# APLICACIONES PARA MÓVILES II

Uziel Pérez Campillo 15610695

## 1.2 Multimedia

Vamos a desarrollar una aplicación cuya actividad principal será mostrarnos los archivos multimedia(audios y video), en una lista personalizada, incluiremos también un buscador que nos permita buscar contenido por nombre, artista o genero. Al pulsar cualquier elemento de la lista nos abrirá una nueva actividad en la cual se reproducirá el contenido multimedia seleccionado.

#### 1

Primero crearemos un nuevo proyecto, en mi caso lo llame 1.2Multimedia con la versión API 26.

En el AndroidManifest agregamos el siguiente permiso :

<uses-permission android:name="android.permission.WAKE\_LOCK" />

para que se siga reproduciendo el audio cuando el dispositivo este inactivo.

#### 2

En nuestro layout principal modificamos el layout a LinearLayout y la orientación a vertical. Y agregamos los siguientes elementos:

<**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
  
 <**EditText  
 android:id="@+id/editTxtBuscar"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="2"  
 android:ems="10"  
 android:hint="Buscar"  
 android:inputType="textPersonName"** />  
  
 <**Button  
 android:id="@+id/btnBuscar"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Buscar"** />  
</**LinearLayout**>  
  
<**ListView  
 android:id="@+id/listViewArchivos"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />

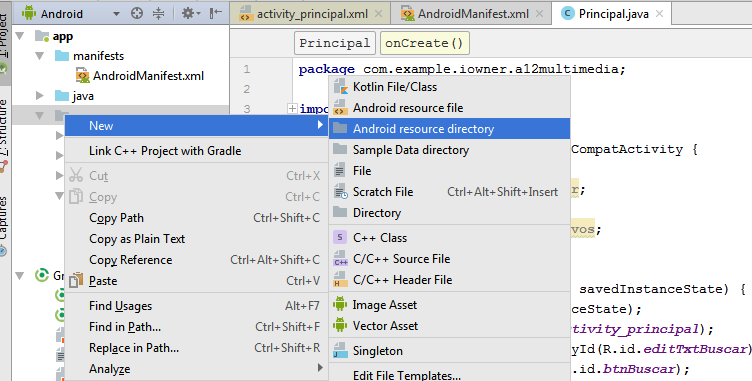
#### 3

Ahora vinculamos los elementos del layout en el código:

**private** EditText **editTxtGuardar**;  
**private** Button **btnBuscar**;  
**private** ListView **listViewArchivos**;  
  
@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_principal***);  
 **editTxtGuardar** = findViewById(R.id.***editTxtBuscar***);  
 **btnBuscar** = findViewById(R.id.***btnBuscar***);  
 **listViewArchivos** = findViewById(R.id.***listViewArchivos***);  
}

#### 4

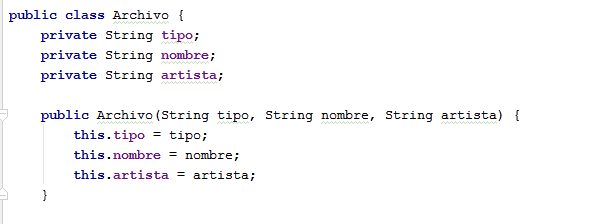
Agregamos un nuevo directorio a "res" y lo llamamos "raw"



en el vamos a guardar nuestros archivos .mp3 y .mp4(es solo necesario que el nombre el archivo solo contenga letras minúsculas y que no tenga espacios ni caracteres especiales)

#### 5

Creamos una clase Archivo con los siguientes atributos:



(en la imagen falta el atributo "int id" igual en el constructor)y agregamos sus respectivos getters y setters

#### 6

Creamos un arreglo con nuestros archivos multimedia:

Archivo[] **archivos** = { **new** Archivo(**"Musica"**,**"Cotton fields"**,**"Credeence"**,R.raw.***cottonfields***),  
 **new** Archivo(**"Musica"**,**"Come on feel the noise"**,**"Credeence"**,R.raw.***comeonfeel***),  
 **new** Archivo(**"Musica"**,**"One"**,**"Apocalyptica"**,R.raw.***cne***),  
 **new** Archivo(**"Video"**,**"Symphony 40"**,**"Metallica"**,R.raw.***symphony***)  
};

#### 7

Creamos un nuevo layout con lo siguiente en mi caso lo llame archivo.xml (este layout es para cada elemento de la lista)

<**LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:orientation="vertical"  
 android:layout\_height="match\_parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 tools:layout\_editor\_absoluteX="8dp"  
 tools:layout\_editor\_absoluteY="8dp"**>  
  
 <**TextView  
 android:id="@+id/txtNombre"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textColor="@android:color/black"  
 android:textSize="30sp"** />  
  
 <**TextView  
 android:id="@+id/textArtista"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textColor="@android:color/background\_dark"  
 android:textSize="18sp"** />

<**TextView  
 android:id="@+id/textTipo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"** />

</**LinearLayout**>  
</**LinearLayout**>

#### 8

Creamos una clase MultimediaAdapter y heredamos de la clase ArrayAdapter (esta clase nos pasa el array de Archivos y nos lo muestra en la lista):

**class** MultimediaAdapter **extends** ArrayAdapter<Archivo> {  
 Context **cContexto**;  
 **int iLayout**;  
 Archivo[] **archivos**;  
 @NonNull  
 @Override  
 **public** View getView(**int** position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 TextView txtTipo, txtNombre, txtArtista;  
  
 View vFila = convertView;  
 **if**(vFila == **null**){*//no existe fila creeamos  
 //PARA CREAR LA VISTA USAMOS LAYOUTINFLATER* LayoutInflater liVista = ((Activity)**cContexto**).getLayoutInflater();  
 vFila = liVista.inflate(**iLayout**,parent,**false**);  
  
 }  
 *//VINCULAR WIDGETS* txtTipo = vFila.findViewById(R.id.***txtTipo***);  
 txtNombre = vFila.findViewById(R.id.***txtNombre***);  
 txtArtista = vFila.findViewById(R.id.***txtArtista***);  
  
 Archivo aActual = **archivos**[position];  
  
 txtTipo.setText(aActual.getTipo());  
 txtNombre.setText(aActual.getNombre());  
 txtArtista.setText(aActual.getArtista());  
 **return** vFila;  
 }  
  
 **public** MultimediaAdapter(@NonNull Context context, **int** resource, @NonNull Archivo[] objects) {  
 **super**(context, resource, objects);  
 **cContexto** = context;  
 **iLayout** = resource;  
 **archivos** = objects;  
 }  
  
}

#### 9

Ahora en el MainActivity(en mi caso Principal) asignamos el adaptador a la lista:

listViewArchivos.setAdapter(**new** MultimediaAdapter(**this**,R.layout.***archivo***,**archivos**));

#### 10

Creamos dos Actividades nuevas: ReproducirMusica y ReproducirVideo:

En el layout de ReproducirMusica agregamos lo siguiente:

<**ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="435dp"  
 android:layout\_weight="5"  
 app:srcCompat="@mipmap/ic\_launcher"** />  
  
<**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="5"  
 android:orientation="horizontal"**>  
  
 <**SeekBar  
 android:id="@+id/seekBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"** />  
</**LinearLayout**>  
  
<**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:orientation="horizontal"**>  
  
 <**ImageButton  
 android:id="@+id/imageButtonpPause"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/ic\_media\_pause"** />  
  
 <**ImageButton  
 android:id="@+id/imageButtonPlay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/ic\_media\_play"** />  
</**LinearLayout**>

Y vinculamos en el código de la actividad

Y en el de ReproducirVideo lo siguiente:

<**FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**VideoView  
 android:id="@+id/videoView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"**/>  
</**FrameLayout**>

Y vinculamos en el codigo de la actividad

#### 11

Creamos dos intentos: uno para ReproducirMusica y otro para Reproducir Video:

Intent **iArchivo**,**iVideo**;//Variables globales

iArchivo = **new** Intent(**this**,ReproduccionArchivo.**class**);//Inicializadas en el onCreate();  
**iVideo** = **new** Intent(**this**,ReproducirVideo.**class**);

#### 12

Le agregamos a la lista un escuchador para hacer algo cuando se seleccione algo de la lista:

listViewArchivos.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  
 **if**(**archivos**[i].getTipo().equals(**"Musica"**)){  
 **iArchivo**.putExtra(**"tipo"**,**archivos**[i].getTipo());  
 **iArchivo**.putExtra(**"nombre"**,**archivos**[i].getNombre());  
 **iArchivo**.putExtra(**"artista"**,**archivos**[i].getArtista());  
 **iArchivo**.putExtra(**"id"**,**archivos**[i].getId());  
 startActivity(**iArchivo**);  
 }**else**{  
 **iVideo**.putExtra(**"id"**,**archivos**[i].getId());  
 startActivity(**iVideo**);  
 }  
 }  
});

En este caso comparamos el elemento "tipo" para saber si llamamos a la actividad de Música o a la de Video.

#### 13

Ahora en el código de la actividad ReproducirMusica agregamos:

Para tomar los datos del elemento elegido:

iDatos = getIntent();  
**int** idC = iDatos.getIntExtra(**"id"**,0);

Creamos el objeto MediaPlayer:

cancion = MediaPlayer.*create*(**this**, idC);

E iniciamos la cancion.

cancion.start();

Creamos un clase que herede de Runnable la cual usaremos para estar escuchando al SeekBar

**class** MiHilo **implements** Runnable{  
 @Override  
 **public void** run() {  
 **while**(**true**){  
 **try** {  
 **int** mCurrentPosition = **cancion**.getCurrentPosition();  
 Message msg = **mHandler**.obtainMessage(1,mCurrentPosition);  
 **mHandler**.sendMessage(msg);  
 Thread.*sleep*(1000);  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 **return**;  
 }  
 }  
 }  
}

Y creamos un objeto de la clase Handler para poder modificar la interfaz grafica

Handler **mHandler** = **new** Handler(){  
 @Override//Sobre escribimos el metodo  
 **public void** handleMessage(Message msg) {  
 **super**.handleMessage(msg);  
 **if**(msg.**what**==1){  
 **int** i = (**int**)msg.**obj**;  
 **mSeekBar**.setProgress(i);//modificamos el seek bar  
 }  
 }  
};

Guardamos la duración de la canción y la asignamos al máximo para el seek bar, creamos el hilo y lo iniciaos.

**int** mFileDuration = **cancion**.getDuration();  
  
**mSeekBar**.setMax(mFileDuration);  
MiHilo miHilo = **new** MiHilo();  
**tHilo** = **new** Thread(miHilo);  
**tHilo**.start();

Utilizamos el método onProgressChanged() para modificar la canción según el seekBar

**public void** onProgressChanged(SeekBar seekBar, **int** i, **boolean** b) {  
 **if**(**cancion** != **null** && b){  
 **cancion**.seekTo(i);  
 }  
}

Sobre escribimos el metodo onDestroy() para terminar la cacion

**protected void** onDestroy() {  
 **super**.onDestroy();  
 **tHilo**.interrupt();  
 **cancion**.stop();  
 **cancion**.release();  
}

Y agremos los onClick() a los botones pause y play:

play.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **cancion**.start();  
 }  
});  
**pause**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **cancion**.pause();

#### 14

En el código de la actividad ReproducirVideo solo agregamos lo siguiente:

controller = **new** MediaController(**this**);  
**video**.setMediaController(controller);  
**video**.setVideoURI(Uri.*parse*(**"android.resource://"** + getPackageName() + **"/"** + iDatos.getIntExtra(**"id"**,0)));  
**video**.requestFocus();  
**video**.start();

#### 15 Listo