



[Cod.: CC Curso: Programación en Dispositivos Móviles]

[Tema: Ejecutando nuestra primera aplicación]

[Prof.: José Manuel Castillo Cara]

Laboratorio dirigido 3

Ejecutando nuestra primera aplicación

Instrucciones:

1. Fecha de entrega será antes del **domingo a las 23:59**.
2. El formato de entrega será *pdf*. Utilizar letra clara y adecuada a un documento técnico con tamaño 12 y márgenes superior e inferior de 3 cm y laterales de 2,5 cm.
3. El laboratorio tendrá una puntuación sobre 20.
4. En el solucionario se deberá copiar los enunciados y dejarlos en negrita.
5. La primera hoja será para la portada que se especificará, el número de laboratorio, nombre y apellidos de los integrantes, nombre de la asignatura y el escudo de la *UNI*.
6. La segunda página será el índice. Donde se deberá tener en cuenta la página de cada Actividad.
7. Las citas y extracciones realizadas de Internet se deberán especificar. **No se corregirá el laboratorio en caso de copiar y pegar fragmentos sin especificar.**
8. Se corregirá la claridad y exactitud de la pregunta, en ningún caso se expondrán fundamentos no preguntados (mucho menos floro :-) de relleno), además de la claridad del documento.
9. En las actividades realice capturas de pantalla para verificar su autoría.



Objetivos

Antes de comenzar a ver los elementos framework de aplicación de Android para empezar a programar, vamos a familiarizarnos con nuestro entorno de programación Android Studio. Este último punto tiene como propósito entender nuestro entorno antes de ver una estructura de proyecto real ejecutando antes una primera aplicación y ejecutándola en nuestra máquina virtual.

Por tanto podemos resumir los siguientes objetivos:

- Saber crear un proyecto Android vacío.
- Saber elegir las versiones de Android para el proyecto.
- Conocer la estructura básica de un proyecto Android.
- Saber ejecutar una aplicación Android.

Contenidos

- Primera aplicación.
- Componentes de nuestra aplicación.
- Acelerador Intel.
- La clase Activity.
- Depuración y pruebas con Android Studio.

1. Primera aplicación

Para crear nuestra primera aplicación tenemos que arrancar el entorno de programación.



Antes de ver en detalle recordemos como era la estructura de un proyecto en Android Studio:

- Carpeta *src*: Archivo de código fuente, con el formato *.java*, y actividades, ubicadas en el paquete java definido.
- Carpeta *res*: Almacena la mayor parte de recursos necesarios en nuestra aplicación: imágenes, estilos, audios, etc. Contiene las subcarpetas:
 1. *Drawable*: archivos de imágenes renderizados por la capa de interfaz de usuario. Son cuatro carpetas, cada una con gráficos a diferentes resoluciones: *hdpi*, *ldpi*, *mldpi*, *xldpi*.
 2. *Layout*: contiene el archivo *activity_main.xml*, que define la interfaz de usuario y se crea por defecto con cada nuevo proyecto.
 3. *Menu*: contiene archivos *XML* de los menús.
 4. *Values*: contiene ficheros *XML* con parejas de constantes y sus respectivos valores. Por ejemplo, el archivo *strings.xml* contiene constantes de cadena con sus valores.
- *AndroidManifest*: Es un archivo muy importante que contiene detalles como permisos para un correcto funcionamiento, niveles de API soportados o cómo se empaquetará la aplicación.

Actividad 1

1. Crea un nuevo proyecto con los conceptos vistos en la teoría.
 1. Teclea “PrimeraAplicacion” en el campo *Application Name*.
Android Studio y verifica que se crea una carpeta con este nombre.

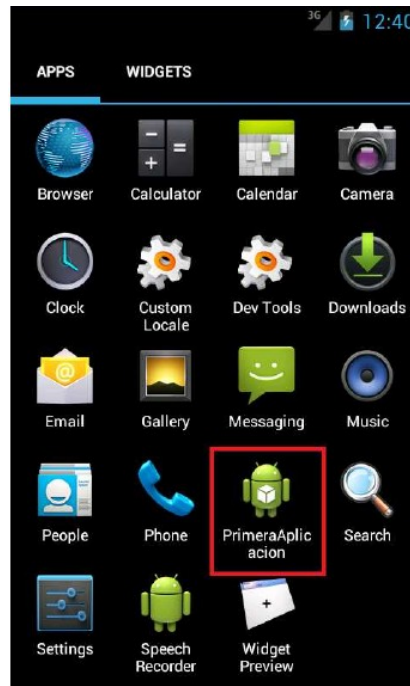


2. Elige el nombre del paquete “*Package Name*” siguiendo el estilo de Java conocido como convenio de dominio inverso:

com.example.manwest.primeraplicacion

En primer lugar, el nombre de dominio garantiza que los nombres de los paquetes sean únicos evitando colisiones de nombres entre organizaciones. Además, este convenio garantiza que los paquetes dentro de una misma organización quedan bien estructurados. Hay que tener en cuenta que cada ‘.’ del nombre del paquete se traducirá en un directorio. De este modo, los directorios se organizarán de más general (‘com’ en nuestro caso) a más particular (‘manwest’).

3. Elige un valor para *Minimum SDK*. Antes de darle a “Next” haz click en “Help me choose”. Explica que ayuda tan importante nos dá.
 4. En la siguientes pantalla se pide los nombres para nuestro proyecto. Apuntar cada nombre.
 5. Le damos a finalizar y vemos nuestro proyecto. Según los diferentes nombres puestos anteriormente localizarlos en nuestro proyecto.
2. Ahora ya puedes ejecutar la aplicación en el emulador que creaste en la tercer unidad. Para ello, selecciona tu proyecto “PrimeraAplicacion” con el cursor en el “*Package Explorer*” (la ventana de la izquierda) y pulsa el icono. Aparecerá el siguiente diálogo para que selecciones un la máquina virtual.



2. Componentes de nuestra aplicación

Una vez que vemos que “PrimeraAplicación” se ejecuta correctamente vamos a ver la estructura de componentes que tenemos en ella identificándolos dentro de Android Studio.

Actividad 2

1. Antes de ver los componentes haga un pequeño resumen de la estructura de proyecto que vimos en teoría con “PrimeraAplicacion”. Seguramente tengas la visión de *Packages* cámbiela a *Android* para verlos de forma muy resumida los ficheros.
2. *AndroidManifest.xml*: En la teoría hemos visto que este fichero identifica el paquete principal y las actividades. Ábralo en el editor y entienda su contenido.



3. Actividades: Desde la Actividad principal se carga el *Layout* o el diseño.

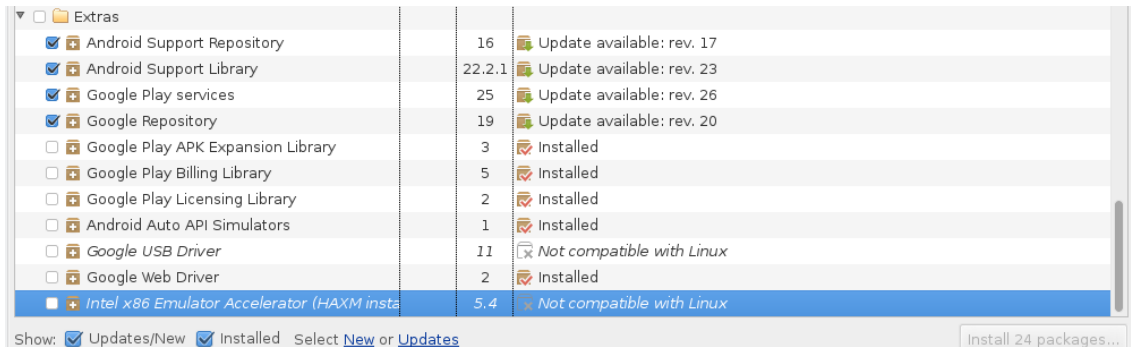
1. Lo primero que vemos en nuestra clase principal Java es que hay una herencia. Explique esa clase padre e identifique su paquete de importación explicándolo también.
2. Un aspecto importante es el método *setContentView*. Este método dibuja el fichero *xml* en pantalla. Identificar dónde se encuentra este fichero en nuestro proyecto. Fijarse que en esa dirección se encontrarán todas las interfaces gráficas.
3. Explicar los otros métodos que nos encontramos en dicho fichero.
4. *Layout*: en esta carpeta nos encontramos los *xml* que definen nuestra interfaz gráfica de la aplicación.
 1. En nuestra aplicación tenemos un *Relative Layout*. Explicar brevemente dos *layout* más.
 2. También hemos aprendido que toda variable *string* se encuentra en su fichero. Verificar la ruta de estas variables.
 3. Aunque ya lo veremos más adelante explicar los otros elementos de este fichero que no se explicaron en teoría.
5. *Strings*: otro aspecto a entender es el fichero *strings.xml*. Cambia la variable llamada en nuestro *xml Activity* y ejecute la aplicación. Comenta los cambios.

3. Acelerador de Intel

Un aspecto importante a la hora mejorar la velocidad de arranque de nuestra máquina virtual es la instalación del paquete en extras de “*Intel x86*”



Emulator Accelerator". Este paquete solamente está disponible para Mac y Windows, Linux no lo necesita. Una vez descargado hay que instalarlo.



| Package Name | Version | Status |
|---|---------|---------------------------|
| Android Support Repository | 16 | Update available: rev. 17 |
| Android Support Library | 22.2.1 | Update available: rev. 23 |
| Google Play services | 25 | Update available: rev. 26 |
| Google Repository | 19 | Update available: rev. 20 |
| Google Play APK Expansion Library | 3 | Installed |
| Google Play Billing Library | 5 | Installed |
| Google Play Licensing Library | 2 | Installed |
| Android Auto API Simulators | 1 | Installed |
| Google USB Driver | 11 | Not compatible with Linux |
| Google Web Driver | 2 | Installed |
| Intel x86 Emulator Accelerator (HAXM installer) | 5.4 | Not compatible with Linux |

Show: ☒ Updates/New ☒ Installed Select [New](#) or [Updates](#) [Install 24 packages...](#)

Actividad 3

1. Abra la carpeta física Android → SDK → Extras → Intel y dentro podrá observar el archivo. Instálelo y verá que mejora notablemente el arranque.

4. La clase Activity

Vamos a seguir viendo una primeras funcionalidades y asentar conocimientos vistos sobre esta clase. Para ello seguimos trabajando en el mismo proyecto.

Para ello vamos ver un avance de Instancias en Android. Aunque vamos a verlo más adelante con más profundidad recordad que un *Intent* sirve par pasar de una actividad a otra.

Actividad 4

1. A continuación vamos crear una segunda actividad. Para crearla creamos una clase pública que herede de *AppCompatActivity* como hemos estado estudiando hasta ahora. Hay que crearla donde está la otra actividad, es decir, src → main → java → com.XXX.



2. Llamamos a nuestra clase “*SegundaActividad*”.
3. Para que sea considerada Actividad deberemos hacer que herede de *AppCompatActivity*.
4. Acordarse de realizar la importación de la librería correspondiente.
5. Implementar el método *onCreate* copiándolo de la anterior clase.
6. Verificar que nos solicita la importación de la librería “*Bundle*”.
7. Cambiamos el nombre de la interfaz gráfica, llamándose ahora “*activity_segunda_actividad*”. Comprobamos que nos aparece un error y es porque no reconoce el *Layout*, es decir, no existe esta interfaz.
8. Copiamos y pegamos el archivo *xml* de *Layout* para crear nuestra nueva interfaz.
9. En el archivo “*activity_segunda_actividad.xml*” cambiamos “*tools:context*” por “*.SegundaActividad*”.
10. Cambiamos el mensaje que muestra el *TextView*. Para ello tendremos que crear una nueva variable *string* y asignarle el nuevo valor en el fichero “*strings.xml*”.
11. Una vez creada nuestra segunda actividad, recordad que se debe de indicar en el archivo “*AndroidManifest.xml*”.

1. Escribir el siguiente código dentro de la etiqueta “*application*”.

```
<activity
    android:name=".SegundaActividad"
    android:label="@string/seg_actividad" >
</activity>
```

2. Vemos que nos aparece un error, eso es porque no hemos creado dicho *string*. Procedemos a crearlo con, por ejemplo, el valor “*Mi segunda Actividad*”



`<string name="seg_actividad">Mi Segunda Actividad</string>`

12. En la pestaña “*Design*” creamos un *button* en el fichero “*activity_primera_aplicación.xml*” que es el *xml* que teníamos en un primer momento. Para ello borramos el *TextView* de “*Hello World!*”.

13. Vemos como en la pestaña *Text* se ha creado el código de `<Button>`. Además nos aparece una advertencia que es que tenemos que definir dicho *string*. En esta ocasión vamos a cambiarlo a través de “*Properties*” en la vista “*Design*”.

1. Pulsamos sobre *Button* y buscamos la propiedad “*text*”.
2. Hacemos click en los puntos suspensivos y luego en *New Resource* → *New string value...*
3. Le indicamos el nombre e identificador. En mi caso he puesto “*Pulsame*”.
4. En la propiedad “*onClick*” vamos a poner nombre al método nuevo al hacer click sobre “*Púlsame*”. En mi caso le he llamado “*onClick*”
5. En el fichero java “*PrimeraAplicacion.java*” añadimos el siguiente método:

```
public void onClick(View view){  
    Intent intent = new Intent(this, SegundaActividad.class);  
    startActivity(intent);  
    finish();  
}
```

6. Del código anterior explique cada línea su funcionalidad.
7. Importamos las clases que nos sugiere.



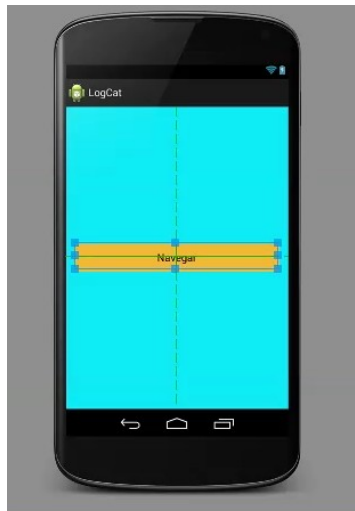
5. Depuración y pruebas con Android Studio

IMPORTANTE: Realizar esta actividad después del apartado práctica.

5.1. Depuración con LogCat

Un aspecto muy importante en la programación es saber manejar correctamente la depuración de nuestro código. Android Studio nos ofrece LogCat. En la siguiente actividad vamos a trabajar este aspecto.

Lo primero que creamos es una aplicación con este aspecto como ya hemos estado viendo.



Actividad 5

1. Una vez creado nuestra interfaz, le damos el valor “conductaIncorrecta” a la propiedad *onClick* de *Button*.
2. Este método no existe de momento en el código Java.
3. Hacemos doble click en la vista diseño de nuestro Button y le damos el valor id “botonQueFalla”.



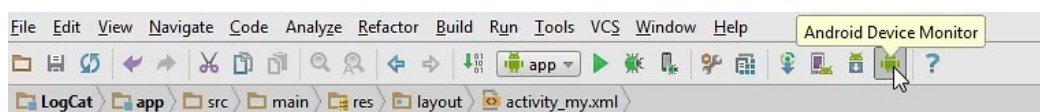
4. Le damos a *Run* y vemos que se compiló correctamente aunque da error al pulsar.
5. En la parte posterior abrimos el apartado “6: Android” y se activa la herramienta LogCat.
6. Volvemos a hacer click y vemos lo que nos dice LogCat. Podemos observar que nos muestra que hubo un “FATAL ERROR”. Transcribe hasta aquí lo que nos muestra.
7. Cambiamos el valor de *onClick* por “informacion” donde en el código java tendrá el siguiente código.

```
Public void informacion (View view){  
    Log.i(“mensaje”, “Hola amigo”);  
}
```

8. Explicar la ejecución ahora y sobre todo los distintos niveles para filtrar información que tenemos.

5.2. Depuración con el monitor de Android

Otra herramienta potente para depurar nuestras aplicaciones es el monitor de Android que se encuentra donde vemos en la imagen.



Permite todos los elementos que se pueden ver en nuestros dispositivos de pruebas. Por tanto vemos los diferentes elementos que tiene.

Actividad 6

1. Ejecute la aplicación anterior de manera que esté su máquina virtual activada.



2. Explique la importancia del icono para revisar toda la estructura de archivos.
3. Explique la importancia del icono para revisar todos los hilos que se están ejecutando.
4. Explique la importancia del icono para revisar el uso de memoria.
5. Explique la importancia del icono para revisar el uso de Internet.
6. Explique la importancia del icono para revisar la carga de trabajo.
7. Realice una llamada telefónica al equipo virtual.
8. Envíe un mensaje al dispositivo.

5.3. Publicación de una aplicación

Vamos a ver como es el proceso de publicación de una aplicación, por lo que no cerramos dicha aplicación de este punto 5.

Actividad 7

1. Hacemos click derecho en la carpeta “build” y pulsamos “Show in Explorer”. Vemos que muestra los archivos.
2. Abrimos *build* → *output* que es la carpeta que contiene los archivos que se están ejecutando en el emulador.
3. Vemos una carpeta “apk” que es el ejecutable de Android. Explique los archivos que nos aparecen.
4. Para publicar en la tienda de Android deberá estar firmada nuestra aplicación.
5. En la Barra de Herramientas pulsamos “Build” y luego “Generate Signed APK...” que vemos que nos produce una aplicación certificada.



6. Par ello creamos un nuevo certificado. Rellenes los datos de dicho archivo.
7. Como estoy preparando mi aplicación para ser publicada voy a elegir *release*. Explique los conceptos que aparecen.
8. Luego de unos minutos vemos que Gradle compila nuestro paquete con nuestro certificado.
9. Ahora entramos en la carpeta *app* de nuestros archivos y vemos el archivo "*app-release.apk*" que está listo para publicar.
10. Para publicar debemos entrar en la página de publicadores de Google y entrar en la sección "*Distribute*" y luego *Developer Console*.
11. En este proceso recordemos que hay que pagar 25US\$ que todavía no vamos a pagar.
12. Una vez dentro se subiría nuestro archivo *apk* que acabamos de generar.

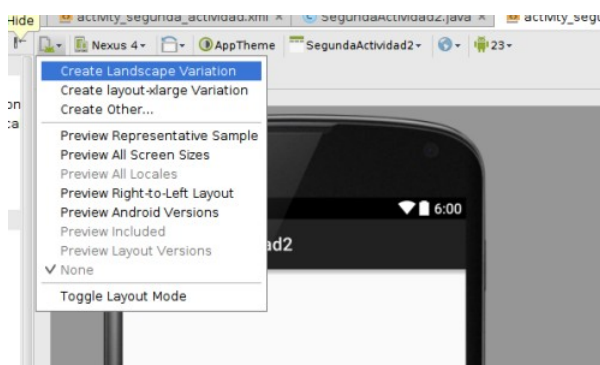


Práctica

1. En la pestaña propiedades de nuestra vista diseño:
 1. Cambiar el color de fondo de “Púlsame”. Exponga la propiedad y como funcionan sus valores.
 2. En la actividad del texto hacemos click en *Properties* y buscamos *size*. De esta forma podemos buscar propiedades que tiene esa palabra. Cambiamos el tamaño del texto con la propiedad adecuada
 3. Cambiamos también el color del texto.
 4. Para cambiar el color de fondo tendremos que hacer click en nuestro *Layout* de *Component Tree* y en *Properties* buscamos la propiedad adecuada. Cambiamos el color.
2. En la vista diseño también podemos ver la apariencia de nuestro dispositivo en vertical y horizontal, que se encuentra entre el elemento que selecciona el dispositivo y “*AppTheme*”.
 1. También podemos ver como quedan en múltiples dispositivos, navegar por esta opción.
 2. Donde está el icono Android en la misma pestaña podemos ver como quedaría nuestra aplicación para diferentes versiones. Inspeccionamos esta opción también.
3. En la última actividad hemos visto como crear una actividad con una instancia. Teniendo esta actividad como referencia, realice los siguientes pasos:
 1. Cree un proyecto nuevo.



2. En la actividad principal cambie el fondo de pantalla y alguna propiedad más.
3. Haz click derecho sobre la carpeta *Layout* y cree una nueva actividad. Fijarse en la carpeta Java y en el archivo “*AndroidManifest.xml*”. ¿Que ha ocurrido?
4. Volver a realizar una Instancia pero cambiando propiedades a los elementos de cada actividad.
4. Otro aspecto importante es establecer diseños cuando la aplicación está en vertical y horizontal. Buscamos el icono que se especifica en la imagen y elegimos “*Create Landscape Variation*”



1. Vemos como ha creado un nuevo “*Layout*”. ¿Donde se ha creado dicho “*Layout*”? Buscar la carpeta “*layout-land*”, de landscape. También podemos observar que los xml se llaman igual, Android se encargará de escogerlo según la rotación del equipo,
2. Damos a la propiedad Background y cambiamos el fondo en, por ejemplo, System → holo_purple.
3. Con esto vemos que tiene una nueva variante para mi orientación de pantalla.



4. En la misma vista diseño rotamos la vista previa. ¿Que nos muestra?



Laboratorio 3: ejecutando nuestra primera aplicación – Programación de Dispositivos Móviles - Escuela Profesional de Ciencias de la Computación - Facultad de Ciencias - Universidad Nacional de Ingeniería por José Manuel Castillo Cara se encuentra bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).