

Media

Créditos

- Tutorial

<https://developer.android.com/guide/topics/media/index.html>¹

¹ Content is licensed under Creative Commons Attribution 2.5

Media

- Reproducir música.
- Grabar audio.
- Reproducir video.
- Tomar fotografías.
- Grabar video.

Tocar música

- Clase *MediaPlayer*.
- Varios formatos (mp3, ogg, etc.)
- Lista completa en:
<https://developer.android.com/guide/appendix/media-formats.html>.
- Varias fuentes de música:
 - Remoto (servidor).
 - Local (proyecto, carpeta */res/raw*).
 - Local (teléfono, memoria interna o SD externa).

Permisos

- Para archivos remotos:

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

- Para archivos en la memoria interna:

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

Archivo en */res/raw*

```
MediaPlayer mediaPlayer = MediaPlayer.create(context, R.raw.sound_file_1);  
mediaPlayer.start(); // no need to call prepare(); create() does that for you
```

Archivo remoto

```
String url = "http://....."; // your URL here
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
mediaPlayer.setDataSource(url);
mediaPlayer.prepare(); // might take long! (for buffering, etc)
mediaPlayer.start();
```

- **Nota:** *prepare()* puede tardar y generar un error ANR (Application Not Responding)

Preparación asíncrona

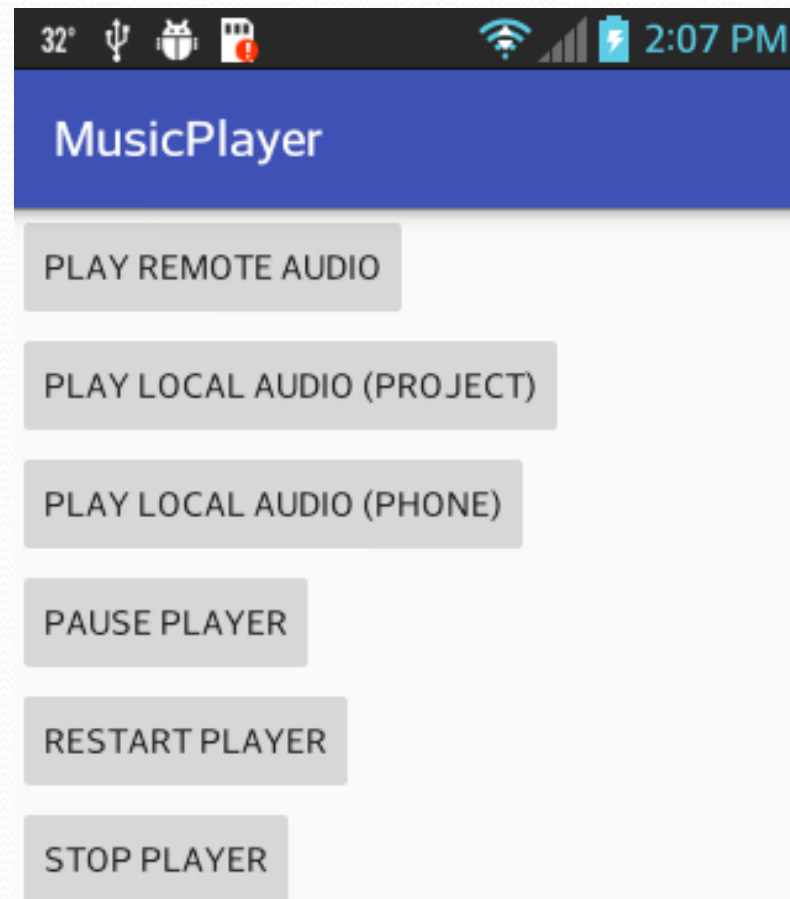
- Usar *prepareAsync()* en vez de *prepare()*.
- *prepareAsync()* se ejecuta en otro hilo.
- Invoca al callback *onPrepare()* al terminar.

Métodos de *MediaPlayer*

- *start()* – comienza a tocar la canción.
- *pause()* – hace una pausa.
- *seekTo(int msec)* – comienza a tocar la canción en el milisegundo dado por *msec*.
- *stop()* – deja de tocar la canción actual.
- *release()* – libera la memoria al final.

```
mediaPlayer.release();  
mediaPlayer = null;
```

Ejemplo



Permisos

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

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tesi.musicplayer.MainActivity">

    <Button
        android:id="@+id/btnPlayRemote"
        android:text="Play Remote Audio"
        android:onClick="buttonOnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    // Los demás botones son iguales

</LinearLayout>
```

Actividad principal

```
public class MainActivity extends AppCompatActivity
{
    implements MediaPlayer.OnPreparedListener {
    private static final String REMOTE_AUDIO_PATH =
        "http://euler.mat.uson.mx/~havillam/android/Programas/tab_anitek_reconstruct.mp3";
    private static final int LOCAL_PROJECT_AUDIO_ID = R.raw.tab_anitek_reconstruct;
    private static final String LOCAL_PHONE_AUDIO_PATH =
        Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC) +
        "/tab_anitek_reconstruct.mp3";

    private MediaPlayer mediaPlayer;
    private int playbackPosition = 0;

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

Callback de los botones

```
public void buttonOnClick(View view)
{
    switch (view.getId()) {

        case R.id.btnPlayRemote:
            playRemoteAudio(REMOTE_AUDIO_PATH);
            break;

        case R.id.btnPlayLocalProject:
            playLocalProjectAudio(LOCAL_PROJECT_AUDIO_ID);
            break;

        case R.id.btnPlayLocalPhone:
            playLocalPhoneAudio(LOCAL_PHONE_AUDIO_PATH);
            break;

        case R.id.btnPause:
            playerPause();
            break;

        case R.id.btnRestart:
            playerRestart();
            break;

        case R.id.btnStop:
            playerStop();
            break;

    }
}
```

Archivo remoto

```
private void playRemoteAudio(String url)
{
    killMediaPlayer();

    try {
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mediaPlayer.setDataSource(url);
        mediaPlayer.setOnPreparedListener(this);
        mediaPlayer.prepareAsync();
    }
    catch (Exception e) {
        Toast.makeText(this, "Error with url: " + url + " " + e,
            Toast.LENGTH_LONG).show();
    }
}
```

Archivo local (proyecto)

```
private void playLocalProjectAudio(int audioId)
{
    try {
        mediaPlayer = MediaPlayer.create(this, audioId);
        mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
        mediaPlayer.start();
    }
    catch (Exception e) {
        Toast.makeText(this, "Error: " + e, Toast.LENGTH_LONG).show();
    }
}
```

Archivo local (teléfono)

```
private void playLocalPhoneAudio(String audioPath)
{
    try {
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setDataSource(audioPath);
        mediaPlayer.prepare();
        mediaPlayer.start();
    }
    catch (Exception e) {
        Toast.makeText(this, "Error: " + e, Toast.LENGTH_LONG).show();
        Log.d("xyz", "Path: <" + audioPath + "> error: " + e.toString());
    }
}
```

Pausa, restart y stop

```
private void playerPause()
{
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        playbackPosition = mediaPlayer.getCurrentPosition();
        mediaPlayer.pause();
    }
}

private void playerRestart()
{
    if (mediaPlayer != null && !mediaPlayer.isPlaying()) {
        mediaPlayer.seekTo(playbackPosition);
        mediaPlayer.start();
    }
}

private void playerStop()
{
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        playbackPosition = 0;
    }
}
```

onPrepare y liberar el player

```
public void onPrepared(MediaPlayer player)
{
    player.start();
}

protected void onDestroy()
{
    super.onDestroy();
    killMediaPlayer();
}

private void killMediaPlayer()
{
    if (mediaPlayer != null) {
        try {
            mediaPlayer.release();
        }
        catch (Exception e) {
            Toast.makeText(this, "Error releasing mediaPlayer: " + e,
                           Toast.LENGTH_LONG).show();
        }
    }
}
```

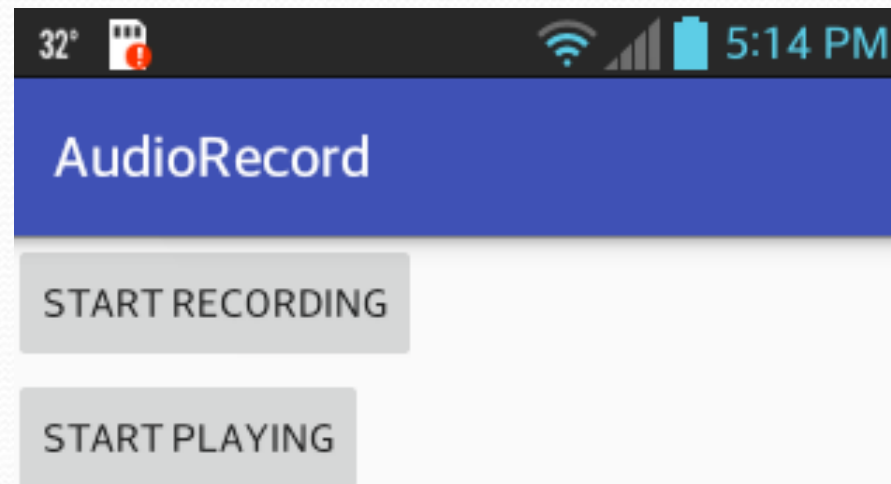
Grabar audio

1. Crear un objeto *MediaRecorder*.
2. Definir la fuente de audio con *setAudioSource()*, p.e. *MediaRecorder.AudioSource.MIC*.
3. Definir el formato de salida con *setOutputFormat()*, p.e. 3GP (*MediaRecorder.OutputFormat.THREE_GPP*)
4. Definir el nombre del archivo de salida con *setOutputFile()*.
5. Definir el codificador con *setAudioEncoder()*, p.e. AMR Audio Encoder (*MediaRecorder.AudioEncoder.AMR_NB*).

Grabar audio

6. Llamar a *prepare()*.
7. Llamar a *start()* para comenzar a grabar.
8. Llamar a *stop()* para dejar de grabar.
9. Llamar al *release()* para liberar memoria.

Ejemplo



Permisos

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

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tesi.audiorecord.MainActivity">

    <Button
        android:id="@+id/btnRecord"
        android:text="Start recording"
        android:onClick="recordOnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnPlay"
        android:text="Start playing"
        android:onClick="playOnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Actividad principal

```
public class MainActivity extends AppCompatActivity {
    private static final String LOG_TAG = "xyz";
    private static String fileName = null;

    private MediaRecorder recorder;
    private MediaPlayer player;

    private boolean startRecording = true;
    private boolean startPlaying = true;

    public MainActivity()
    {
        fileName = Environment.getExternalStorageDirectory().getAbsolutePath();
        fileName += "/audiorecordtest.3gp";
    }

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

Callbacks

```
public void recordOnClick(View view)
{
    Button button = (Button) view;
    onRecord(startRecording);
    if (startRecording) {
        button.setText("Stop recording");
    }
    else {
        button.setText("Start recording");
    }
    startRecording = !startRecording;
}

public void playOnClick(View view)
{
    Button button = (Button) view;
    onPlay(startPlaying);
    if (startPlaying) {
        button.setText("Stop playing");
    }
    else {
        button.setText("Start playing");
    }
    startPlaying = !startPlaying;
}
```

Grabar audio

```
private void onRecord(boolean start)
{
    if (start) {
        startRecording();
    }
    else {
        stopRecording();
    }
}

private void startRecording()
{
    recorder = new MediaRecorder();
    recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    recorder.setOutputFile(fileName);
    recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    try {
        recorder.prepare();
    }
    catch (IOException e) { }
    recorder.start();
}

private void stopRecording()
{
    recorder.stop();
    recorder.release();
    recorder = null;
}
```

Reproducir audio

```
private void onPlay(boolean start)
{
    if (start) {
        startPlaying();
    }
    else {
        stopPlaying();
    }
}

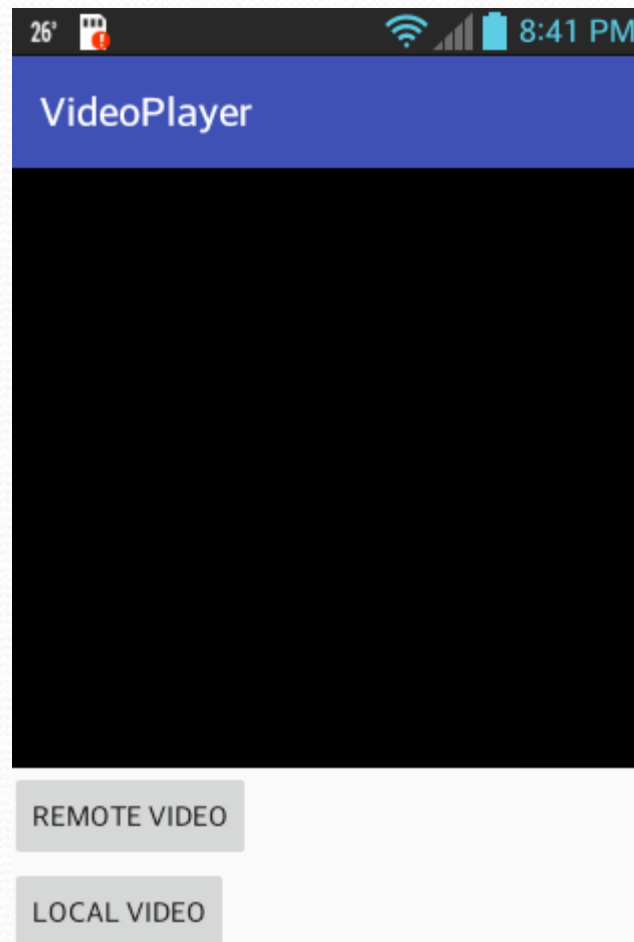
private void startPlaying()
{
    player = new MediaPlayer();
    try {
        player.setDataSource(fileName);
        player.prepare();
        player.start();
    }
    catch (IOException e) { }
}

private void stopPlaying()
{
    player.release();
    player = null;
}
```

Reproducir video

- Clases utilizadas:
- *VideoView* – despliega un archivo de video.
- *MediaController* – view que contiene controles para el *VideoView* como **Play/Pause**, **Rewind**, etc.
- El video puede ser remoto o local.

Ejemplo



Permisos

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

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tesi.videoplayer.MainActivity">

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="300dp" />

    <Button
        android:id="@+id/btnRemote"
        android:text="Remote video"
        android:onClick="onClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <Button
        android:id="@+id/btnLocal"
        android:text="Local video"
        android:onClick="onClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Actividad principal

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

Callback del botón

```
public void onClick(View view)
{
    File path = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES);

    VideoView videoView = (VideoView) findViewById(R.id.videoView);
    MediaController controller = new MediaController(this);
    videoView.setMediaController(controller);
    try {
        if (view.getId() == R.id.btnRemote) {
            videoView.setVideoURI(Uri.parse("http://euler.mat.uson.mx/~havillam/android/" +
                "Programas/video.mp4"));
        }
        else {
            videoView.setVideoPath(path + "/video.mp4");
        }
        videoView.requestFocus();
        videoView.start();
    }
    catch (Exception e) {
        Log.d("xyz", "Error: " + e);
        Toast.makeText(this, e.toString(), Toast.LENGTH_LONG).show();
    }
}
```

Mas información

- Media Playback.

<https://developer.android.com/guide/topics/media/mediaplayer.html>

- Audio Capture.

<https://developer.android.com/guide/topics/media/audio-capture.html>