

PRÁCTICA 1: PROGRAMACIÓN PARA MÓVILES ANDROID

Instalación del software de desarrollo

Se deberá instalar el SDK (*Software Development Kit*) de Android disponible en: <http://developer.android.com/sdk/index.html>

Seguir las instrucciones de la instalación <http://developer.android.com/sdk/installing/bundle.html>

Creación de la primera App

Se creará una aplicación muy fácil (*Hello World!!*) para Android usando Android studio. Sigue los pasos indicados en el tutorial de la siguiente dirección: <https://developer.android.com/training/basics/firstapp/creating-project>

Tras marcar las opciones por defecto se crearán una serie de directorios dentro de la carpeta seleccionada.

Un proyecto Android se compone de:

- Un elemento descriptor de la aplicación (el archivo *AndroidManifest.xml*),
- El código fuente en Java y
- Ficheros de recursos.

Los más representativos son:

- Carpeta **src**: contiene el código fuente de la aplicación, clases auxiliares, etc.
- Carpeta **gen**: contiene el código generado automáticamente al compilar nuestra aplicación. Nunca cambiar este código manualmente.
Contiene dos ficheros o clases java:
 - *BuildConfig.java*: Define la constante Debug para que desde Java puedas saber si tu aplicación está en fase de desarrollo.
 - *R.java*: Define una clase que asocia los recursos de la aplicación con identificadores. De esta forma los recursos podrán ser accedidos desde Java.

-
- Carpeta **res**: de ella cuelgan los subdirectorios y ficheros con los recursos que usará la aplicación. Entre ellos se encuentra el fichero `/res/layout/main.xml` que describe la pantalla principal de la interfaz gráfica.
 - Carpeta **bin**: contiene los elementos compilados de la aplicación, como el .apk, fichero comprimido que contiene la aplicación final, lista para ser instalada.
 - Archivo **AndroidManifest.xml**: contiene información esencial acerca de la aplicación, que será requerida por el sistema antes de su ejecución. En él se declaran los permisos requeridos por la aplicación, el nivel mínimo de la API de Android requerido, se enumeran las bibliotecas vinculadas a la aplicación, la versión de la aplicación, etc.

Ejecución de la App creada

Sigue los pasos indicados en la siguiente lección del tutorial de Android: <http://developer.android.com/training/basics/firstapp/running-app.html>

La ejecución de la App se puede realizar de dos modos: en caso de disponer de un móvil Android (y cable de conexión USB) se podrá ejecutar la aplicación directamente en él. En caso de no disponer del mismo, se podrán ejecutar en un emulador de dispositivos (con AVD Manager):

- 1) Sobre un móvil que soporte Android, conectado al PC/Portátil y con la opción de desarrolladores activada en el móvil, generalmente en el menú: *Settings > Applications > Development*.
- 2) Sobre un emulador, habiendo creado previamente un dispositivo móvil virtual. Para ello hay que lanzar el AVD (*Android Virtual Device*) Manager con el botón correspondiente de la barra de tareas.

Modificaciones en el Interfaz Gráfico (UI)

La primera modificación que se puede hacer a la aplicación anterior se centra en el interfaz gráfico de usuario (UI). Para ello, utilizar como referencia los contenidos sobre el uso del UI y de las cadenas de texto y botones en la siguiente lección del tutorial:

<http://developer.android.com/training/basics/firstapp/building-ui.html>

En este punto se deberá incluir en la aplicación un botón con el texto “PLAY”. El modo más intuitivo es para diseñar el interfaz gráfico es el empleo de la *paleta*, aunque puede comprobar o editar los cambios en el código XML asociado (fichero `/res/layout/main.xml`).

Reproducción de contenido multimedia

Hasta este punto, al pulsar el botón no ocurre nada. Para iniciar una nueva tarea al pulsarlo hay que asociarle una tarea al objeto botón. Este proceso se detalla en la siguiente parte del tutorial, dentro de una aplicación algo más compleja:

<http://developer.android.com/training/basics/firstapp/starting-activity.html>

Para esta práctica, utilice el siguiente código de ejemplo, en el cual se implementa la interfaz `OnClickListener` para detectar eventos del botón (`PlayButton`). Recuerde que en la clase `R.java` puede acceder a los recursos de la aplicación, como por ejemplo, al botón, de la siguiente forma: `R.id.PlayButton`. Véase también cómo la función `onClick` se encarga de coger la dirección de un fichero de video almacenado en la memoria externa del teléfono (SD) y ejecutar la actividad, por lo que este código no funcionará correctamente en el emulador. Si lo desea, el video de ejemplo se encuentra ubicado en http://www.mobvcasting.com/android/video/Test_Movie.m4v

```
package com.apress.proandroidmedia.ch9.videoplayerintent;

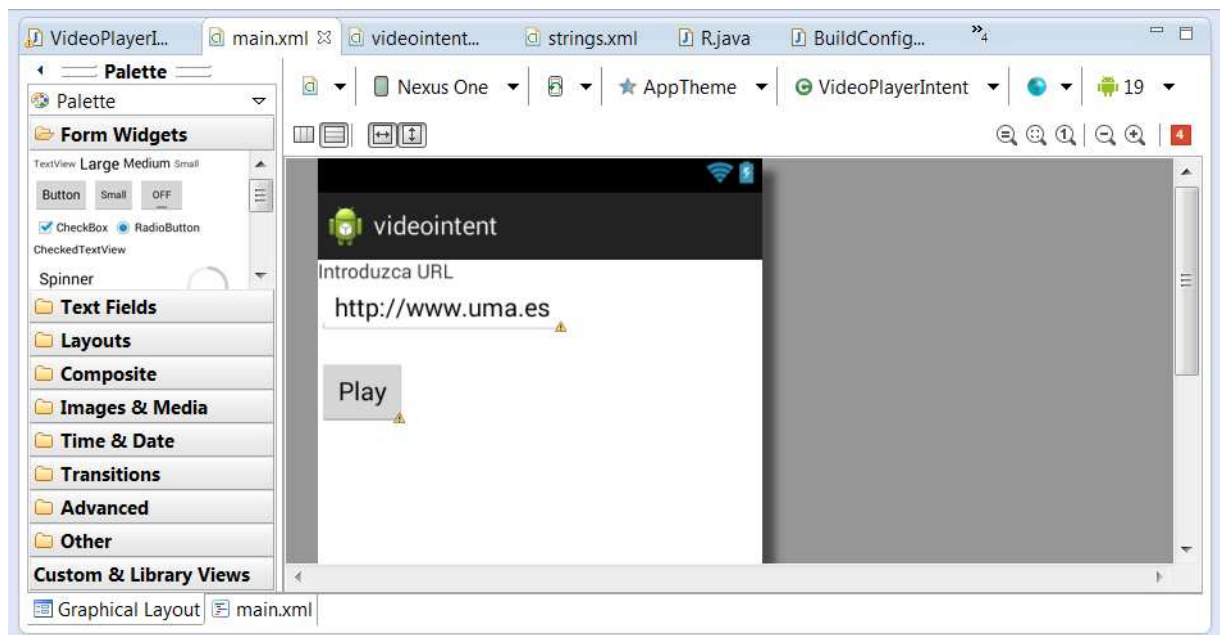
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class VideoPlayerIntent extends Activity implements OnClickListener {
    Button playButton;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        playButton = (Button) this.findViewById(R.id.PlayButton);
        playButton.setOnClickListener(this);
    }

    public void onClick(View v) {
        Intent intent = new Intent(android.content.Intent.ACTION_VIEW);
        Uri data = Uri.parse(Environment.getExternalStorageDirectory().getPath() +
            "/Test_Movie.m4v");
        intent.setDataAndType(data, "video/mp4");
        startActivity(intent);
    }
}
```

Modifique el código anterior y el interfaz gráfico para la aplicación pueda abrir un video almacenado en Internet. Para ello, se deberá añadir a la UI un campo editable para introducir una URL cualquiera, como se muestra en la figura.



NOTA: Para que la aplicación tenga permisos para acceder a Internet, es necesario añadir dicho permiso en el fichero `AndroidManifest.xml`:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Con la aplicación anterior, pruebe a acceder a un contenido de video, accediendo a la URL: http://www.mobvcasting.com/android/video/Test_Movie.m4v

Es muy probable que ciertos formatos de fichero no puedan reproducirse con esta clase, lo cual depende del tipo de codificación, versión de Android, etc. Los ejemplos detallados en esta práctica han sido probados satisfactoriamente con Android 4.3.

Acceso a páginas web

Pruebe a ejecutar la aplicación (en el emulador y/o terminal real) para acceder a una página web. Debes especificar el protocolo en la URL, por ej: <http://www.uma.es>. Como observará, la aplicación no es capaz de acceder a dicha página debido a que el tipo MIME definido es "video/mp4". Modifique dicho tipo para que se pueda acceder a páginas web.

Una lista bastante completa de tipos MIME puede encontrarse en la siguiente dirección: <http://reference.sitepoint.com/html/mime-types-full>

Haga los cambios necesarios en la aplicación para que no haya que modificar el código fuente dependiendo de si se quiere acceder a una página web o a un contenido multimedia. Para ello, puede utilizar un selector, detección automática o lo que se le ocurra.

