

# PDM Avanzado Reprodución de Audio

---

## Sumario

---

### Introdución

#### MediaPlayer

- Reproducir música
- Caso práctico
  - Preparación
  - Código do layout xml
  - Código da Activity

#### Carga asíncrona

- Caso práctico
  - Preparación
  - Código da Activity

## Introdución

---

No caso da reprodución do audio, teremos que facer uso das clase **MediaPlayer**.

Clases que se empregan:

- **MediaPlayer** (<http://developer.android.com/reference/android/media/MediaPlayer.html>): esta clase é a principal utilizada para reproducir audio – vídeo.
- **AudioManager** (<http://developer.android.com/reference/android/media/AudioManager.html>): permite establecer o volume e o tipo de son (vibración, normal, silencio) nunha chamada. Nestes exemplos non imos facer uso dela.

Permisos necesarios a engadir no arquivo AndroidManifest.xml (nos imos engadir todos):

- Se o dispositivo necesita conexión a internet (para escoitar música en streaming, por exemplo) temos que engadir o permiso:

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

- Permiso Wake-Lock: no momento no que o S.O. non vexa 'movemento' no dispositivo (ou sexa uso) pasará a un estado de modo suspendido. Se queremos que a nosa activity non entre en dito estado podemos facer uso do método `MediaPlayer.setWakeMode()` e polo tanto necesitaremos o permiso `wake_lock`:

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

- Permiso READ SD CARD: se queremos que poida ler algún arquivo na SD Card externa, teremos que engadir este permiso

Ler este punto da Wiki ([https://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Avanzado\\_Datos\\_Persistentes\\_Arquivos#A\\_clase\\_File\\_e\\_a\\_clase\\_Environment](https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Datos_Persistentes_Arquivos#A_clase_File_e_a_clase_Environment)) para entender cando é necesario o permiso.

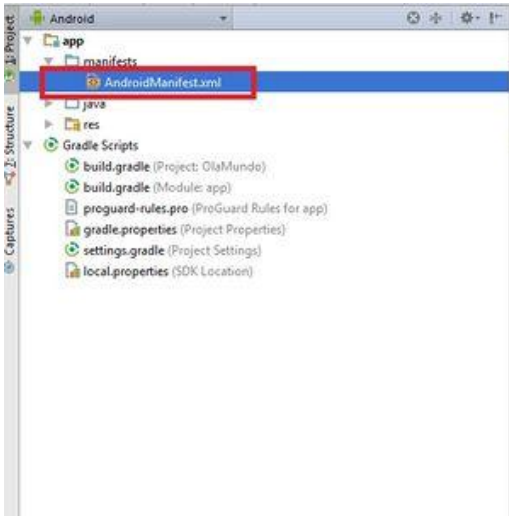
Indicar que este tipo de permiso é considerado 'perigoso' polo S.O. polo que hai que programar pedilo a partires da API 23, como está indicado no enlace anterior.

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

- Permiso de escritura na SD Card: Para ter acceso á memoria externa teremos que especificar no arquivo AndroidManifest.xml que a nosa aplicación necesita permiso de lectura/escritura en dicha memoria.

- Se imos ler na tarxeta SD:
  - Se a versión do S.O. Android é inferior á 4.1 non precisamos ningún permiso.
  - Se a versión do S.O. Android é superior ou igual á 4.1 debemos engadir o permiso: **<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>**
- Se imos escribir na tarxeta SD:
  - Se a versión do S.O. Android é inferior á 4.4 o permiso é: **<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>** .
  - Se a versión do S.O. Android é a 4.4 ou superior. Podemos poñer o mesmo permiso anterior pero as aplicacións dispoñen dun cartafol para escribir na SD (cartafol Android/data/paquete/) sen necesidade de ter o permiso anterior.

Os permisos necesarios son postos no ficheiro **AndroidManifest.xml** da aplicación.



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.angel.olamundo" >
3
4   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
5
6   <application
7       android:allowBackup="true"
8       android:icon="@mipmap/ic_launcher"
9       android:label="@string/app_name"
10      android:supportRtl="true"
11      android:theme="@style/AppTheme" >
12       <activity android:name=".MainActivity" >
13           <intent-filter>
14               <action android:name="android.intent.action.MAIN" />
15
16               <category android:name="android.intent.category.LAUNCHER" />
17           </intent-filter>
18       </activity>
19   </application>
20
21 </manifest>

```

- **Importante:** Débense poñer antes da etiqueta <application>.

Nota: se queremos ter unha música de fondo na nosa aplicación o máis lóxico sería ter un servizo.

- **Nota importante:** A maiores, se imos instalar a nosa aplicación sobre un dispositivo cunha API<=23 é necesario pedir o permiso de acceso á tarxeta SD por programación como está indicado neste punto da Wiki ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Avanza do\\_Permisos\\_AndroidManifest.xml](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanza_do_Permisos_AndroidManifest.xml)).

## MediaPlayer

Os recursos (arquivos de audio) que podemos utilizar con esta clase poden ser:

- **Locais:** arquivos locais gardados no cartafol /res/raw/ (deberemos crear o cartafol /raw/ se non existe, explicado a continuación).

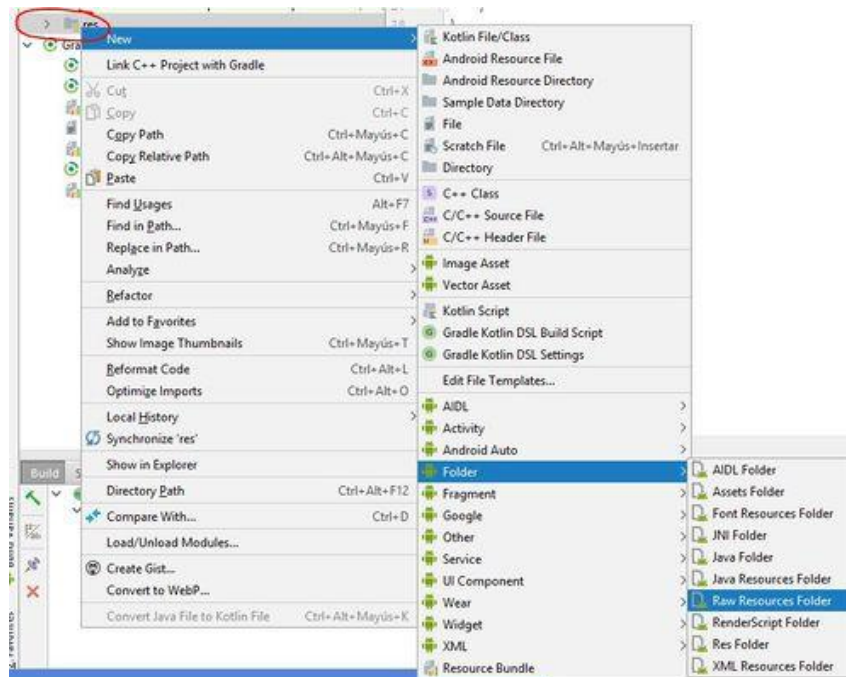
Nota: Tamén se podería empregar un cartafol 'assets' que veremos na UD de bases de datos, pero que nos non empregaremos nesta unidade.

- **URLs internas:** unha URI é unha forma de identificar de forma unívoca un recurso (neste caso multimedia). Podemos referenciar arquivos multimedia gardados nas diferentes tarxetas de memoria do dispositivo.
- **URL Externos (streaming):** arquivos multimedia que se atopan en Internet.

Podesdes consultar a lista de formatos de audio – vídeo soportados en <http://developer.android.com/guide/appendix/media-formats.html>

Para poder acceder aos diferentes cartafóis (da memoria interna ou da tarxeta SD) imos empregar unha serie de métodos propios da clase Activity e outros da clase Environment que nos darán a ruta aos cartofoles desas unidades. Podesdes consultar a lista de métodos nesta sección ([https://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Avanzado\\_Datos\\_Persistentes\\_Arquivos#A\\_clase\\_File\\_e\\_a\\_clase\\_Environment](https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Datos_Persistentes_Arquivos#A_clase_File_e_a_clase_Environment)).

Imos facer unha pequena práctica na que imos reproducir un arquivo multimedia. Creamos no proxecto de Android - Multimedia un cartafol de nome 'raw' dentro do cartafol /res/ do proxecto:



## Reproducir música

Para cargar un arquivo de audio temos que seguir os seguintes pasos:

- Crear un obxecto da clase MediaPlayer.

```
1 private MediaPlayer mediaPlayer;
2 .....
3 mediaPlayer = new MediaPlayer();
```

- Chamar ao método **setDataSource** que queiramos (pode dar lugar a varios tipos de excepcións, polo que teremos que usar try catch para cada tipo delas ou ben capturalos cunha clase **Exception** aínda que esta forma non é a recomendada).

No exemplo usamos **Exception** para simplificar o código xa que se non teríamos que ter isto:

```
1 try {
2     mediaPlayer.setDataSource(path);
3 } catch (IllegalArgumentException e) {
4     // TODO Auto-generated catch block
5     e.printStackTrace();
6 }
```

```

6      } catch (SecurityException e) {
7          // TODO Auto-generated catch block
8          e.printStackTrace();
9      } catch (IllegalStateException e) {
10         // TODO Auto-generated catch block
11         e.printStackTrace();
12     } catch (IOException e) {
13         // TODO Auto-generated catch block
14         e.printStackTrace();
15     }

```

Con este método indicamos de onde imos a cargar o audio.

Podemos ter varias posibilidades:

- O arquivo se atopa en /res/raw:

```

1      MediaPlayer mediaplayer = new MediaPlayer();
2      Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.snowflake_persephone);
3      try {
4          mediaplayer.setDataSource(getApplicationContext(), uri);
5          .....
6      }

```

- O arquivo provén da tarxeta SD.

Nota: Lembrar poñer o permiso <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> no AndroidManifest.xml

Temos varias opcións:

```

1      ....
2      File file = new File(Environment.getExternalStorageDirectory().getAbsolutePath() + File.separator + "Download"
+ File.separator + "snowflake_persephone.mp3");
3      if (!file.exists()){
4          Toast.makeText(this, "NON EXISTE", Toast.LENGTH_LONG).show();
5          return;
6      }
7      Uri uri = Uri.fromFile(file);
8      try {
9          mediaPlayer.setDataSource(getApplicationContext(), uri);
10     } catch (IOException e) {
11         e.printStackTrace();
12     }

```

```

1 MediaPlayer mediaplayer = new MediaPlayer();
2 String path = Environment.getExternalStorageDirectory().getAbsolutePath() + File.separator + "snowflake_persephone.mp3";
3 try {
4     mediaplayer.setDataSource(Uri.encode(path));
5     .....
6 }

```

```

1 MediaPlayer mediaplayer = new MediaPlayer();
2 String path = Environment.getExternalStorageDirectory().getAbsolutePath() + File.separator + "snowflake_persephone.mp3";
3 Uri uri = Uri.parse(path);
4 try {
5     mediaplayer.setDataSource(getApplicationContext(), uri);
6     .....
7 }

```

- O arquivo provén de Internet (música en Streaming).

Nota: Se estades a utilizar un dispositivo real ten que ter conexión a Internet para que funcione.  
Temos varias opcións:

```

1 MediaPlayer mediaplayer = new MediaPlayer();
2 Uri uri = Uri.parse("http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3");
3 try {
4     mediaplayer.setDataSource(getApplicationContext(),uri);
5     .....
6 }

```

```

1 MediaPlayer mediaplayer = new MediaPlayer();
2 String url="http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3";
3 try {
4     mediaplayer.setDataSource(url);
5     .....
6 }

```

Pode suceder que a dirección da URL teña espazos en branco ou caracteres especiais. Nese caso teremos que 'codificar' antes a url a partires do último carácter '/':

```

1 MediaPlayer mediaplayer = new MediaPlayer();
2 String url="http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3";
3 int pos = url.lastIndexOf('/') + 1;
4 Uri uri = Uri.parse(url.substring(0, pos) + Uri.encode(url.substring(pos)));
5 try {
6     mediaplayer.setDataSource(getApplicationContext(),uri);
7     .....
8 }

```

- Chamar ao método **prepare** (pode lanzar unha excepción IOException).
- Chamar ao método **start**.

Todos estes pasos son necesarios facelos nesta orde xa que o MediaPlayer vai pasar por unha serie de **estados** que nos van obrigar a poder facer só unha accións determinadas dependendo do estado no que nos atopemos.

Vexamos o diagrama de estados do MediaPlayer:

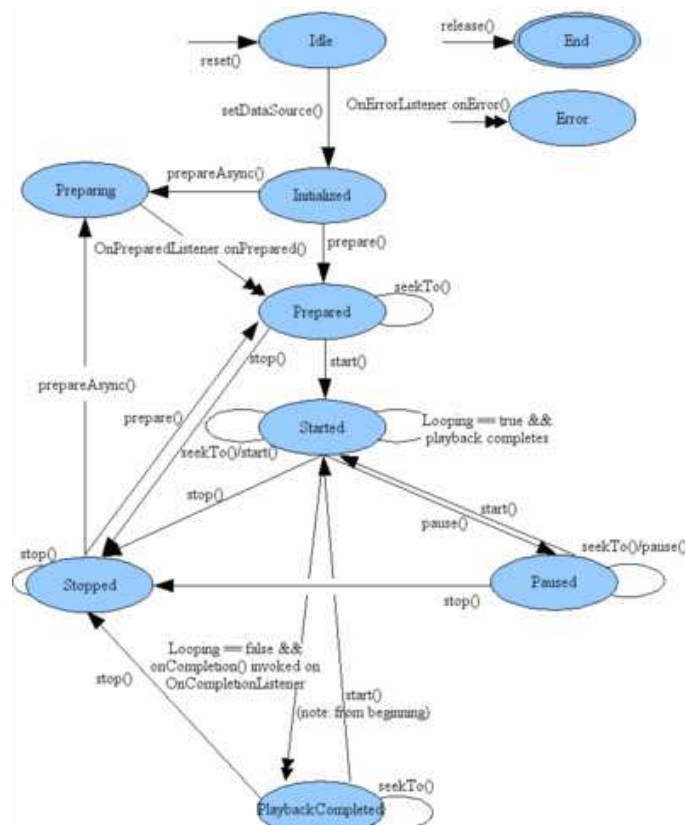


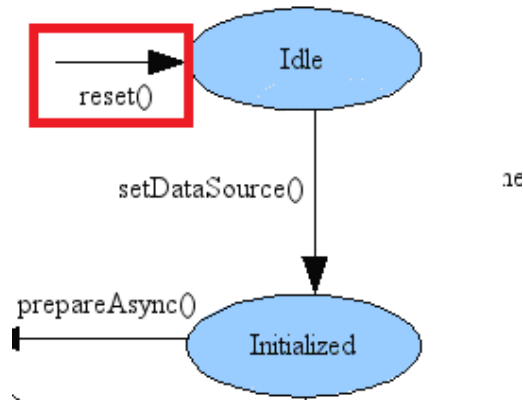
Gráfico obtido dende <http://developer.android.com/reference/android/media/MediaPlayer.html>

Imos analizar dito diagrama.

Cando instanciamos o MediaPlayer (facemos o new) nos situamos no **estado idle**. Dende dito estado so podemos pasar o **estado initialized** chamando o método `setDataSource()`.

Se intentemos chamar a outro método (coma `start()`,...) daranos unha excepción (`IllegalStateException`). Cando estamos a tocar unha canción non podemos cambiar por outra, xa que aínda que chamemos o método `stop` e pasemos o **estado de stop**, dende dito estado non podemos chamar ao método `setDataSource` (mirar diagrama).

A única forma de cambiar de canción será chamando ao método `reset()`, que volve ao estado Idle como amosamos neste anaco do diagrama anterior:



Podemos cargar un arquivo de audio sen facer os pasos anteriores da seguinte forma:

```

1 MediaPlayer mediaPlayer=MediaPlayer.create(this, R.raw.snowflake_persephone);
2 mediaPlayer.start();
  
```

Sendo `R.raw.snowflake_persephone` un arquivo de audio que se atopa no cartafol `/res/raw`.

Desta forma estamos a pasar directamente ao estado Prepared do diagrama anterior. No nosa práctica non usaremos esta forma de cargar o audio.

## Caso práctico

- Partimos que xa temos creado o proxecto inicial como xa indicamos anteriormente ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

Se non o temos creado antes, crearemos un novo paquete de nome: **Multimedia** como un subpaquete do teu paquete principal.

- Dentro do paquete **Adaptadores** crear unha nova 'Empty Activity' de nome: **UD05\_01\_Audio\_Reproducir** de tipo Launcher e sen compatibilidade.

Modifica o arquivo **AndroidManifest.xml** e engade unha label á activity como xa vimos na creación do proxecto base ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

Nota: Nos volcados de pantalla pode ser que apareza outra unidade no título. Non facede caso.

O obxectivo desta práctica é reproducir un arquivo de música gardado en /res/raw e outro en streaming de Internet.

Nota: Se instalades a aplicación nun dispositivo real tedes que ter conexión a Internet para que funcione a reprodución en streaming e deberedes de poñer o permiso correspondente no AndroidManifest.xml



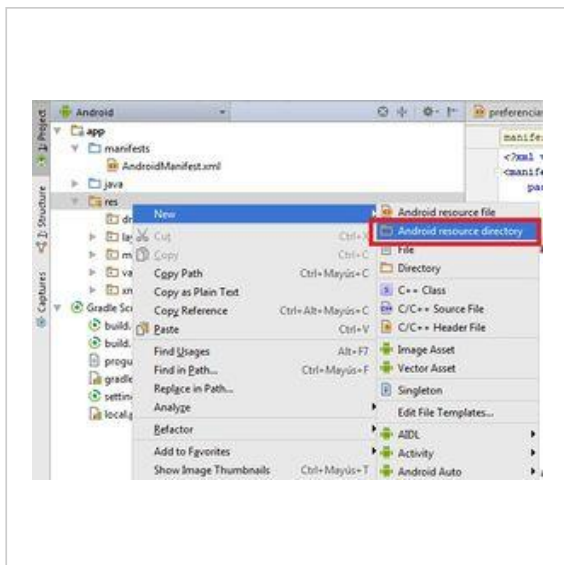
### Preparación

- Copiamos o arquivo de audio (ou outro calquera que podades utilizar) ao cartafol /res/raw.

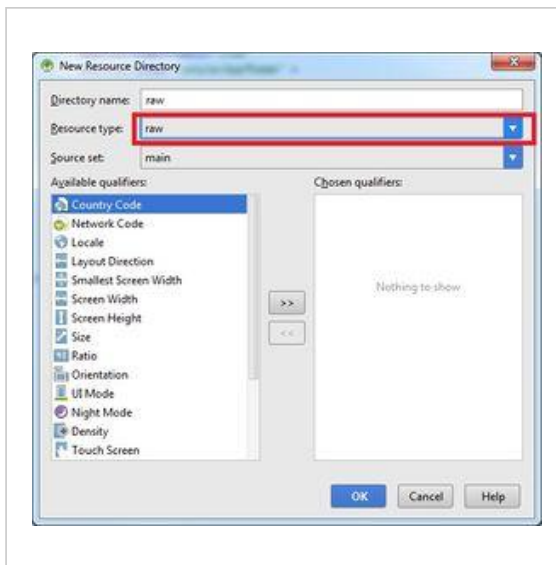
Media:snowflake\_persephone.mp3 .

**ATENCIÓN:** Ao ser un arquivo que vai ser gardado no interior do cartafol /res de Android, o arquivo **non pode ter letras maiúsculas**.

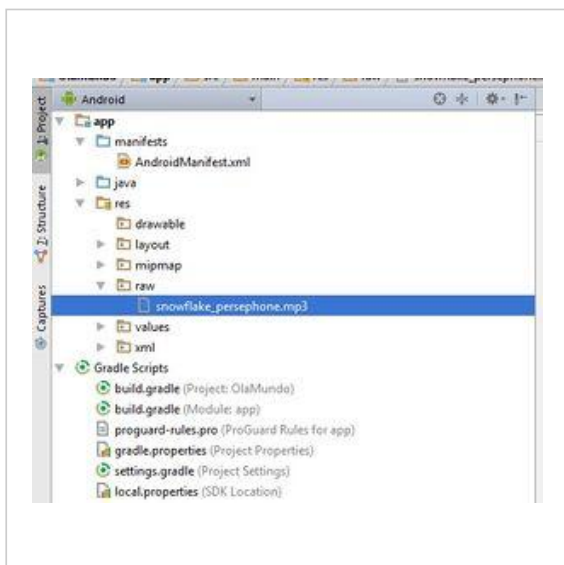
**Copiando arquivo de audio a /res/raw**



Creamos el cartafol **raw** dentro de /res.



Escogemos el tipo raw.



Copiamos el archivo de audio anterior. **TODAS LAS LETRAS DEBEN IR EN MINÚSCULAS.**

### Código del layout xml

Nota: Por motivos de tiempo para el alumnado el diseño no hace uso de constantes externas definidas en el cartafol values. Queda claro que esta debería ser la opción escollida para el diseño de las Interfaces de Usuario.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="wrap_content"
7   android:orientation="vertical" >
8
9
10  <TextView
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:gravity="center_horizontal"
14    android:layout_marginBottom="20dp"
15    android:text="UD6_01_Multimedia"
16    android:textSize="20sp" />
17
18

```



```

19 <TableLayout
20     android:layout_width="match_parent"
21     android:layout_height="wrap_content"
22     android:layout_gravity="bottom"
23     android:isScrollContainer="true"
24 >
25
26 <TableRow
27     android:id="@+id/tableRow1"
28     android:layout_width="match_parent"
29     android:layout_height="wrap_content" >
30
31
32     <Button
33         android:id="@+id/btnReprLocal_UD05_01_audio_repr"
34         android:width="0dp"
35         android:layout_weight="1"
36         android:layout_width="wrap_content"
37         android:layout_height="wrap_content"
38         android:text="LOCAL"
39         android:textSize="10sp"
40     />
41
42     <Button
43         android:id="@+id/btnReprInternet_UD05_01_audio_repr"
44         android:width="0dp"
45         android:layout_weight="1"
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
48         android:text="INTERNET"
49         android:textSize="10sp"
50     />
51
52 </TableRow>
53
54 <TableRow
55     android:id="@+id/tableRow2"
56     android:layout_width="match_parent"
57     android:layout_height="wrap_content" >
58
59     <Button
60         android:id="@+id/btnPararRepr_UD05_01_audio_repr"
61         android:layout_width="0dp"
62         android:layout_height="wrap_content"
63         android:layout_weight="1"
64         android:text="PARAR"
65         android:textSize="10sp" />
66
67 </TableRow>
68 <TableRow
69     android:id="@+id/tableRow4"
70     android:layout_width="match_parent"
71     android:layout_height="wrap_content" >
72
73     <Button
74         android:id="@+id/btnSair_UD05_01_audio_repr"
75         android:layout_width="0dp"
76         android:layout_height="wrap_content"
77         android:layout_weight="1"
78         android:text="SAIR"
79         android:textSize="10sp"
80     />
81
82 </TableRow>
83
84
85 </TableLayout>
86
87 </LinearLayout>

```

## Código da Activity

**Obxectivo:** Reproducir e parar música local e de Internet.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Multimedia;
2
3 import android.Manifest;
4 import android.app.Activity;
5 import android.content.pm.PackageManager;

```

```

6 import android.media.MediaPlayer;
7 import android.net.Uri;
8 import android.os.Build;
9 import android.os.Bundle;
10 import android.util.Log;
11 import android.view.View;
12 import android.widget.Button;
13 import android.widget.Toast;
14 import es.cursoandroid.cifprodolfoucha.aprendiendo.R;
15
16 public class UD05_01_Audio_Reproducir extends Activity {
17     private MediaPlayer mediaplayer;
18     private boolean pause; // Indica se o mediaplayer estaba tocando cando cambiamos de aplicación
19
20
21     /**
22      * Cambia a canción no MediaPlayer
23      * @param uri
24      */
25     private void cambiarCancion(Uri uri){
26         try {
27             mediaplayer.reset();
28
29             mediaplayer.setDataSource(getApplicationContext(),uri);
30             mediaplayer.prepare();
31             mediaplayer.start();
32         } catch (Exception e) {
33             // TODO Auto-generated catch block
34             e.printStackTrace();
35             Log.e("MULTIMEDIA",e.getMessage());
36         }
37     }
38
39     private void xestionarEventos(){
40
41         Button btnLocal = (Button)findViewById(R.id.btnReprLocal_UD05_01_audio_repr);
42         btnLocal.setOnClickListener(new View.OnClickListener() {
43
44             @Override
45             public void onClick(View v) {
46                 // TODO Auto-generated method stub
47                 Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.snowflake_persephone);
48                 cambiarCancion(uri);
49             }
50         });
51
52         Button btnInternet = (Button)findViewById(R.id.btnReprInternet_UD05_01_audio_repr);
53         btnInternet.setOnClickListener(new View.OnClickListener() {
54
55             @Override
56             public void onClick(View v) {
57                 // TODO Auto-generated method stub
58                 String url = "http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3";
59                 // Se a URL ven con espacios en branco teremos que facer un parteUrl + Uri.encode(parteUltinaURL) sendo parteUltinaURL a
60                 parte que vai no último lugar.
61                 Uri uri = Uri.parse(url);
62                 cambiarCancion(uri);
63             }
64         });
65         Button btnParar = (Button)findViewById(R.id.btnPararRepr_UD05_01_audio_repr);
66         btnParar.setOnClickListener(new View.OnClickListener() {
67
68             @Override
69             public void onClick(View v) {
70                 // TODO Auto-generated method stub
71                 if (mediaplayer.isPlaying())
72                     mediaplayer.stop();
73                 pause=false;
74             }
75         });
76
77         Button btnSair = (Button) findViewById(R.id.btnSair_UD05_01_audio_repr);
78         btnSair.setOnClickListener(new View.OnClickListener() {
79
80             @Override
81             public void onClick(View v) {
82                 // TODO Auto-generated method stub
83                 finish();
84             }
85         });
86
87     }
88
89     @Override
90     protected void onPause() {
91         super.onPause();
92     }

```

```

93
94     if (mediaplayer.isPlaying()){
95         mediaplayer.pause();
96         pause = true;
97     }
98 }
99
100 @Override
101 protected void onResume() {
102     super.onResume();
103
104     if (pause) {
105         mediaplayer.start();
106         pause = false;
107     }
108 }
109
110
111
112 @Override
113 protected void onDestroy() {
114     super.onDestroy();
115
116     if (mediaplayer.isPlaying()) mediaplayer.stop();
117
118     if (mediaplayer != null) mediaplayer.release();
119     mediaplayer = null;
120 }
121
122
123 @Override
124 protected void onCreate(Bundle savedInstanceState) {
125     super.onCreate(savedInstanceState);
126     setContentView(R.layout.activity_ud05_01__audio__reproducir);
127
128     mediaplayer = new MediaPlayer();
129     pause = false;
130     xestionarEventos();
131
132 }
133 }
134 }

```

- Liña 18: Definimos o reprodutor.
- Liña 19: Definimos unha variable booleana que nos indique cando o reprodutor está en estado de PAUSE.
- Liñas 26-40: Cambiamos de canción. Fixarse como sempre chamamos ao método reset() para pasar ao estado que nos permite cambiar a fonte de audio.
- Liñas 48-52: Prememos o botón de LOCAL e definimos a URI para buscar o arquivo en /res/raw.
- Liñas 59-64: Prememos o botón de INTERNET e definimos a URI para buscar o arquivo indicada na URL.
- Liñas 71-76: Prememos o botón de STOP. Comprobamos se o mediaplayer está tocando para paralo.
- Liñas 106-109: Comprobamos se o mediaplayer está tocando para poñelo en pause. Cambiamos o valor da variable pause a true. Isto é necesario xa que non temos forma de saber a través do mediaplayer se este se atopa nese estado.
- Liñas 106-109: Se volvemos á aplicación comprobamos se o mediaplayer estaba tocando (pause=true) e nese caso continuamos tocando a canción. Isto pasará cando cambiemos de aplicación sen pechala.
- Liñas 118-121: Se saímos da aplicación paramos de tocar e liberamos o mediaplayer.
- Liña 128: Instanciamos o mediaplayer.

- **Nota:** Podemos controlar cando unha canción chega ao seu final coa interface MediaPlayer.OnCompletionListener() e chamando ao método setOnCompletionListener(Interface) do MediaPlayer.

## Carga asíncrona

Cando poñemos en marcha un arquivo de audio que non se atopa localmente, non é boa idea facelo no fío principal da aplicación, xa que se leva un tempo cargar a música, e a aplicación quedaría bloqueada mentres tanto. Para evitalo teríamos que crear nós un fío de execución separado do principal, pero este traballo xa o temos feito, se chamamos ao método prepareAsync() (<http://developer.android.com/reference/android/media/MediaPlayer.html#prepareAsync%28%29>) . Cando a mediaplayer estea listo, chamará automaticamente ao método onPrepared da interface MediaPlayer.OnPreparedListener. Para asociar dita interface o mediaplayer, faremos uso do método setOnPreparedListener().

Os pasos serían:

- Asociar a interface ao mediaplayer:

```

1      mediaPlayer.setOnPreparedListener(new OnPreparedListener(){
2
3      }
4      );

```

- Ao definir internamente a clase que xestionará o preparedlistener, é necesario implementar o método da interface dentro da definición:

```

1      mediaPlayer.setOnPreparedListener(new OnPreparedListener(){
2
3          public void onPrepared(MediaPlayer arg0) {
4              // TODO Auto-generated method stub
5
6          }
7      }
8      );

```

- Agora xa podemos codificar o método onPrepared que será chamado automaticamente cando o música estea preparada. O único que temos que facer dentro de dito método será unha chamada ao método start() da clase MediaPlayer.

Nota: Agora en vez de chamar o método prepare, teremos que chamar o método prepareAsync().

## Caso práctico

O obxectivo desta práctica é facer o mesmo que na práctica anterior pero chamando ao método **prepareAsync**.

### Preparación

- Partimos que xa temos creado o proxecto inicial como xa indicamos anteriormente ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

Se non o temos creado antes, crearemos un novo paquete de nome: **Multimedia** como un subpaquete do teu paquete principal.

- Dentro do paquete **Adaptadores** crear unha nova 'Empty Activity' de nome: **UD05\_02\_Audio\_Reproducir** de tipo Launcher e sen compatibilidade.

Modifica o arquivo **AndroidManifest.xml** e engade unha label á activity como xa vimos na creación do proxecto base ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

Na activity cargaremos o mesmo layout que no caso práctico anterior.

Quen o prefira pode facelo sobre un layout novo copiando todo o código e cambiando os nomes (id's) dos views.

### Código da Activity

**Obxectivo:** Amosar como cargar unha canción de forma asíncrona.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Multimedia;
2
3
4 import android.app.Activity;
5 import android.media.MediaPlayer;
6 import android.net.Uri;
7 import android.os.Bundle;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.Button;

```

```

11
12 import es.cursoandroid.cifprodolfoucha.aprendiendo.R;
13
14 public class UD05_02_Audio_Reproducir extends Activity {
15     private MediaPlayer mediaPlayer;
16     private boolean pause; // Indica se o mediaPlayer estaba tocando cando cambiamos de aplicación
17
18     /**
19      * Cambia a canción no MediaPlayer
20      * @param uri
21      */
22     private void cambiarCancion(Uri uri){
23         try {
24             mediaPlayer.reset();
25
26             mediaPlayer.setDataSource(getApplicationContext(),uri);
27             mediaPlayer.prepareAsync();
28             // mediaPlayer.start(); O FACEMOS DE FORMA ASINCRONA
29         } catch (Exception e) {
30             // TODO Auto-generated catch block
31             e.printStackTrace();
32             Log.e("MULTIMEDIA",e.getMessage());
33         }
34     }
35
36 }
37
38 private void xestionarEventos(){
39     mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
40
41         @Override
42         public void onPrepared(MediaPlayer mp) {
43             // TODO Auto-generated method stub
44             mediaPlayer.start();
45         }
46     });
47
48     Button btnLocal = (Button)findViewById(R.id.btnReprLocal_UD05_01_audio_repr);
49     btnLocal.setOnClickListener(new View.OnClickListener() {
50
51         @Override
52         public void onClick(View v) {
53             // TODO Auto-generated method stub
54             Uri uri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.snowflake_persephone);
55             cambiarCancion(uri);
56         }
57     });
58
59     Button btnInternet = (Button)findViewById(R.id.btnReprInternet_UD05_01_audio_repr);
60     btnInternet.setOnClickListener(new View.OnClickListener() {
61
62         @Override
63         public void onClick(View v) {
64             // TODO Auto-generated method stub
65             String url = "http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3";
66             // Se a URL ven con espacios en branco teremos que facer un parteUrl + Uri.encode(parteUltinaURL) sendo parteUltinaURL a
67             parte que vai no último lugar.
68             Uri uri = Uri.parse(url);
69             cambiarCancion(uri);
70         }
71     });
72     Button btnParar = (Button)findViewById(R.id.btnPararRepr_UD05_01_audio_repr);
73     btnParar.setOnClickListener(new View.OnClickListener() {
74
75         @Override
76         public void onClick(View v) {
77             // TODO Auto-generated method stub
78             if (mediaPlayer.isPlaying())
79                 mediaPlayer.stop();
80             pause=false;
81         }
82     });
83
84     Button btnSair = (Button) findViewById(R.id.btnSair_UD05_01_audio_repr);
85     btnSair.setOnClickListener(new View.OnClickListener() {
86
87         @Override
88         public void onClick(View v) {
89             // TODO Auto-generated method stub
90             finish();
91         }
92     });
93
94 }
95
96 @Override
97

```

```
98     protected void onPause() {
99         super.onPause();
100
101         if (mediaplayer.isPlaying()){
102             mediaplayer.pause();
103             pause = true;
104         }
105     }
106
107     @Override
108     protected void onResume() {
109         super.onResume();
110
111         if (pause) {
112             mediaplayer.start();
113             pause = false;
114         }
115     }
116
117     @Override
118     protected void onDestroy() {
119         super.onDestroy();
120
121         if (mediaplayer.isPlaying()) mediaplayer.stop();
122
123         if (mediaplayer != null) mediaplayer.release();
124         mediaplayer = null;
125     }
126
127
128     @Override
129     protected void onCreate(Bundle savedInstanceState) {
130         super.onCreate(savedInstanceState);
131         setContentView(R.layout.activity_ud05_01__audio__reproducir);
132
133         mediaplayer = new MediaPlayer();
134         pause = false;
135         xestionarEventos();
136
137     }
138 }
139 }
```

- **Liña 133:** Fixarse como estamos a cargar o layout da práctica 1.

**[Enlace a la página principal de la UD7 \(https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n\\_de\\_dispositivos\\_m%C3%B3viles#UNIDADE\\_7:\\_Multimedia\)](https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n_de_dispositivos_m%C3%B3viles#UNIDADE_7:_Multimedia)**

**[Enlace a la página principal del curso \(https://wiki.cifprodolfoucha.es/index.php?title=Programación\\_de\\_dispositivos\\_móviles\)](https://wiki.cifprodolfoucha.es/index.php?title=Programación_de_dispositivos_móviles)**

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2014).

---

Obtenido de «[https://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Avanzado\\_Reproducción\\_de\\_Audio&oldid=12082](https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Reproducción_de_Audio&oldid=12082)»

---

Esta página se editó por última vez el 10 ene 2021 a las 19:02.