contiene descripciones en XML de las características de la aplicación; por ejemplo, el nombre, el ícono, sus componentes y los permisos para ejecutar diferentes servicios, como el acceso a internet.

- b. Java: se almacena el código fuente de la aplicación, es decir, todas las clases de Java.
- c. res: se encuentran los recursos del proyecto como imágenes, el *layout* (diseño de la aplicación) y variables, entre otros.
- d. *Gradle Script*: contiene la información necesaria para la compilación de la aplicación. En esta sección se digita el nombre que va a recibir la actividad y el *layout* dentro de la aplicación. Esto se explica más adelante.

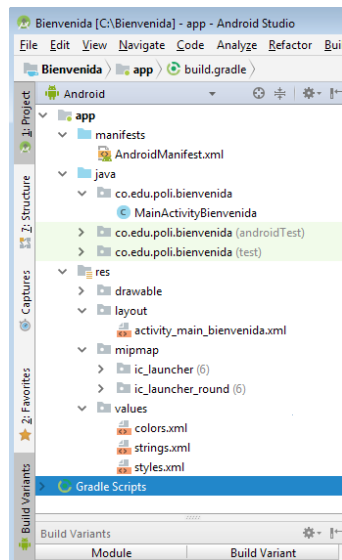


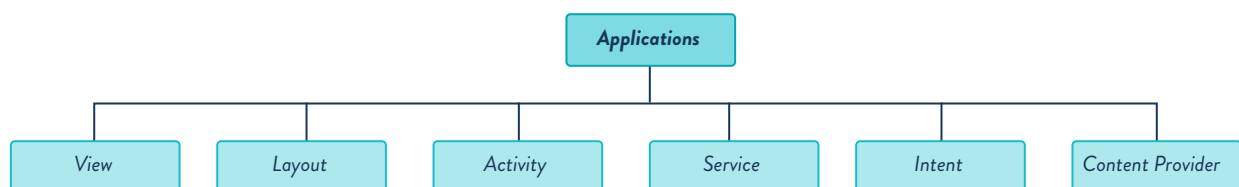
Figura 8. Captura de pantalla Estructura de un proyecto en Android Studio en uso de software gratuito

Fuente: elaboración propia

2.3. Componentes de una aplicación Android

El *framework* de Android permite desarrollar aplicaciones haciendo uso de una serie de componentes reutilizables. A continuación, se describen algunos de los componentes más importantes (Figura 9):

Figura 9. Componentes de una Aplicación Android



Fuente: elaboración propia

- a. *View*: las vistas representan la interfaz gráfica de una aplicación Android, tal como un botón de comandos o un campo de texto.
- b. *Layout*: representa un conjunto de vistas agrupadas; esto permite organizar las vistas en una determinada posición, alineación o justificación en la pantalla.
- c. *Activity*: conforma un conjunto de elementos a visualizar; una actividad es una ventana en donde se dibuja la interfaz del usuario.
- d. *Service*: corresponde a un servicio que se ejecuta en segundo plano sin necesidad de la interacción del usuario y sin interfaz gráfica; por ejemplo, se pueden actualizar datos en el dispositivo, mostrar notificaciones, gestionar conexiones de red, entre otros.
- e. *Intent*: es un mensaje que permite la comunicación entre elementos de la misma aplicación o de otras aplicaciones; por ejemplo, realizar una llamada telefónica, mostrar una página web, iniciar otra aplicación, etc.
- f. *Content Provider*: permite que se compartan datos entre aplicaciones sin comprometer la seguridad del sistema de archivos; por ejemplo, se puede compartir la lista de contactos.

Referencias

Gironés, J. (2013). *El gran libro de Android*. Barcelona, España. Editorial: Marcombo.

Robledo, D. (2017). *Desarrollo de aplicaciones para Android I*. España: Ministerio de Educación, Cultura y Deporte. Recuperado de <https://ebookcentral-proquest-com.loginbiblio.poligran.edu.co/lib/bibliopoligransp/detail.action?docID=5214411>

*Texto aclaratorio

En este material se han tomado capturas de pantalla de las plataformas Oracle y Android Studio con fines netamente educativos.

Android Studio. (2018). Capturas de pantalla desde *Android Studio* [Capturas de pantalla]. Recuperado de <https://developer.android.com/studio/?hl=es-419#downloads>

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN



Módulo: Herramientas de Programación Móvil I

Unidad 1: Conceptos básicos de Android y estructura de programación

Escenario 2: Estructura de un proyecto en Android Studio

Autor: Manuel Báez

Asesor Pedagógico: Angie Laitón

Diseñador Gráfico: Andrés Felipe Figueroa

Asistente: Alejandra Morales

Este material pertenece al Politécnico Gran Colombiano.

Prohibida su reproducción total o parcial.