

1 PDM Avanzado Captura de Audio

1.1 Sumario

- 1 Introducion
- 2 Clase **MediaRecorder**
 - ◆ 2.1 Caso práctico
 - ◇ 2.1.1 Creamos a Activity

1.2 Introducion

Nesta parte imos aprender como capturar audio dende Android.

A clase que imos utilizar é a **Clase MediaRecorder**.

Podedes ver os formatos de audio soportados en: <http://developer.android.com/guide/appendix/media-formats.html>

Nota: Poderíamos usar a clase **AudioRecord** pero como non imos procesar e analizar o audio capturado e polo tanto non a usaremos.

Para facelo necesitaremos engadir unha serie de permisos ó arquivo AndroidManifest.xml.

- Permiso para capturar audio.

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

- Permiso para gardar o audio capturado na tarxeta SD.

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

1.3 Clase MediaRecorder

Os pasos que temos que dar para capturar audio son os seguintes:

- Crear un obxecto da clase **MediaRecorder**.

```
private MediaRecorder mediaRecorder;  
.....  
mediaRecorder = new MediaRecorder();
```

- Establecer de onde vén a fonte de audio. O normal é que sexa do MIC da cámara web. Isto se fai chamando ó método **setAudioSource(fonte)** onde fonte é unha constante que atopades en **MediaRecorder.AudioSource.XXXXX**.

```
mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
```

- Establecer o formato de saída, chamando ó método **setOutputFormat(formato)** onde formato é unha constante que se atopa en **MediaRecorder.OutputFormat.XXX**

```
mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
```

Type	Format / Codec	Encoder	Decoder	Details	Supported File Type(s) / Container Formats
Audio	AAC LC	*	*	Support for mono/stereo /5.0/5.1 content with standard sampling rates from 8 to 48 kHz.	<ul style="list-style-type: none"> • 3GPP (.3gp) • MPEG-4 (.mp4, .m4a) • ADTS raw AAC (.aac, decode in Android 3.1+, encode in Android 4.0+, ADIF not supported) • MPEG-TS (.ts, not seekable, Android 3.0+)
	HE-AACv1 (AAC+)	*	*		
	HE-AACv2 (enhanced AAC+)		*	Support for stereo/5.0/5.1 content with standard sampling rates from 8 to 48 kHz.	
	AAC ELD (enhanced low delay AAC)	*	*	Support for mono/stereo content with standard sampling rates from 16 to 48 kHz	
	AMR-NB	*	*	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)
	AMR-WB	*	*	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)
	FLAC		*	Mono/Stereo (no multichannel). Sample rates up to 48 kHz (but up to 44.1 kHz is recommended on devices with 44.1 kHz output, as the 48 to 44.1 kHz downsampler does not include a low-pass filter).	FLAC (.flac) only

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

A columna de **Supported File Type** indica o formato de saída. No noso caso usaremos `MediaRecorder.OutputFormat.TREE_GPP`.

- **Optativo:** Establecer o tempo máximo de gravación chamando o método `setMaxDuration(int msec)` do obxecto `MediaRecorder`.

Nos imos poñer 10000 (10 segundos).

```
mediaRecorder.setMaxDuration(10000);
```

- **Optativo:** Indicar a tasa de bit's, chamando ó método `setAudioEncodingBitRate(int valor)`.

Os permitidos o tedes no [na imaxe anterior](#).

Nota: A cantidade de bit's vai establecer a calidade do audio capturado.

Algúns valores típicos:

- Tasas de bits de compresión a MP3:

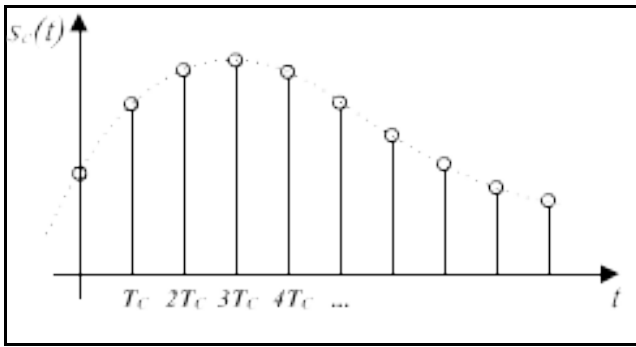
4 kbit/s Mínimo para recoñecer a fala.
 8 kbit/s Calidade telefónica convencional
 32 kbit/s Radio AM
 96 kbit/s Radio FM
 128 kbit/s Son calidade semi CD, muy común en MP3
 192 kbit/s Son calidade CD en formato MP3
 320 kbit/s Máxima calidade para formato MP3

Nos imos probar con $32 * 1024 = 32768$

```
mediaRecorder.setAudioEncodingBitRate(32768);
```

- **Optativo:** Indicar a frecuencia de muestreo, chamando ó método `setAudioSamplingRate(int valor)`.

ATENCIÓN: No emulador este valor non pode ser superior a 8000.



Máis información: http://es.wikipedia.org/wiki/Frecuencia_de_muestreo

Un exemplo, no caso de calidade CD o sample rate é de 44,1Khz. No noso caso o valor estará limitado pola táboa de formatos de audio. Por exemplo AAC está limitado entre 8Khz e 96Khz, AMR_NB só a 8Khz e AMR_WB a 16Khz.

Nos imos probar con 8000 (8Khz) pola limitación do emulador.

```
mediaRecorder.setAudioSamplingRate(8000); // No emulador só 8000
```

- Indicar a codificación de audio utilizada chamando ó método `setAudioEncoder(método)` onde método é unha constante que se atopa en `MediaRecorder.AudioEncoder.XXX`.

No noso caso imos utilizar `MediaRecorder.AudioEncoder.AAC`.

```
mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
```

- Indicar o arquivo a gravar, indicando a ruta nome e extensión do arquivo, chamando ó método `setOutputFile(arquívogravar)`.

Aquí temos varias posibilidades. Unha delas é darlle coma valor un String no que vai a ruta e o nome de arquivo e extensión. A ruta será algún cartafol da SDCARD. Para obter esa ruta faremos uso da clase `Environment`.

Para saber onde gardar o arquivo tedes que facer uso da clase `Environment`: <http://developer.android.com/reference/android/os/Environment.html>

Exemplos:

```
Environment.getExternalStorageDirectory().getAbsolutePath()
```

Con isto obtemos a ruta a SDCARD (raíz).

```
String state = Environment.getExternalStorageState();
if (!state.equals(Environment.MEDIA_MOUNTED)) {
    // A TARXETA NON ESTA MONTADA
}
```

Con isto sabemos se a tarxeta SD está montada.

Nota: Tamén podemos crear a ruta na SDCARD, facendo uso de obxectos da clase `File`, ou comprobando se existe o cartafol. O veremos máis adiante.

```
private String arquivoGravar;
.....
String timeStamp = DateFormat.getDateTimeInstance().format(new Date()).replaceAll(":", "").replaceAll("/", "_").replaceAll(" ", "_");

arquivoGravar = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC) + timeStamp + ".3gp";

mediaRecorder.setOutputFile(arquivoGravar);
```

Neste exemplo creamos un nome para o arquivo gravado baseado na data-hora e o vai gardar no cartafol da SD Externa que o S.O. reserva para gardar música.

- Chamar ó método `prepare()`. Pode producir excepcións (debe ir con `try...catch`).

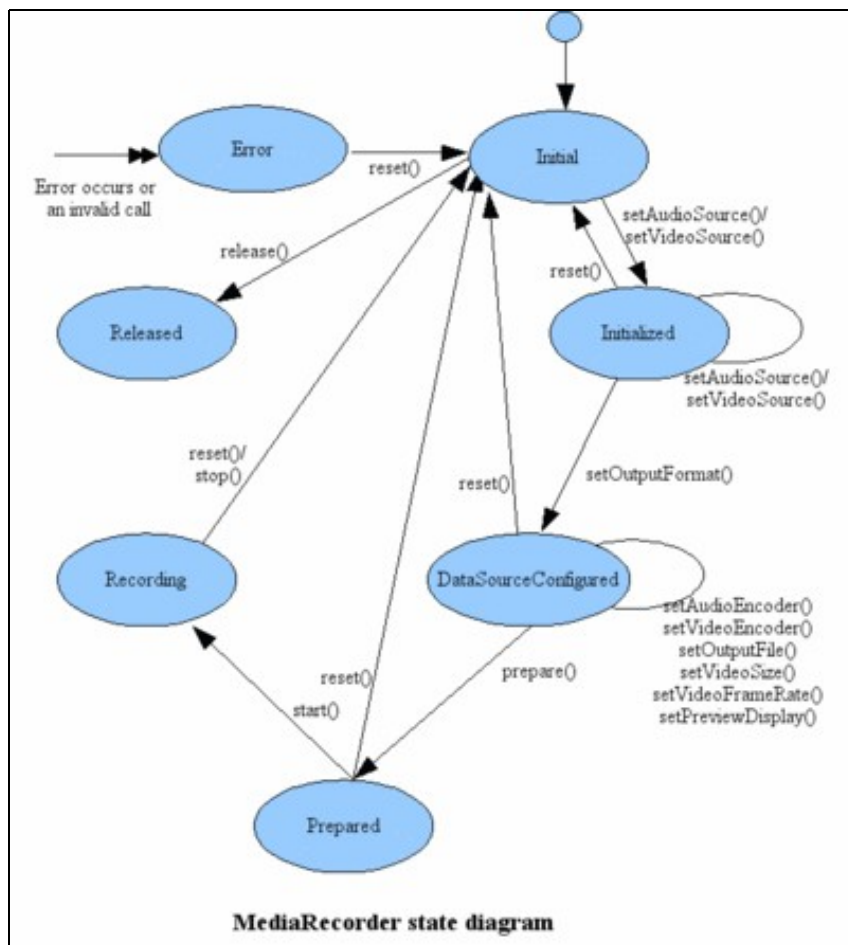
```
mediaRecorder.prepare();
```

- Chamar ó método `start()`.

```
mediaRecorder.start();
```

Neste intre se procede a gravar no cartafol indicado. A gravación non parará ata que chamemos ó método `stop` do `MediaRecorder` ou se acade o tempo máximo de gravación.

É importante seguir a orden, xa que por exemplo, a chamada a `setAudioEncoder` non pode ir antes de `setOutputFormat` (mirade o diagrama de estados seguinte). Ó igual que o `MediaPlayer`, o `MediaRecorder` ten un diagrama de estados que xa non debería ter dificultade en entenderse:



Imaxe obtido de <http://developer.android.com/reference/android/media/MediaRecorder.html>

Cando rematemos de gravar (por tempo) ou se paramos nos premendo o botón correspondente se recomenda liberar os recursos do `MediaRecorder`.

O faremos así:

- Chamando o método `stop`.
- Chamando o método `release`.
- Igualando a null o obxecto da clase `MediaRecorder`.

Tamén deberemos liberar os recursos se cambiamos de aplicación ou se a pechamos.

1.3.1 Caso práctico

O obxectivo desta práctica é gravar un arquivo de audio no cartafol da SD Externa reservado á música. Ó premer o botón Gravar amosaremos unha caixa de diálogo indicando que se está a gravar e o saír de dita caixa pararemos de gravar. Ó premer o botón Reproducir reproduciremos o arquivo gardado.



1.3.1.1 Creamos a Activity

- Nome do proxecto: **UD2_03_MultimediaGravador**
- Nome da activity: **UD2_03_MultimediaGravador.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.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="${relativePackage}.${activityClass}" >

    <Button
        android:id="@+id/UD2_03_btnGravar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="GRAVAR AUDIO" />

    <Button
        android:id="@+id/UD2_03_btnReproducir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/UD2_03_btnGravar"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="26dp"
        android:text="REPRODUCIR AUDIO" />

</RelativeLayout>
```

Código da clase UD2_03_MultimediaGravador

Obxectivo: Amosar como gravar un arquivo de audio.

```
import java.io.File;
```

```

import java.text.DateFormat;
import java.util.Date;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class UD2_03_MultimediaGravador extends Activity {

    private MediaRecorder mediaRecorder;
    private String archivoGravar;

    private void abrirDialogo(String tipo){
        if (tipo == "GRAVAR") {
            AlertDialog.Builder dialog = new AlertDialog.Builder(this)
                .setMessage("GRAVANDO").setPositiveButton(
                    "PREME PARA PARAR",
                    new DialogInterface.OnClickListener() {

                        @Override
                        public void onClick(DialogInterface dialog,
                            int which) {
                            // TODO Auto-generated method stub
                            mediaRecorder.stop();
                            mediaRecorder.release();
                            mediaRecorder = null;
                        }
                    });
            dialog.show();
        }

        if (tipo == "REPRODUCIR") {

            if ((archivoGravar==null) | archivoGravar=="") return;
            File archivo = new File(archivoGravar);
            if (!archivo.exists()) return;

            final MediaPlayer mediaPlayer = new MediaPlayer();
            try {
                mediaPlayer.setDataSource(archivoGravar);
                mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
                mediaPlayer.prepare();
                mediaPlayer.start();
            } catch (Exception e) {
                Toast.makeText(getApplicationContext(),
                    "ERRO:" + e.getMessage(), Toast.LENGTH_LONG).show();
            }

            AlertDialog.Builder dialog = new AlertDialog.Builder(this)
                .setMessage("REPRODUCINDO").setPositiveButton(
                    "PREME PARA PARAR",
                    new DialogInterface.OnClickListener() {

                        @Override
                        public void onClick(DialogInterface dialog,
                            int which) {
                            // TODO Auto-generated method stub
                            mediaPlayer.stop();
                            mediaPlayer.release();
                        }
                    });
            dialog.show();
        }
    }
}

```

```

}

}

private void xestionarEventos(){

    Button btnGravar = (Button)findViewById(R.id.UD2_03_btnGravar);
    btnGravar.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            String timeStamp = DateFormat.getDateInstance().format(
                new Date()).replaceAll(":", "").replaceAll("/", "_")
                .replaceAll(" ", "_");

            mediaRecorder = new MediaRecorder();
            arquivoGravar = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC) + timeStamp + ".3gp";
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
            mediaRecorder.setMaxDuration(10000);
            mediaRecorder.setAudioEncodingBitRate(32768);
            mediaRecorder.setAudioSamplingRate(8000); // No emulador só 8000 coma
            mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
            mediaRecorder.setOutputFile(arquivoGravar);
            try {
                mediaRecorder.prepare();
            } catch (Exception e) {
                // TODO Auto-generated catch block
                mediaRecorder.reset();
            }
            mediaRecorder.start();
            abrirDialogo("GRAVAR");

        }
    });

    Button btnReproducir = (Button)findViewById(R.id.UD2_03_btnReproducir);
    btnReproducir.setOnClickListener(new OnClickListener() {

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

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ud2_03__multimedia_gravador);

    xestionarEventos();

}
}

```

- Liña 20: Definimos o MediaRecorder.
- Liña 21: Propiedade que garda o nome e ruta do arquivo gravado. Necesario para cando prememos o botón de reproducir.
- Liñas 25-41: Diálogo que aparece cando prememos sobre o botón de Gravar. Se prememos o botón de Aceptar dentro do diálogo paramos de gravar e liberamos o MediaRecorder.
- Liñas 43-75: Diálogo que aparece cando prememos sobre o botón de Reproducir. Xa visto no punto anterior desta Unidade Didáctica.
- Liñas 91-107: Xestión do evento click sobre o botón de gravar. Preparamos o MediaRecorder e chamamos a abrir o diálogo de gravación.
- Liñas 117-120: Xestión do evento click sobre o botón de reproducir. Abrimos o diálogo de Reproducir.

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