ANDROID STUDIO.

Vinicio Geovanny Guachamin Quingaluisa

vynypm@hotmail.com

Jessica Fernanda Loachamin Topón

[jessy.flt@gmail.com](mailto:jessy.flt@gmail.com)

Escuela Politécnica Nacional

**RESUMEN:** *El desarrollo de aplicaciones móviles en la actualidad es lo más demandado por los programadores, cabe aclara que para desarrollar una aplicación móvil existen diferentes formas y métodos, pero una de ellas y una de las más utilizadas es mediante Android Studio, esta es la razón por la cual nuestro trabajo se centrara en Android Estudio, pero antes de esto debemos definir ¿Que es Android?*

*Para comenzar diremos que Android es un sistema operativo enfocado a celulares táctiles o tablets, basado en una versión modificada del núcleo Linux. En sus inicios fue desarrollado por la empresa Android Inc., posteriormente fue comprada por Google y en la actualidad lo desarrollan los miembros de la Open Handset Alliance y por Google, este es un sistema abierto, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones las cuales se las desarrolla en los lenguajes de programación Java y C.*

*Android Studio es un entorno de desarrollo integrado para la plataforma Android, de manera que nuestra investigación nos llevara a detallar los requerimientos del sistema para este entorno, su instalación, definiremos básicamente los componentes y herramientas y realizaremos una pequeña aplicación a manera de ejemplo didáctico.*

# INTRODUCCIÓN

En este informe nos vamos a enfocar más al entorno de desarrollo Android Studio el cual nos sirve para crear aplicaciones Android, para entrar un poco más en detalle de este entorno de desarrollo diremos que que Android Stuido fue lanzado oficialmente el 16 de mayo del 2013 y de esta forma dejo atrás o mejor dicho reemplazó a Eclipse.

Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. [1]

Este programa se lo puede descargar desde https://developer.android.com/sdk/index.html

# CARACTERÍSTICAS

## CARACTERÍSTICAS GENERALES

Android Studio se caracteriza por ser pesado, ya que tiene un peso aproximado a 800 mb, además para que esta plataforma sea funcional se tiene que descargar varios APIS y herramientas.

“Android Studio se ha mantenido como una versión beta, pero desde el 8 de diciembre de 2014, en que se liberó la versión estable de Android Studio 1.0, Google ha pasado a recomendarlo como el IDE para desarrollar aplicaciones.

Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.

Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna).

Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.

Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.

Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion.

Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.

Vista previa en diferentes dispositivos y resoluciones.

Integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.

Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo XML.” [2]

# REQUERIMIENTOS DEL SISTEMA

Los requerimientos de Android Studio depende mucho del sistema operativo en el que va ser instalado, a continuación detallaremos los requisitos:

*Tabla 1. Requerimientos del Sistema.*

|  |  |  |
| --- | --- | --- |
| **Windows**  **Mac OS**  **Linux** | | |
| Microsoft Windows 8/7/Vista/2003 (32 o 64 bit) | Mac OS X 10.8.5 o superior, hasta la 10.9 (Mavericks) | GNOME o entorno de escritorio KDE |
| Mínimo de 2 GB de RAM, recomendado 4 GB de RAM | | |
| 400 MB de espacio en disco | | |
| Necesita de al menos 1 GB para Android SDK, emulador de imágenes del sistema, y cachés | | |
| Resolución mínima de pantalla de 1280 x 800 | | |
| Java Development Kit (JDK) 7 o superior | | |

Para los sistemas Linux será necesaria la biblioteca de C GNU (glibc) 2.11 o posterior. Además para los sistemas Mac OS será necesario ejecutar Android Studio con Java Runtime Environment (JRE) 6 para la renderización optimizada de fuentes.

# INSTALACIÓN

En primer lugar se deberá descargar Android Studio. Todo depende del sistema operativo en el que vamos a instalar la plataforma, en nuestro caso utilizaremos Windows 7 64 bits.

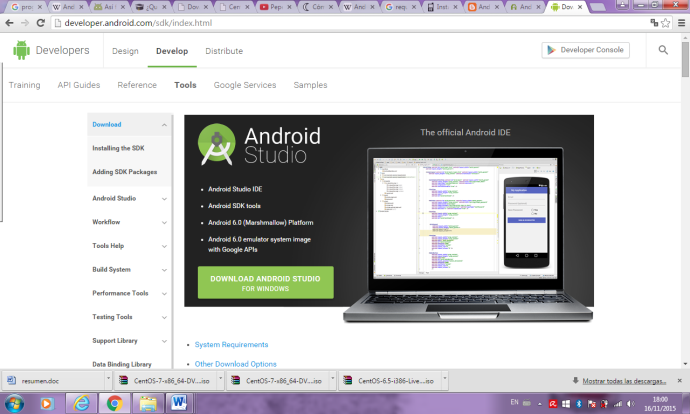
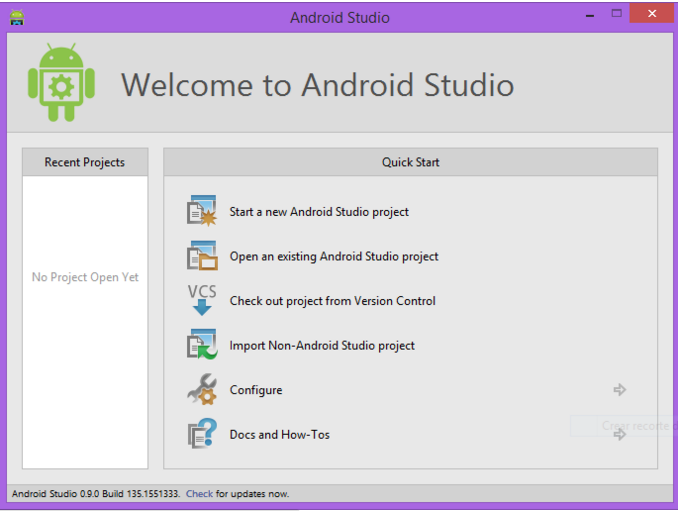
La descarga de Android Studio se puede realizar a través de: <http://developer.android.com/sdk/index.html>.

Figura 1. Página principal de descarga de Android

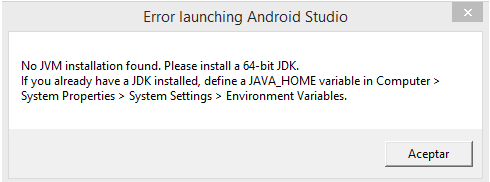
Una vez hecho la descarga, seleccionamos la arquitectura del microprocesador (studio.exe correspondiente a la arquitectura, en nuestro caso studio64.exe que corresponde a 64 bits.

Previo a la instalación de Android Studio deberemos tener instalado Java SDK, o podemos descargarnos este programa de <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Si ya tenemos instalado SDK de Java, simplemente se debemos realizar doble click sobre el archivo con extensión .exe para acceder a la pantalla de bienvenida de Android Studio.

Figura 2. Página principal de instalación

En caso de no tener bien instalado SDK de java tendremos un error.

Figura 3. Error de instalación

Para solucionar esto deberemos crear una variable de entorno, así que accederemos a la opción “Equipo”-“Propiedades del sistema”.

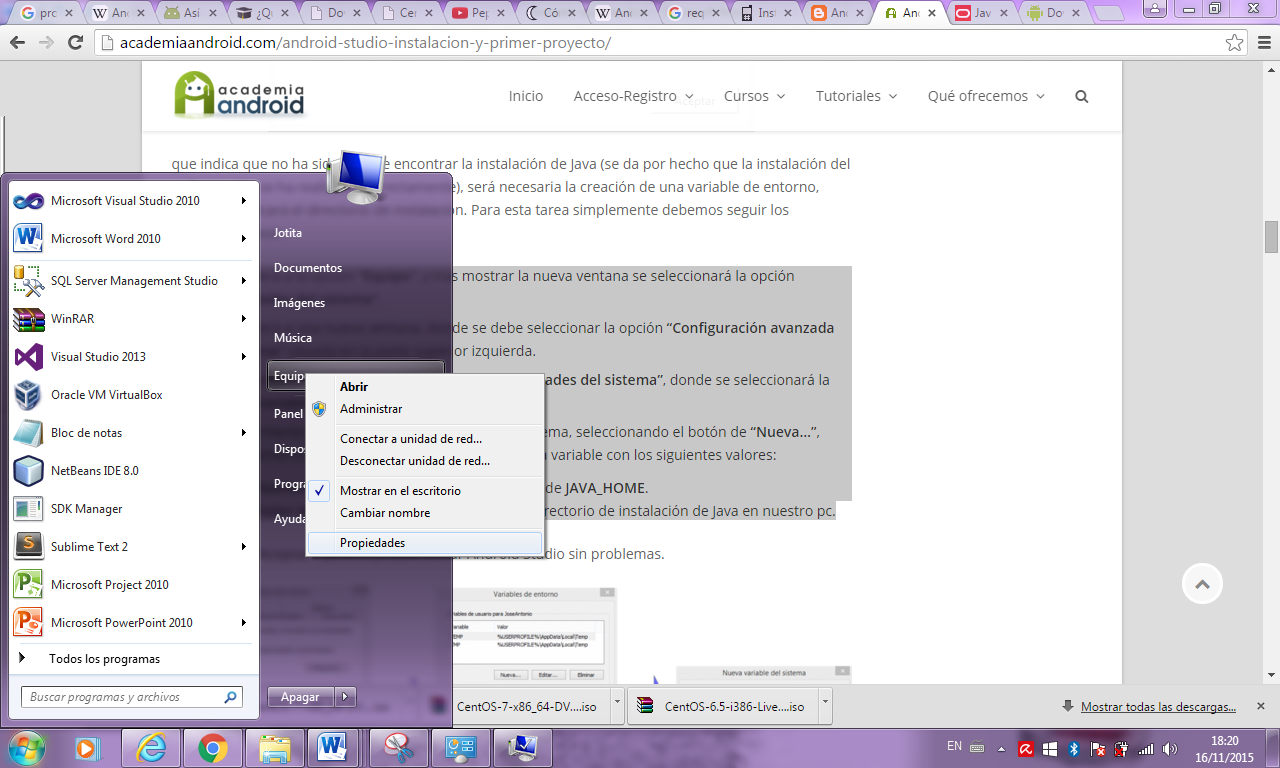


Figura 3. Propiedades del sistema

Accederemos a una nueva ventana, donde seleccionaremos la opción “Configuración avanzada del sistema” y aparecerá una ventana llamada “Propiedades del sistema”, donde se seleccionará la opción “Variables de entorno”.

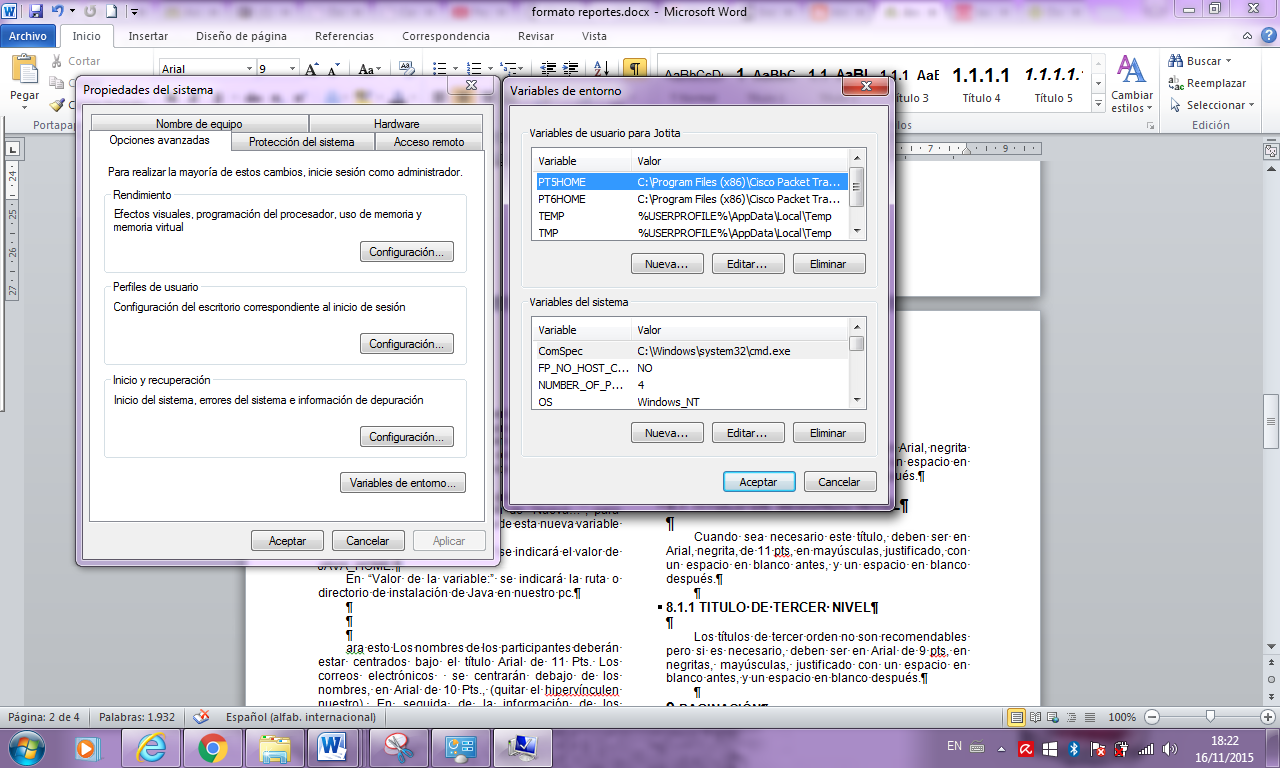


Figura 4. Variables del entorno

Creamos una nueva variable de sistema, seleccionando el botón de “Nueva”, donde escribiremos “Nombre de la variable:” se indicará el valor de JAVA\_HOME y en “Valor de la variable:” se indicará la ruta o directorio de instalación de Java en nuestro pc.

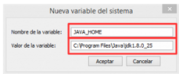


Figura 5. Creación de la variable SDK

Una vez creada la variable volvemos a ejecutar el instalador de Android Studio.

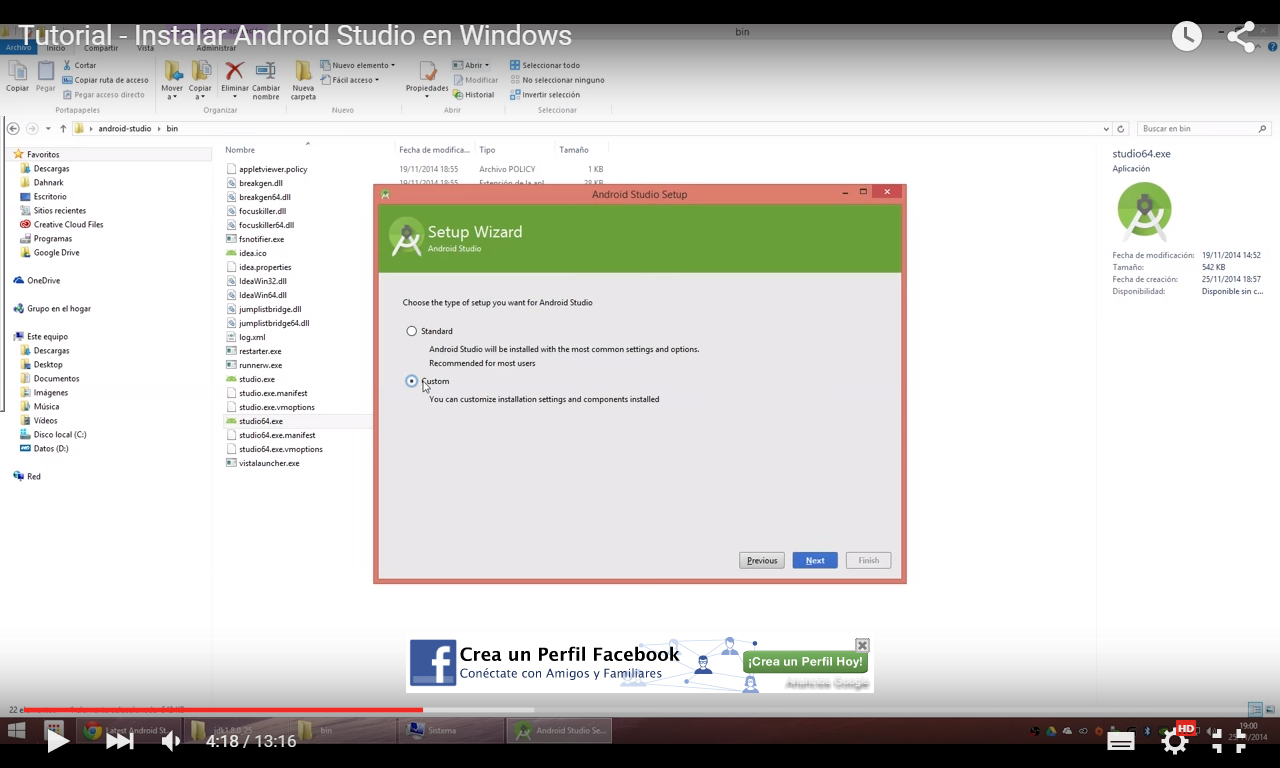


Figura 6. Instalación Android Studio

En la siguiente ventana nos indicara la ruta donde se guardara el programa por lo cual solo damos finalizar.

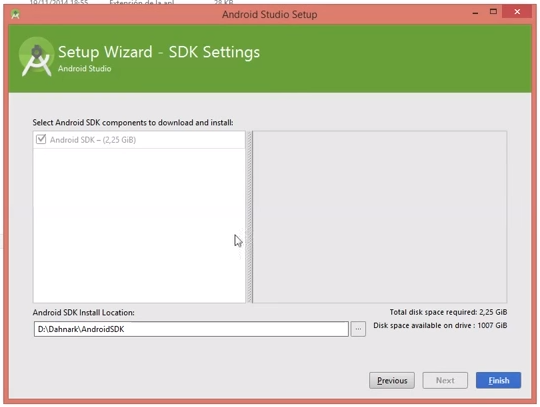


Figura 7. Instalación Android Studio

Con esto empezara a instalarse la plataforma.

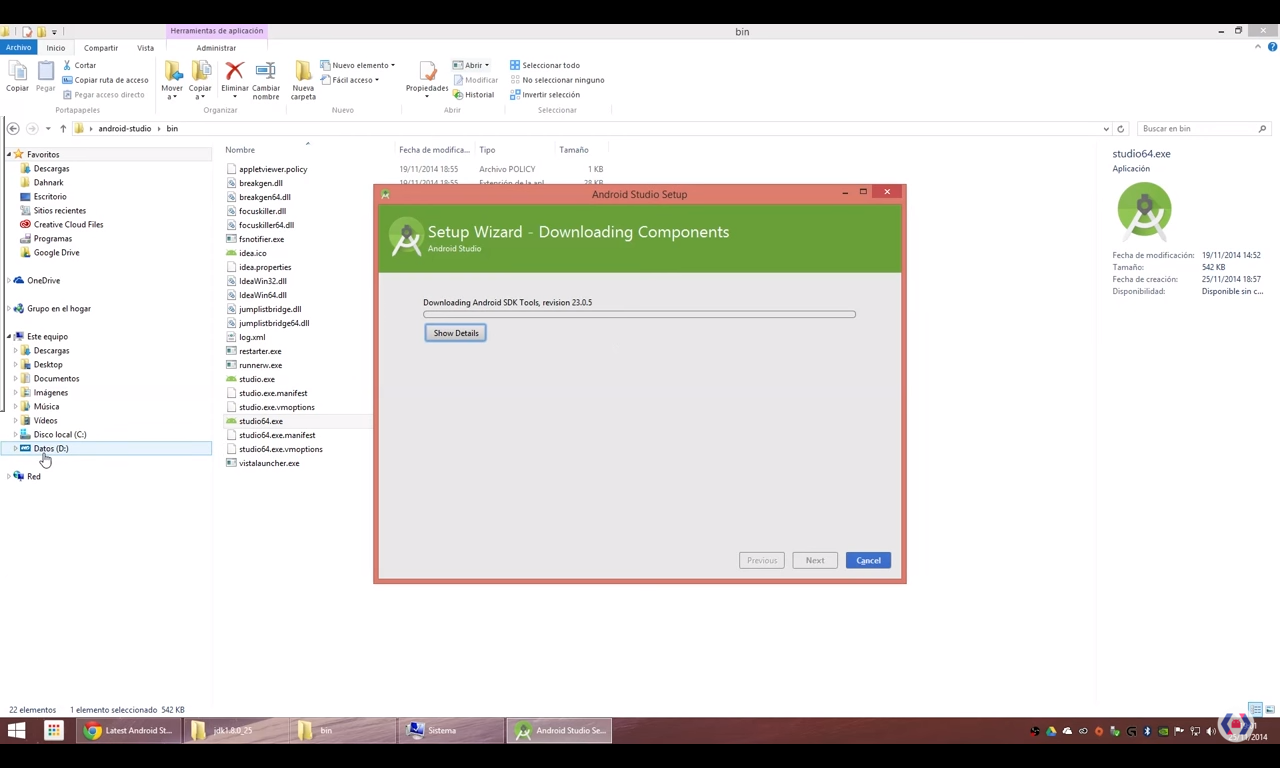


Figura8. Instalación Android Studio

Una vez terminada la instalación procederemos a instalar el SDK Manager el cual nos ayuda a instalar todas las librerías o APIS de Android Studio y sus herramientas.

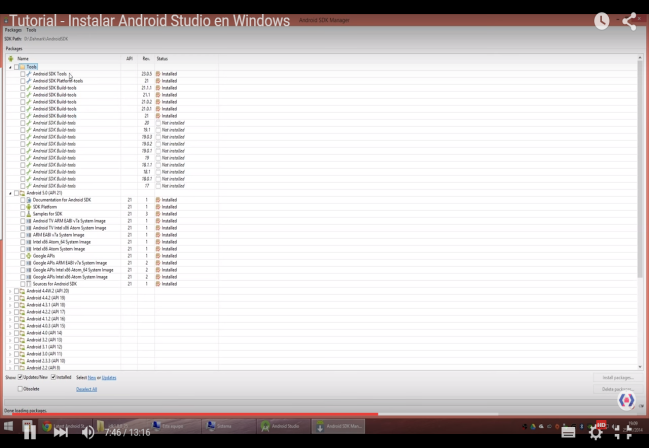


Figura9. Instalación SDK Manager

Como recomendación debemos estar conectados a internet para la instalación de Android Studio además de que debemos instalar todas las herramientas sugeridas por el SDK Manager ya que sin ellas el programa no servirá,

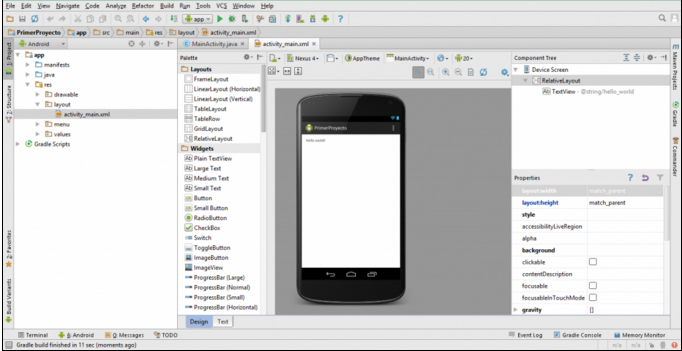
Este programa es muy pesado y a su vez el tiempo de instalación es muy largo.

Figura 10. Android Studio

# APIS DE ANDROI

## DEFINICIÓN

“La interfaz de programación de aplicaciones o API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las "librerías".

Es una fuente de código basado en la especificación destinada a ser utilizada como una interfaz de componentes de software para comunicarse entre sí.

Una API puede incluir especificaciones para las rutinas y estructuras de datos, las clases de objetos y variables.

Una API difiere de una interfaz de aplicación binaria (ABI) en la que el primero es de código abierto basado mientras que el segundo es una interfaz binaria. Por ejemplo, es una API POSIX, mientras que la base estándar de Linux es un ABI.” [3]

## APIS DE ANDROID

Android al ser un sistema relativamente nuevo inicio su uso en teléfonos móviles y es por esto que posee su propia API para la gestión de llamadas telefónicas y mensajes de texto.

Esta API se encuentra disponible en el paquete de Android. Telephony, permitiendo así el acceso a la información de telefonía.

Es por esto que al realizar una aplicación para este sistema operativo es necesario definir los permisos en el archivo manifest.xml del proyecto.

El SDK de Android tiene clases que permite hacer uso del teléfono siempre que el dispositivo cuente con este módulo.

Las clases más importantes de esta API TelephonyManager y SMSManager.

Otras librerías importantes de Android son:

android.util: El paquete básico de servicios públicos contiene las clases de bajo nivel, como contenedores especializados, formateadores de cadenas, y de análisis XML de servicios públicos.

android.os: El paquete de sistema operativo permite el acceso a los servicios básicos como el paso de mensajes, la comunicación entre procesos y funciones de reloj.

android.graphics: La API de gráficos, es el suministro de las clases de bajo nivel como lienzos de apoyo, colores y las primitivas de dibujo. También le permite dibujar sobre lienzos.

android.text: Las herramientas de procesamiento de texto para mostrarlo y analizarlo.

android.database: Proporciona las clases de bajo nivel necesario para la manipulación de cursores cuando se trabaja con bases de datos.

android.content: El contenido de la API se utiliza para admirar el acceso a los datos y a la publicación, proporcionando los servicios para hacer frente a los recursos, los proveedores de contenido y los paquetes.

android.view: Las vistas son un núcleo de la interfaz de usuario. Todos los elementos de la interfaz se construyen utilizando una serie de vistas que proporcionan los componentes de interacción con el usuario.

android.widget: Construido sobre el paquete de Vista, estan las clases widget "aquí está uno que creamos antes", elementos de la interfaz de usuario para su uso en las aplicaciones. Se incluyen listas, botones y diseños.

com.google.android.maps: API de alto nivel que proporciona acceso a los controles de mapas que usted puede utilizar en su aplicación. Incluye el control MapView así como la superposición y la clase MapController utilizados para anotar y controlar dichos mapas.

android.app: Paquete de alto nivel que proporciona el acceso al modelo de solicitud. Este, incluye la actividad de servicios y las API que forman la base de todas sus aplicaciones.

android.provider: Para facilitar el acceso a los desarrolladores a determinados proveedores de contenidos estándar, el paquete proveedor ofrece clases para todas sus distribuciones.

android.telephony: Las API´s de telefonía le dan la posibilidad de interactuar directamente con el dispositivo de Teléfono, permitiéndole realizar, recibir y controlar las llamadas de teléfono, su estado y mensajes SMS.

android.webkit: Ofrece funciones para trabajar con contenido basado en web, incluyendo un control WebView para incrustar los navegadores en sus actividades y un administrador de cookies.

# PROGRAMACION EN ADROID STUDIO

## VARIABLES TIPO CHAR O DE CARÁCTER:

Son variables que almacenan caracteres individuales (letra, numero, signo, etc...). El carácter que se inicializa debe ir entre apóstrofes o comillas simples 'a'.

El código de caracteres empleado por Java es Unicode y recoge los caracteres de prácticamente todos los idiomas importantes del mundo (son unos 65.536). Los caracteres Unicode del alfabeto occidental corresponden a los primeros 256 enteros; es decir van desde [0, 255].

A cada carácter le corresponde un número entero perteneciente al intervalo [0, 65536] o a [0, 255] si se trabaja sólo con el alfabeto occidental. Por ejemplo, la letra ñ es el entero 164. Más adelante se verá que el casting entre variables primitivas enteras y la variable char están permitido. Por ejemplo: char miCaracter='n';

char miCaracter1=110; (ídem antes, pero mediante el entero que le corresponde según Unicode) char miCaracter2='\u006E'; (ídem antes, pero según notación Unicode.

La notación Unicode, en general, es de esta forma: \uXXXX siendo X un dígito o cifra fundamental del sistema de numeración hexadecimal (0, 1,2,..., 9, A, B,..., F))

Asociado a este tipo de variable se tienen las secuencias de escape. Se emplean para representar caracteres especiales (por ejemplo, unas comillas dentro de una instrucción que exige una cadena entrecomillada) y caracteres no imprimibles como el tabulador, salto de línea, etc. Van precedidos de la barra "\", por ejemplo:

*Tabla 2. Tipo de caracteres*

|  |  |
| --- | --- |
| **Carácter** | **Denominación** |
| \n | Salto de línea |
| \t | Tabulador |
| \" | Comilla doble |
| \\ | Contrabarra |

## REALES O DE COMA FLOTANTE

Son variables que almacenan datos numéricos con decimales. Se pueden dividir en los siguientes tipos:

*Tabla 3. Tipo de caracteres*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Bytes ocupados en memoria** | **Nombre** | **Rango** |
| float | 4 | real de simple precisión | Negativos: [-3.4E38,-1.4E-45] Positivos: [1.4E-45,3.4E38] |
| double | 8 | real de doble precisión | Negativos: [-1.8E308,-4.9E-324] Positivos: [4.9E-324,1.8E308] |

## TIPO ENTERO

Son variables que almacenan números enteros. Se pueden dividir en los siguientes tipos:

*Tabla 3. Tipo de caracteres*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Bytes ocupados en memoria** | **Nombre** | **Rango** |
| byte | 1 | byte | [-128,127] |
| short | 2 | entero corto | [2768,32767] |
| int | 4 | entero | [-2^31,2^31-1] |
| long | 8 | entero largo | [-2^63,2^63-1] |

# EJERCICIO PRÁCTICO

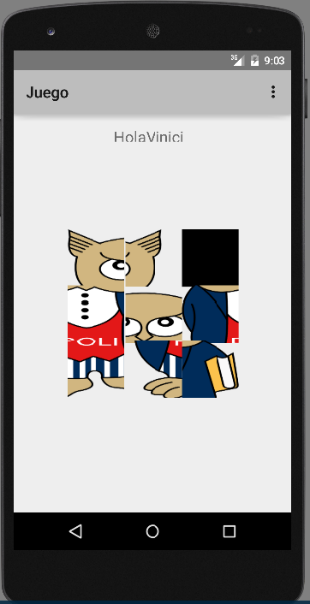
**Funcionalidad de aplicación**

Para la explicación práctica de lo que es Android Studio hemos pensado en realizar una aplicación muy didáctica como lo es un rompecabezas, esta aplicación consta de dos ventanas:

1. Ventana de entrada: donde para ingresar al juego nos pide nuestro nombre con la finalidad de que la aplicación sea amigable con el usuario y aparte del botón “Jugar” que obviamente ya sabemos para qué sirve, esta aplicación posee el típico botón “Acerca de” en la cual nos brindará la información de las personas que realizamos esta aplicación.



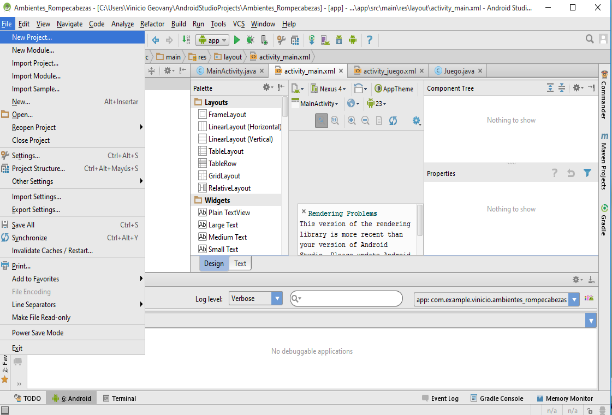
1. Ventana de Juego: ya en esta ventana es donde está nuestro rompecabezas, el cual consiste en organizar correctamente las piezas deslizándose de un lado a otro hasta formar el dibujo del Búho politécnico.



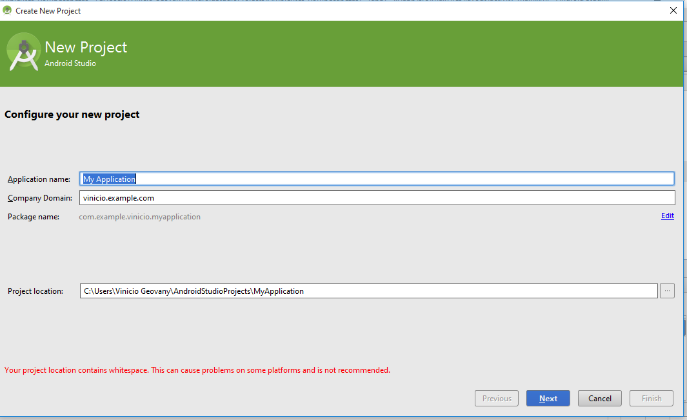
**Desarrollo aplicación**

Bueno lo primero que debemos hacer es crear un proyecto nuevo en Android Studio, a continuación explicaremos de forma detallada como crear un proyecto:

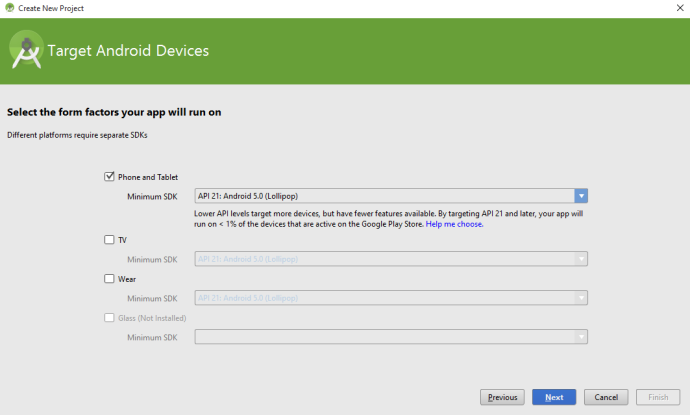
1. Una vez iniciado Android Studio, damos clic en la pestaña File y luego en New Project.



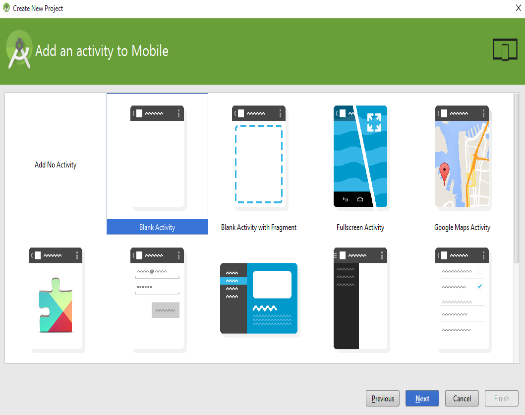
1. Le damos un nombre a nuestro a nuestro proyecto y damos siguiente.



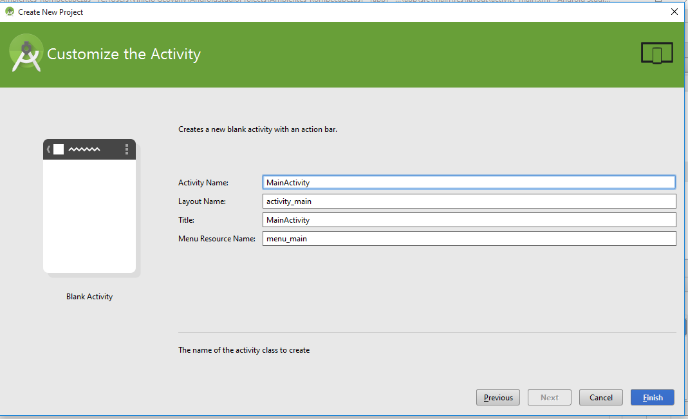
1. Seleccionamos para que plataforma vamos a desarrollar la aplicación, en nuestro caso seleccionamos la opción Phone and Tablet y damos clic en siguiente.



1. Aquí escogemos con qué tipo de actividad vamos a comenzar a desarrollar nuestra aplicación por lo general escogemos Blank Activity y le damos siguiente.



1. Por ultimo damos un nombre a nuestra Activity que hemos escogido.



Ahora automáticamente se nos van a generar dos archivos el uno en formato xml donde vamos a realizar el diseño de nuestra aplicación (ver gráfico 11) y el otro en formato java donde vamos a poder realizar la programación de nuestra aplicación (ver gráfico 12).

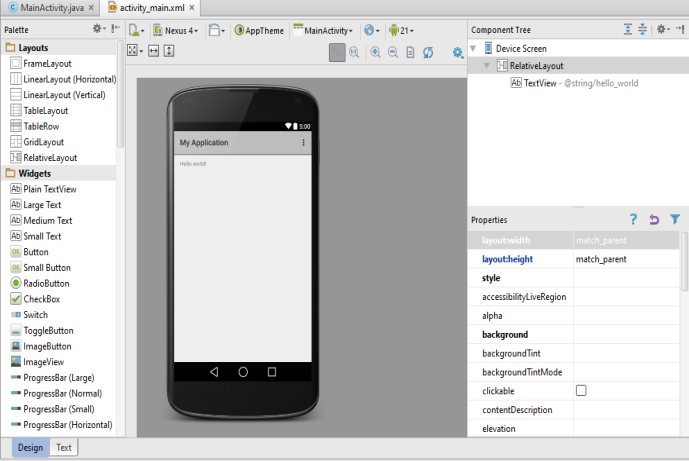


Gráfico 11.

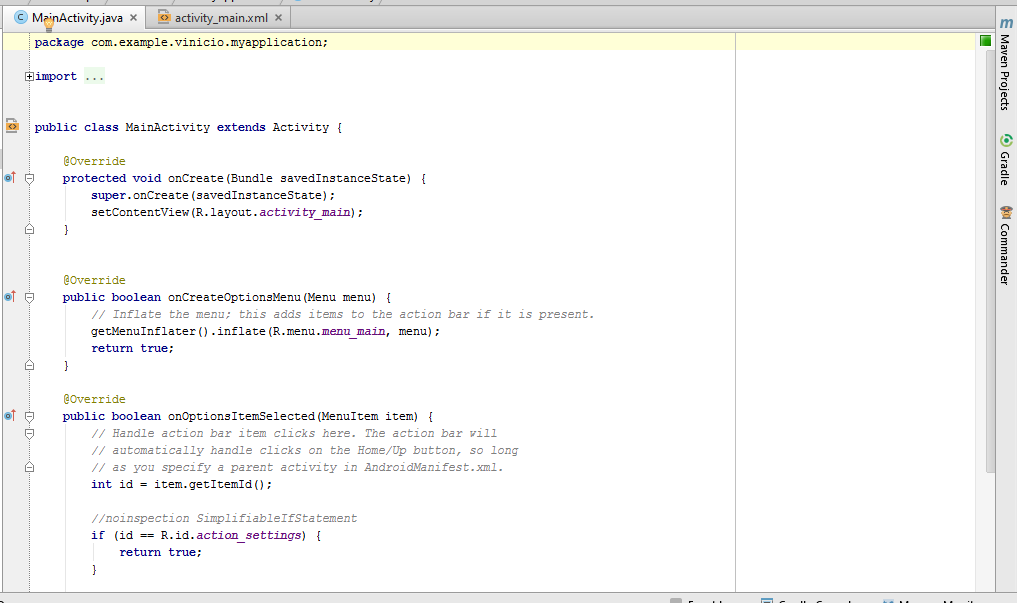
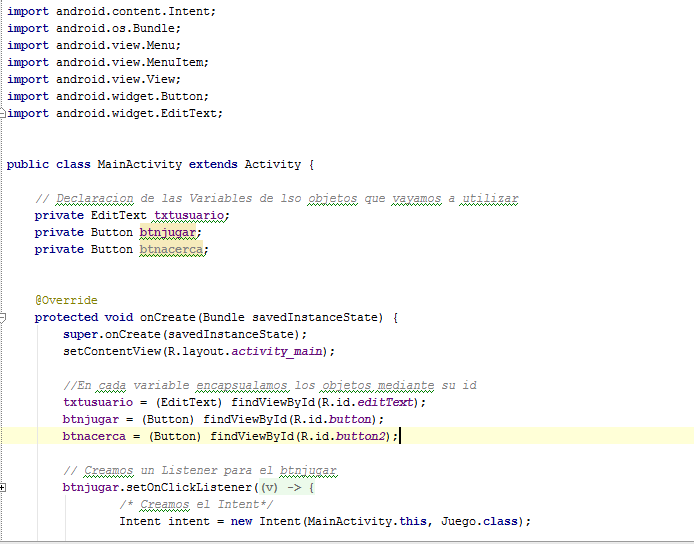


Gráfico 12.

En nuestro caso para realizar la aplicación hemos tenido que crear un archivo en formato java y otro en formato xml aparte de los que ya se crean por defecto, para el desarrollo de esta aplicación, la razón es que nuestra aplicación presenta dos ventanas entonces cada ventana es una archivo xml con su respectivo archivo en formato java también cabe mencionar que para programar en Android Studio debemos saber muy bien programar en java, a continuación presentaremos unas imágenes del código fuente de nuestra aplicación desarrollada:





# REFERENCIAS

|  |  |
| --- | --- |
| [1] | E. C., «Wikipedia,» 13 02 2015. [En línea]. Available: https://es.wikipedia.org/wiki/Android\_Studio. [Último acceso: 15 11 2015]. |
| [2] | A. Andrpid, «Academia Android,» [En línea]. Available: http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/. [Último acceso: 15 11 2015]. |
| [3] | . R. Camacho, «Dispositivos Moviles Laboratorio Android :),» 12 04 2012. [En línea]. Available: http://rcmdispmoviles.blogspot.com/2012/04/interfaz-de-programacion-de.html. |
| [4] | «Androformacion,» Curso de Programación en Android con Android Studio, 01 01 2015. [En línea]. Available: http://www.adrformacion.com/cursos/androidst/leccion2/tutorial4.html. [Último acceso: 15 11 2015]. |

.