***ANDROID STUDIO***

***PRIMEROS PASOS***

***Docente: German Riveros***

***Estudiante: Eric Monsalve***

Toda aplicación que crea para android sigue una misma estructura básica, que se compone del código fuente en sí, archivos de recursos y vistas, librerías de código y el android manifest.

Directorio SRC

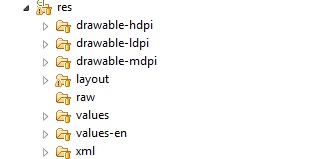
Se encuentra **toda la lógica de aplicación**, todas las clases programadas en JAVA. Dentro de ella puedes definir distintos paquetes, donde puedes dividir en capas tus reglas de negocio.

Android Library

Aquí se encuentran todas las librerías propias del SDK de android, dependiendo la versión elegida al crear el proyecto tendrá una versión u otra.

Directorios RES

Se encuentran todos los archivos con los recursos que usan la aplicación. Las imágenes, archivos de idiomas, estilos, etc.. Un pequeño listado de estos puede ser:



* **Drawable** => Carpeta con todas las imágenes de la app. Se subdivide en múltiples carpetas desde la versión 1.6, que contienen las imágenes en distintas resoluciones y tamaños que se usarán dependiendo el dispositivo usado.
* **Directorio layout** => Aquí se encuentran las distintas “pantallas” de la aplicación, es decir, los archivos xml con las interfaces visual asociadas a las activities.
* **Values** => Carpeta con los xml de contenido de la app. En ella puede haber definidas las constantes de la aplicación, dando la posibilidad del multidioma. También puedes definir estilos para tus componentes. Y todo tipo de configuraciones.

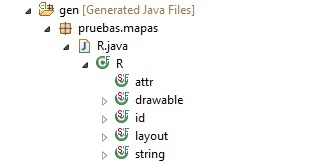
Directorio bin

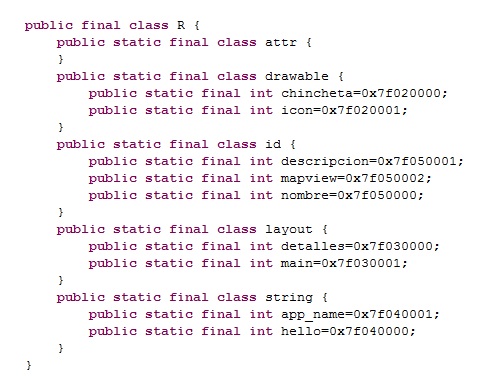
Aquí se encuentran todos los archivos generados por la propia app. Como sabeis, android usa la máquina virtual dalvik, primero se traduce a los típicos archivos .class de java y posteriormente es traducido a los archivo .dex propios de android.

También está el **ejecutable de la aplicación "apk"**, sería el equivalente a los "exe" de windows. Es el archivo que deberías instalar en cualquier teléfono android para probar la aplicación.

Directorio gen

En esta carpeta esta el archivo **R.class, éste contiene lo identificadores los recursos usados por tu proyecto**: imágenes, layout, etc...





Directorio assests

Carpeta donde se encuentran los **archivos auxiliares de tu aplicación**: imágenes, audios, vídeos... la diferencia con los que se encuentran con la carpeta "RES", es que los archivos incluidos aquí no generarán un identificador dentro del archivo R.class anteriormente descrito.

Para usar estos archivos, en vez de referenciarlos por un ID, habría que usar la ruta física como cualquier otro archivo

Directorio lib

Aquí irán las librerías externas importados que necesites. Por ejemplo, si deseas meter publicidad en tu app, aquí ira la librería necesaria para ello.

## Estructura del Android Manifest

Empieza con una definición del xml en el que se encoda:

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="nombre.paquete"

android:versionCode="1"

android:versionName="1.0">

El número de versiones debe irse modificando en el android manifest con cada versión "publicada".

Icono de la aplicación que aparecerá posteriormente al instalarse en los menus y nombre de la misma almacenador en el archivo de strings. <application android:icon="@drawable/icon" android:label="@string/app\_name">

Todas y cada una de las activitys de la app deben ser declaradas en esta sección, sino no funcionarán. La actividad inicial de la aplicación debe ser declarada de tipo "LAUNCHER":

<application android:icon="@drawable/icon" android:label="@string/app\_name">

//Declaración de todas las activities de la app:

<activity android:name=".actividad1" android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

//El resto son opcionales dependiendo de la app:

<activity-alias>

<intent-filter> . . . </intent-filter>

<meta-data />

</activity-alias>

<service>

<intent-filter> . . . </intent-filter>

<meta-data/>

</service>

<receiver>

<intent-filter> . . . </intent-filter>

<meta-data />

</receiver>

<provider>

<grant-uri-permission />

<meta-data />

</provider>

//Pueden ser declaradas constantes para su uso en todo el programa:

<meta-data android:value="1" android:name="constante1" />

</application>

<uses-permission /> //Permisos que el usuario debe aceptar al instalar

<permission /> //Permisos de la app

<uses-sdk /> //Configuración de las versiones soportadas.

<supports-screens /> //Pantallas soportadas

<compatible-screens /> //Pantallas compatibles

<uses-configuration />

<uses-feature />

<permission-tree />

<permission-group />

<instrumentation />

</manifest>

#### ****Crear una aplicación desde cero****

#### ****Descarga e instalación de los archivos necesarios****

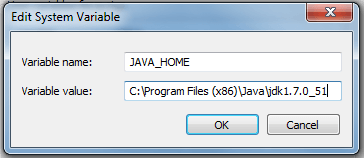
##### Instalación:

* Asegurarse de instalar la versión correcta de los programas (32 o 64 bits)
* **Posibles problemas en este primer paso:**

Muchas veces **Android Studio** falla al arrancar porque tiene problemas para detectar el JDK. Esto puede deberse a:

**1.**Instalación incorrecta de Java: los sistemas operativos de 64 bits permiten la instalación de la versión 32 bits del JDK sin ningún problema y viceversa. El problema de instalar la versión incorrecta llega cuando intentamos ejecutar Android Studio. Revisa la versión de tu sistema operativo y asegúrate de haber instalado la versión apropiada del JDK.

**2.** Si ese no es tu caso y el problema persiste siempre podemos indicarle manualmente a Android Studio dónde se encuentra instalado Java. Para eso tenemos que dirigirnos a … Panel de control → Sistema → Opciones avanzadas del sistema → Avanzado → Variables de entorno y una vez ahí crear una variable llamada “JAVA\_HOME” (sin comillas) y agregarle como valor la ubicación del Java Development Kit en nuestra PC. Nos debería quedar algo así:



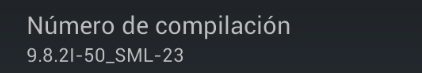
#### ****2. Activar la depuración USB en nuestro teléfono****

La depuración USB es una modalidad exclusiva de Android que autoriza el envío de comandos desde la computadora al celular por medio de un cable USB. Es necesario activarla para poder previsualizar la aplicación que vamos a programar en nuestro propio dispositivo.

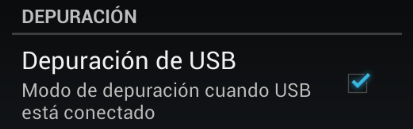
**1.** Ir a Configuración → Opciones desarrollo y activarlas presionando el interruptor “Encendido – Apagado”.

2

**2.** Si la sección “Opciones de desarrollo” no aparece como opción en “Configuración” (como es el caso de los Google Nexus) tenemos que dirigirnos a “Acerca del teléfono” y tocar el “Número de compilación” repetidamente hasta que un mensaje nos indique que ya somos desarrolladores.

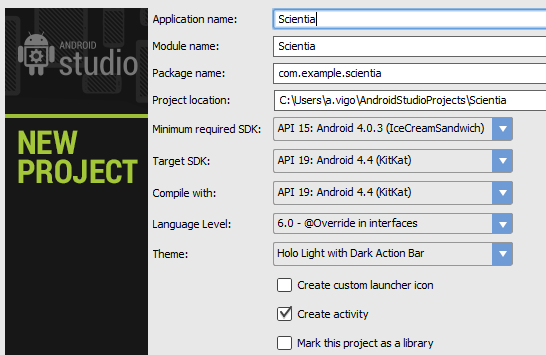


**3.** Una vez habilitadas las opciones de desarrollo debemos activar la **depuración de USB,**que se encuentra debajo de la categoría “Depuración”

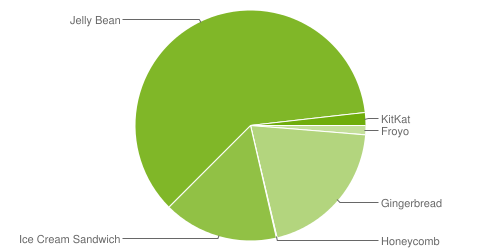


#### 3. Creando la aplicación

**1.** Una vez en Android Studio vamos a comenzar un **Nuevo proyecto.**Se mostrará una lista de campos que debemos completar de la siguiente manera:



* **Nombre de la aplicación:**el nombre de la aplicación. Será mostrado – por ejemplo – en Google Play.
* **Nombre del módulo:**utilizado por Android Studio. Suele ser igual al nombre de la aplicación.
* **Nombre del paquete:**un identificador único de nuestra aplicación.
* **SDK mínimo requerido:**versión de Android mínima necesaria para correr la aplicación. Trabajar con versiones más viejas aumenta la cuota de mercado pero reduce las posibilidades de la app.
* **SDK objetivo:**máxima versión de Android con la que podrá ejecutarse la aplicación.
* **Nivel de lenguaje:**versión de Java con la que se desarrollará la aplicación. Recomiendo utilizar Java 6 porque la única manera de que nuestra aplicación soporte todas las características de Java 7 es eligiendo API 19como SDK mínimo.
* **Ícono de la aplicación:**el ícono que representará la aplicación. Si ya tienes un imagen seleccionada marca el casillero “create custom launcher icon”

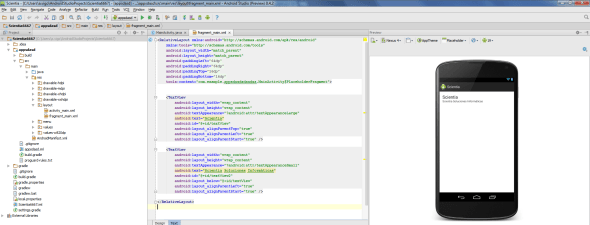


Cuota del mercado de Android

**2.**Seleccionamos “Continuar” y elegimos la opción “Actividad en blanco”. Este modelo de aplicación es el más básico de todos; presenta un ícono en la esquina superior derecha que desplegará una lista de opciones y una flecha situada en la esquina superior izquierda que nos permite ir atrás.

**3.**En la próxima parte se mostrarán tres nuevos campos. Vamos a dejarlos sin editar y proseguiremos a terminar.

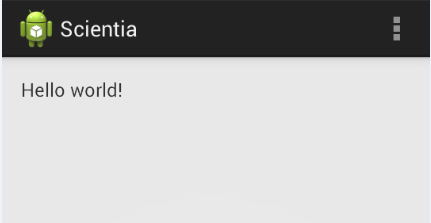
**4.**Ahora queda esperar a que Android Studio configure la aplicación correctamente. El proceso demora de 2 a 5 minutos.

[](https://i1.wp.com/www.programandoamedianoche.com/wp-content/uploads/2014/03/entorno.png)

Entorno de la aplicación

#### 4. Probando la aplicación

**1.**Una vez Android Studio haya creado el proyecto procedemos a conectar nuestro celular a la computadora. Hecho esto, utilizamos el atajo Shift+F10en AS para probar la aplicación. En alrededor de 15 segundos, la aplicación debería estar reproduciéndose en tu celular. En nuestro caso, quedó algo así:



Reproduciéndose en un Motorola Droiz RAZR i

#### 5. Agregando contenido a la aplicación

**1.**Para agregar contenido a nuestra aplicación debemos editar el archivo **fragment\_main.xml** ubicado en la carpeta layout.

Para dejar la aplicación en blanco debemos remover las siguientes líneas del archivo .xml

|  |  |
| --- | --- |
| 1 | <TextView |
| 2 | android:text="@string/hello\_world" | |

|  |  |
| --- | --- |
| 3 | android:layout\_width="wrap\_content" |
| 4 | android:layout\_height="wrap\_content" /> | |

Una vez hecho eso vamos a agregar un título que diga “Scientia”:

|  |  |
| --- | --- |
| 1 | <TextView |
| 2 | android:layout\_width="wrap\_content" | |

|  |  |
| --- | --- |
| 3 | android:layout\_height="wrap\_content" |
| 4 | android:textAppearance="?android:attr/textAppearanceLarge" | |

|  |  |
| --- | --- |
| 5 | android:text="Scientia" |
| 6 | android:id="@+id/textView" | |

|  |  |
| --- | --- |
| 7 | android:layout\_alignParentTop="true" |
| 8 | android:layout\_alignParentLeft="true" | |

|  |  |
| --- | --- |
| 9 | android:layout\_alignParentStart="true" /> |

Y por debajo del título un texto de tamaño normal:

|  |  |
| --- | --- |
| 1 | <TextView |
| 2 | android:layout\_width="wrap\_content" | |

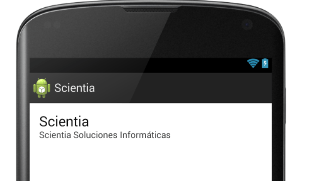
|  |  |
| --- | --- |
| 3 | android:layout\_height="wrap\_content" |
| 4 | android:textAppearance="?android:attr/textAppearanceSmall" | |

|  |  |  |
| --- | --- | --- |
| 5 | android:text="Scientia Soluciones Informáticas" | |
| 6 | android:id="@+id/textView2" |

|  |  |
| --- | --- |
| 7 | android:layout\_below="@+id/textView" |
| 8 | android:layout\_alignParentLeft="true" | |

|  |  |
| --- | --- |
| 9 | android:layout\_alignParentStart="true" /> |

Una vez hecho esto la aplicación debería haber quedado algo así:



Preview de la aplicación en Android Studio