1 PDM Avanzado Reprodución de Audio

1.1 Sumario

1.2 Introdución

No caso da reprodución do audio, teremos que facer uso das clases **MediaPlayer** e **AudioManager**.

Clase utilizadas: MediaPlayer: esta clase é a principal utilizada para reproducir audio ? vídeo. AudioManager: manexa fontes de audio e gravacións de audio en dispositivos. Nesta parte só imos utilizala para indicar o tipo de audio.

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:

```
<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:

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

- 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.

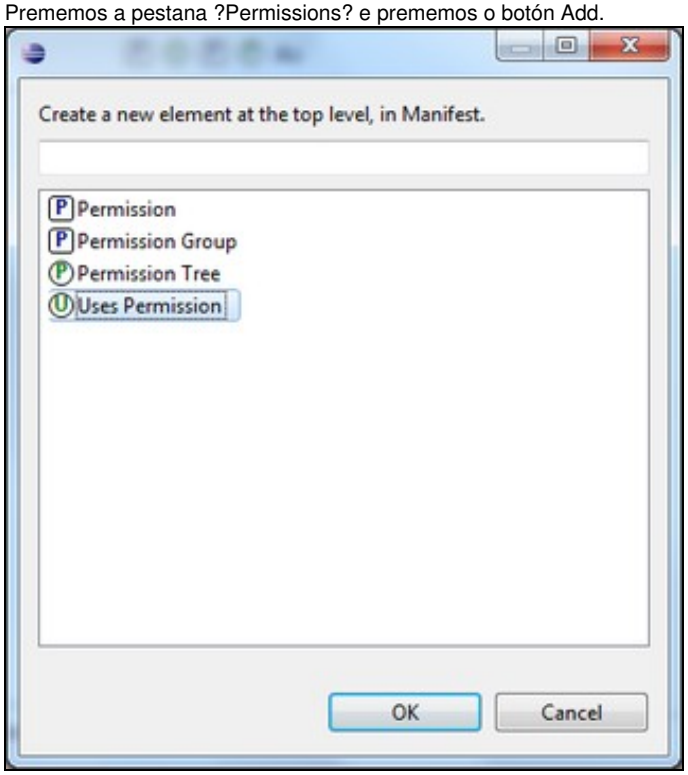
```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

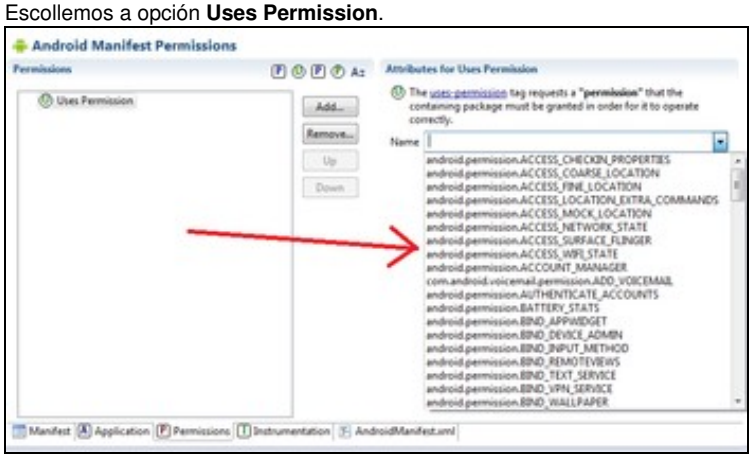
Nota: se queremos ter unha música de fondo na nosa aplicación o máis lóxico sería ter un servizo.

Para engadir ditos permisos de forma gráfica, nos abrimos o arquivo **AndroidManifest.xml**.

- Proceso para engadir permisos de forma gráfica

- 

Prememos a pestana ?Permissions? e prememos o botón Add.
- 

Escollemos a opción **Uses Permission**.
- 

Graficamente escollemos os permisos que necesitemos.

1.3 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).
- **URIs 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.

Podedes consultar a lista de formatos de audio ? vídeo soportados en http://developer.android.com/guide/appendix/media-formats.html

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:

1.3.1 Reproducir música

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

- Crear un obxecto da clase MediaPlayer.

```
private  MediaPlayer mediaplayer;
        ..............
        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:

```
try {
mediaplayer.setDataSource(path);
} catch (IllegalArgumentException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (SecurityException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (IllegalStateException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
```

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

Podemos ter varias posibilidades:

◊ O arquivo se atopa en /res/raw:

```
MediaPlayer mediaplayer = new MediaPlayer();
        Uri uri = Uri.parse("android.resource://" + getPackageName()+ "/" + R.raw.snowflake_persephone);
        try {
                mediaplayer.setDataSource(getApplicationContext(), uri);
                        .....
        }
```

◊ O arquivo provén da tarxeta SD.

Temos varias opcións:

```
MediaPlayer mediaplayer = new MediaPlayer();
String path=Environment.getExternalStorageDirectory().getAbsolutePath()+File.separator + "snowflake_persep
try {
mediaplayer.setDataSource(path);
        .....
        }

MediaPlayer mediaplayer = new MediaPlayer();
String path =Environment.getExternalStorageDirectory().getAbsolutePath()+File.separator + "snowflake_perse
Uri uri = Uri.parse(Uri.encode(path));
try {
mediaplayer.setDataSource(getApplicationContext(),uri);
        .....
        }
```

◊ 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:

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

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 '/':

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

- Chamar ao método **setAudioStreamType(AudioManager.STREAM_MUSIC)** para indicarlle que o que imos reproducir será música e poñerá o volume que teña o S.O. (lembrar que en Android podemos cambiar o volume da música, notificacións e alarmas).
- 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:

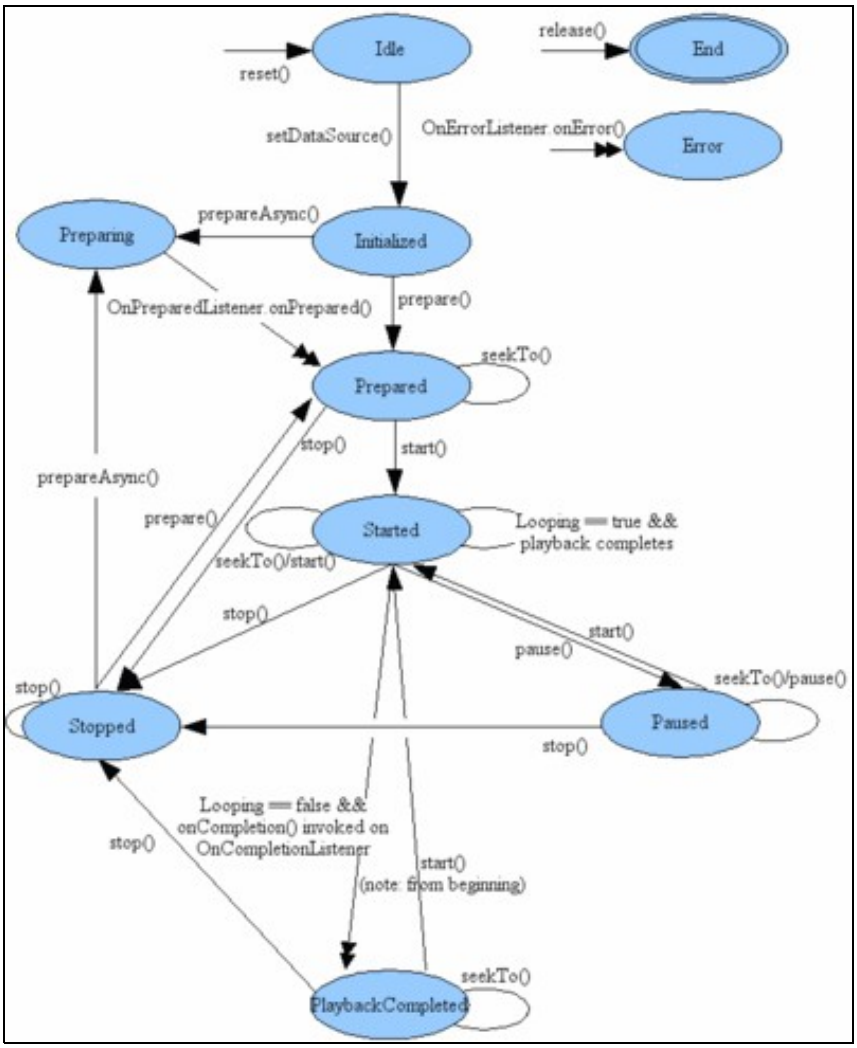


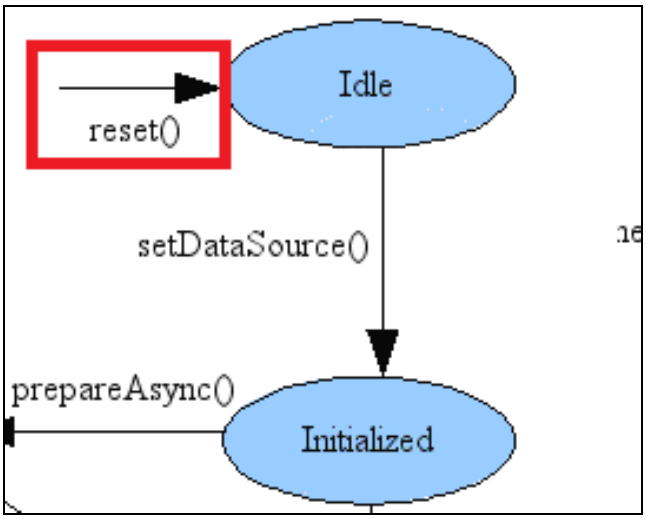
Gráfico obtido dende http://developer.android.com/reference/android/media/MediaPlayer.html

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 (IlegalStateException). 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:

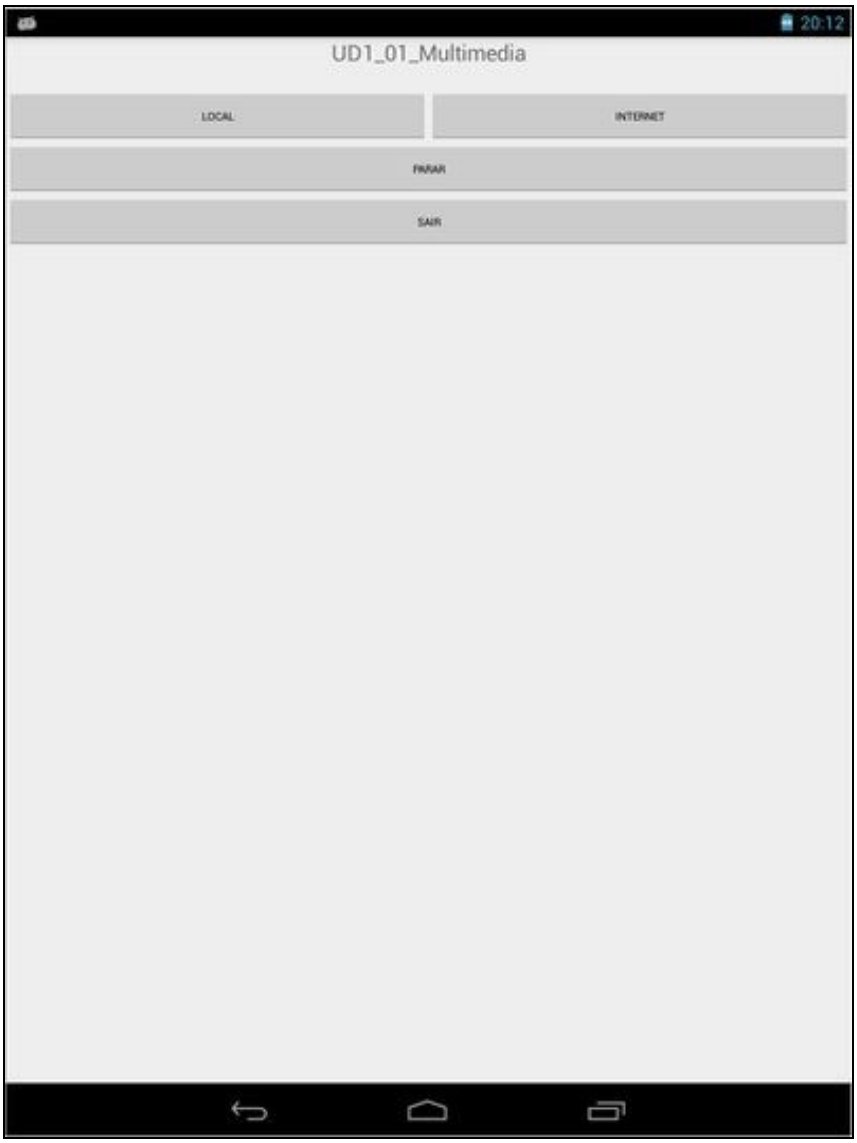
```
MediaPlayer mediaPlayer=MediaPlayer.create(this, R.raw.snowflake_persephone);
        mediaPlayer.start();
```

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.

1.3.2 Caso práctico

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.

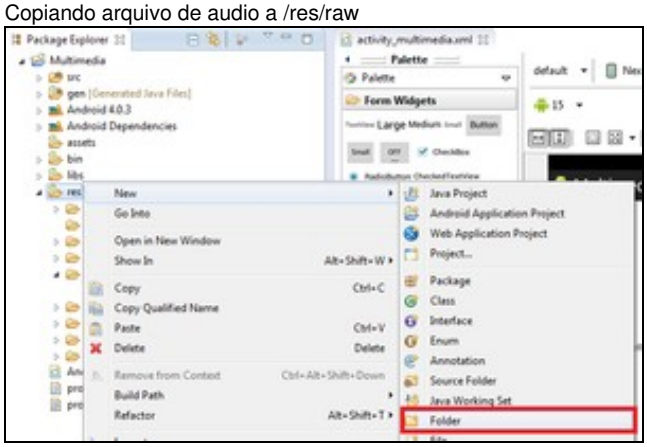


1.3.2.1 Preparación

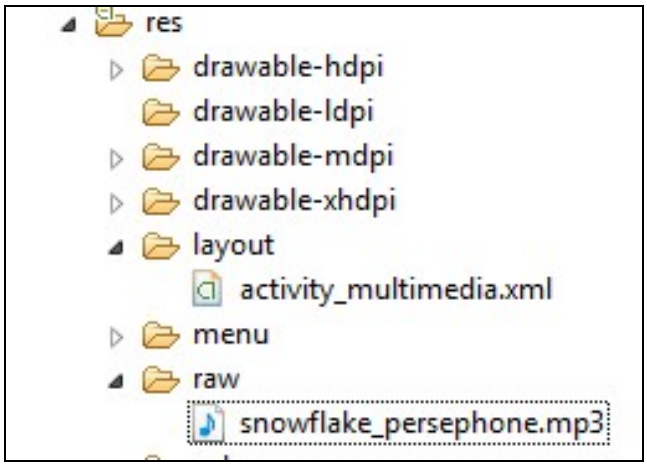
- Copiamos o arquivo de audio (ou outro calquera que podades utilizar) ao cartafol /res/raw.

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 o cartafol **raw** dentro de /res.

- 

Copiamos o arquivo de audio anterior. TODAS AS LETRAS DEBEN IR EN MINÚSCULAS.

1.3.2.2 Creamos a Activity

- Nome do proxecto: **UD2_01_MultimediaReproductor**
- Nome da activity: **UD2_01_MultimediaReproductor.java**

Código do layout xml

Nota: Por motivos de tempo para o alumnado o deseño non fai uso de constantes externas definidas no cartafol values. Queda claro que esta debería ser a opción escollida para o deseño das Interfaces de Usuario.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >


    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:layout_marginBottom="20dp"
        android:text="UD2_01_Multimedia"
        android:textSize="20sp" />


        <TableLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom"
            android:isScrollContainer="true"
            >

        <TableRow
            android:id="@+id/tableRow1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >


            <Button
                android:id="@+id/UD2_01_btnReprLocal"
                android:width="0dp"
                android:layout_weight="1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="LOCAL"
                android:textSize="10sp"
                 />

            <Button
                android:id="@+id/UD2_01_btnReprInternet"
```

```
                android:width="0dp"
                android:layout_weight="1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="INTERNET"
                android:textSize="10sp"
                />

        </TableRow>

        <TableRow
            android:id="@+id/tableRow2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <Button
                android:id="@+id/UD2_02_btnPararRepr"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="PARAR"
                android:textSize="10sp" />

        </TableRow>
        <TableRow
            android:id="@+id/tableRow4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <Button
                android:id="@+id/UD2_02_btnSairAplicacion"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="SAIR"
                android:textSize="10sp"
                />

        </TableRow>


   </TableLayout>

</LinearLayout>
```

Código da clase UD2_01_MultimediaReproductor

Obxectivo: Reproducir e parar música local e de Internet.

```
package com.platega.angel.multimedia;

import java.io.File;

import android.app.Activity;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class UD2_01_MultimediaReproductor extends Activity {

private  MediaPlayer mediaplayer;
private boolean pause;// Indica se o mediaplayer estaba tocando cando cambiamos de aplicación


/**
     * Cambia a canción no MediaPlayer
```

```
     * @param uri
     */
    private void cambiarCancion(Uri uri){
        try {
        mediaplayer.reset();

        mediaplayer.setDataSource(getApplicationContext(),uri);
                mediaplayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
                mediaplayer.prepare();
        mediaplayer.start();
  } catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
Log.e("MULTIMEDIA",e.getMessage());
  }

    }

private void xestionarEventos(){

Button btnLocal = (Button)findViewById(R.id.UD2_01_btnReprLocal);
btnLocal.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Uri uri = Uri.parse("android.resource://" + getPackageName()+ "/" + R.raw.snowflake_persephone);
cambiarCancion(uri);
}
});

Button btnInternet = (Button)findViewById(R.id.UD2_01_btnReprInternet);
btnInternet.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
String url = "http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3";
Uri uri = Uri.parse(Uri.encode(url));
cambiarCancion(uri);
}
});

Button btnParar = (Button)findViewById(R.id.UD2_02_btnPararRepr);
btnParar.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
if (mediaplayer.isPlaying())
mediaplayer.stop();
pause=false;
}
});

Button btnSair = (Button) findViewById(R.id.UD2_02_btnSairAplicacion);
btnSair.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
finish();
}
});


}

    @Override
    protected void onPause() {
        super.onPause();

        if (mediaplayer.isPlaying()){
```

```
            mediaplayer.pause();
            pause = true;
            }
        }

        @Override
        protected void onResume() {
            super.onResume();

            if (pause) {
            mediaplayer.start();
            pause = false;
            }
        }


        @Override
        protected void onSaveInstanceState(Bundle estado) {
                estado.putBoolean("MEDIAPLAYER_PAUSE", pause);
                super.onSaveInstanceState(estado);
        }

        @Override
        protected void onRestoreInstanceState(Bundle savedInstanceState) {
                super.onRestoreInstanceState(savedInstanceState);
                savedInstanceState.putBoolean("MEDIAPLAYER_PAUSE", false);
                pause = savedInstanceState.getBoolean("MEDIAPLAYER_PAUSE");
        }

        @Override
        protected void onDestroy() {
            super.onDestroy();

            if (mediaplayer.isPlaying()) mediaplayer.stop();

            if (mediaplayer != null) mediaplayer.release();
            mediaplayer = null;

        }

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_ud2_01__multimedia_reproductor);

            mediaplayer = new MediaPlayer();
            pause = false;
            xestionarEventos();


        }
    }
```

- 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 113-124: Se cambiamos de aplicación recuperamos o estado do mediaplayer. Despois poñemos a false a variable pause por se xiramos o dispositivo estando tocando, para que non recupere o último estado.
- Liñas 126-135: De saímos da aplicación paramos de tocar e liberamos o mediaplayer.
- Liña 142: Instanciamos o mediaplayer.

1.4 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() . 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:

```
mediaplayer.setOnPreparedListener(new OnPreparedListener(){

        }
        );
```

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

```
mediaplayer.setOnPreparedListener(new OnPreparedListener(){

public void onPrepared(MediaPlayer arg0) {
// TODO Auto-generated method stub

}
        }
        );
```

- 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().

1.4.1 Caso práctico

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

1.4.1.1 Preparación

O alumno debe crear unha nova activity de nome **UD2_02_MultimediaReproductor**.

Esta activity ten o mesmo layout que a activity anterior (activity_ud2_01__multimedia_reproductor.xml) xa que non imos facer modificacións no seu aspecto.

1.4.1.2 Creamos a activity

- Nome do proxecto: **UD2_02_MultimediaReproductor**
- Nome da activity: **UD2_02_MultimediaReproductor.java**

Código da clase UD2_02_MultimediaReproductor
Obxectivo: Amosar como cargar unha canción de forma asíncrona.

```
package com.platega.angel.multimedia;

import android.app.Activity;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnPreparedListener;
import android.net.Uri;
import android.os.Bundle;
```

```
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class UD2_02_MultimediaReproductor extends Activity {

private  MediaPlayer mediaplayer;
private boolean pause;// Indica se o mediaplayer estaba tocando cando cambiamos de aplicación


/**
     * Cambia a canción no MediaPlayer
     * @param uri
     */
    private void cambiarCancion(Uri uri){
        try {
        mediaplayer.reset();

        mediaplayer.setDataSource(getApplicationContext(),uri);
                mediaplayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
                mediaplayer.prepareAsync();
//              mediaplayer.start();            O FACEMOS DE FORMA ASINCRONA
  } catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
Log.e("MULTIMEDIA",e.getMessage());
  }

    }

private void xestionarEventos(){

mediaplayer.setOnPreparedListener(new OnPreparedListener() {

@Override
public void onPrepared(MediaPlayer mp) {
// TODO Auto-generated method stub
mediaplayer.start();
}
});

Button btnLocal = (Button)findViewById(R.id.UD2_01_btnReprLocal);
btnLocal.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
Uri uri = Uri.parse("android.resource://" + getPackageName()+ "/" + R.raw.snowflake_persephone);
cambiarCancion(uri);
}
});

Button btnInternet = (Button)findViewById(R.id.UD2_01_btnReprInternet);
btnInternet.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
String url = Uri.encode("http://www.mfiles.co.uk/mp3-downloads/edvard-grieg-peer-gynt1-morning-mood.mp3");
Uri uri = Uri.parse(Uri.encode(url));
cambiarCancion(uri);
}
});

Button btnParar = (Button)findViewById(R.id.UD2_02_btnPararRepr);
btnParar.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
if (mediaplayer.isPlaying())
mediaplayer.stop();
```

```
pause=false;
}
});

Button btnSair = (Button) findViewById(R.id.UD2_02_btnSairAplicacion);
btnSair.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
finish();
}
});


}

    @Override
    protected void onPause() {
        super.onPause();

        if (mediaplayer.isPlaying()){
        mediaplayer.pause();
        pause = true;
        }
    }

    @Override
    protected void onResume() {
        super.onResume();

        if (pause) {
        mediaplayer.start();
        pause = false;
        }
    }


    @Override
    protected void onSaveInstanceState(Bundle estado) {
            estado.putBoolean("MEDIAPLAYER_PAUSE", pause);
            super.onSaveInstanceState(estado);
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
            super.onRestoreInstanceState(savedInstanceState);
            savedInstanceState.putBoolean("MEDIAPLAYER_PAUSE", false);
            pause = savedInstanceState.getBoolean("MEDIAPLAYER_PAUSE");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        if (mediaplayer.isPlaying()) mediaplayer.stop();

        if (mediaplayer != null) mediaplayer.release();
        mediaplayer = null;

    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ud2_02__multimedia_reproductor);

        mediaplayer = new MediaPlayer();
        pause = false;
        xestionarEventos();


    }
```

```
}
```

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