



**Asignatura:** Application Development for Mobile Devices.

**Tema:** Reproducción de Audio y Video.

## INTRODUCCIÓN.

Los formatos disponibles para la reproducción de aplicaciones multimedia se encuentran en la siguiente dirección, en la sección **Core Media Format**:

<http://developer.android.com/guide/appendix/media-formats.html>

Para el sistema, la lectura de un archivo de sonido solamente es un recurso. Los sonidos se deben almacenar en la carpeta **/res/raw/** del proyecto Android.

Android suministra la clase `MediaPlayer` para la lectura de los archivos multimedia (sonidos y videos). Además, del constructor predeterminado, `MediaPlayer` utiliza el método estático `create()`, que devuelve una instancia de `MediaPlayer` lista para su uso:

```
static MediaPlayer create(Context context, int resourceid)
```

El primer parámetro representa el contexto de ejecución de la aplicación y el segundo parámetro es el identificador del recurso multimedia que se debe recuperar. Por ejemplo:

```
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.misonido);
```

Una vez agregado el archivo **misonido.mp3** a la carpeta **/res/raw/** del proyecto, se invoca con el método `start()` de la clase `MediaPlayer`:

```
mediaPlayer.start();
```

Si se desea reproducir un archivo en streaming, por ejemplo un vídeo de **Youtube**, se puede utilizar el método `setVideoURI()` o reproducir un archivo de vídeo almacenado en la memoria **SDCard** con `setVideoPath()`.

Con el método `setMediaController()` se agrega el objeto `MediaController`, que permite el control de la reproducción del vídeo agregando los botones de pausa, stop, play, por ejemplo. Para iniciar la reproducción del vídeo se utiliza `start()`.

Si quieres reproducir un vídeo de **Youtube**, se debe utilizar la URI que se utiliza en RTSP/3GP, es decir, de la forma:

```
http://m.youtube.com/add_favorite?v=XXXX
```

Donde, **XXXX** es el identificador del video que se muestra al reproducir un vídeo en el navegador:

```
http://www.youtube.com/watch?v=XXXXXX
```

## DESARROLLO.

### Ejemplo 1.

1. Crear un nuevo proyecto en Android Studio.
2. Modificar el archivo `MainActivity.java` con el siguiente código.

```
import android.os.Bundle;
import android.app.Activity;
import android.media.MediaPlayer;
import android.view.View;
import android.widget.Button;
public class MainActivity extends Activity implements View.OnClickListener {
    MediaPlayer mp;
    Button jbn1, jbn2, jbn3, jbn4;
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
```



```
jbn1 = (Button) findViewById(R.id.xbn1);
jbn2 = (Button) findViewById(R.id.xbn2);
jbn3 = (Button) findViewById(R.id.xbn3);
jbn4 = (Button) findViewById(R.id.xbn4);
jbn1.setOnClickListener(this);
jbn2.setOnClickListener(this);
jbn3.setOnClickListener(this);
jbn4.setOnClickListener(this);
}
public void onClick(View v) {
    if (v.getId() == R.id.xbn1) {
        reproducir(R.raw.adele_hello);
    }
    if (v.getId() == R.id.xbn2) {
        reproducir(R.raw.you_got_it);
    }
    if (v.getId() == R.id.xbn3) {
        reproducir(R.raw.ahhh);
    }
    if (v.getId() == R.id.xbn4) {
        reproducir(R.raw.aplauso);
    }
}
void reproducir(int cancion){
    if(mp != null) mp.release();
    mp = MediaPlayer.create(this, cancion);
    mp.seekTo(0);
    mp.start();
}
public void onPause(){
    super.onPause();
    if(mp != null)
        mp.release();
}
}
```

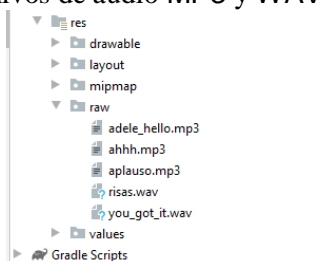
### 3. Modificar el archivo **activity\_main.xml** con el siguiente código.

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/xl11"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <LinearLayout
        android:id="@+id/xl12"
        android:layout_weight="1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
        <Button
            android:id="@+id/xbn1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:gravity="center"
            android:text="Cancion1"/>
        <Button
```



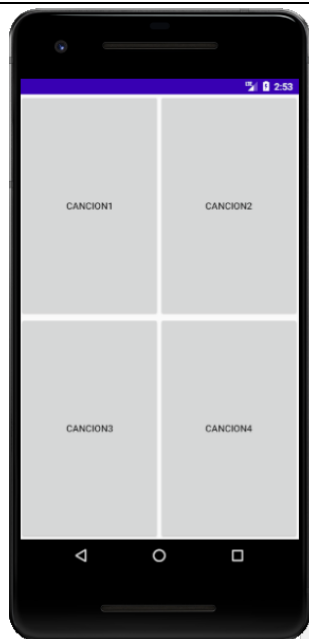
```
        android:id="@+id/xbn2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Cancion2"/>
</LinearLayout>
<LinearLayout
    android:id="@+id/xl13"
    android:layout_weight="1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <Button
        android:id="@+id/xbn3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Cancion3"/>
    <Button
        android:id="@+id/xbn4"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center"
        android:text="Cancion4"/>
</LinearLayout>
</LinearLayout>
```

4. Crear la carpeta `/res/raw`. Agregar los archivos de audio MP3 y WAV a la carpeta `raw` recién creada.



**Figura 1.** La carpeta `raw` y sus archivos multimedia.

5. Ejecutar la aplicación. Se muestra una imagen similar a la siguiente. Al digitar un botón se reproduce el sonido o la melodía correspondiente.



**Figura 2.** Botones de reproducción.

6. Agregar un botón extra para reproducir el quinto archivo `risas.wav` faltante.

## Ejemplo 2.

1. Crear un nuevo proyecto en Android Studio.
2. Modificar el archivo `MainActivity.java` con el siguiente código:

```
import android.app.*;
import android.net.*;
import android.os.*;
import android.view.View;
import android.view.View.*;
import android.widget.*;
public class MainActivity extends Activity implements OnClickListener {
    Button jbn;
    VideoView vvw;
    Uri uri;
    MediaController mcr;
    @Override
    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        jbn = (Button) findViewById(R.id.xbn); jbn.setOnClickListener(this);
    }
    public void onClick(View v){
        vvw = (VideoView) findViewById(R.id.xvv1);
        uri = Uri.parse("android.resource://com.example.myapplication/"+
R.raw.this_is_love);
        mcr = new MediaController(this);
        vvw.setMediaController(mcr);
        vvw.setVideoURI(uri);
        vvw.start();
    }
}
```



```
}
```

3. Modificar el archivo `activity_main.xml` con el siguiente código.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:orientation="vertical" >
    <Button
        android:id="@+id/xbn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="REPRODUCIR"/>
    <VideoView
        android:id="@+id/xvv1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

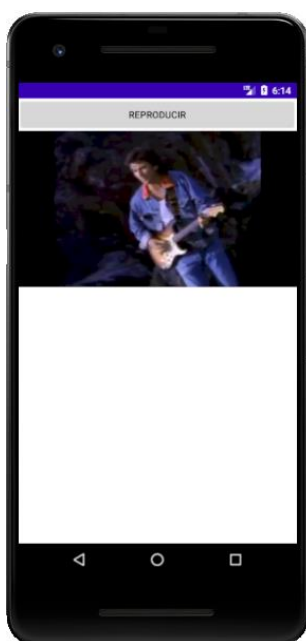
4. Modificar el archivo `strings.xml` con el siguiente código.

```
<resources>
    <string name="app_name">Multimedia</string>
    <string name="reproducir">REPRODUCIR</string>
</resources>
```

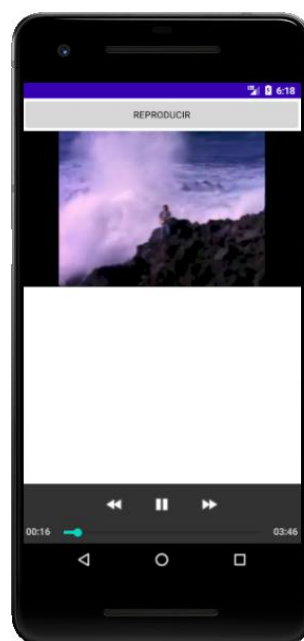
5. Ejecutar la aplicación. Se muestra una imagen similar a la figura 1. Cuando se digita el botón `REPRODUCIR`, se inicia la reproducción del video, como se muestra en la figura 2. Si se digita sobre la imagen del video en reproducción, en la sección inferior se muestra una barra con tres botones para controlar la reproducción del video, como se muestra en la figura 3.



**Figura 1.** El botón de inicio.



**Figura 2.** Video en curso.



**Figura 3.** Botones de control del video.

**Ejercicio 1.**

Diseñar una aplicación móvil que permita seleccionar la reproducción de audio o video. Utilizar la clase `Intent` y `Bundle` para enviar las solicitudes de ejecución. Se pueden utilizar los archivos de audio o video de su preferencia.