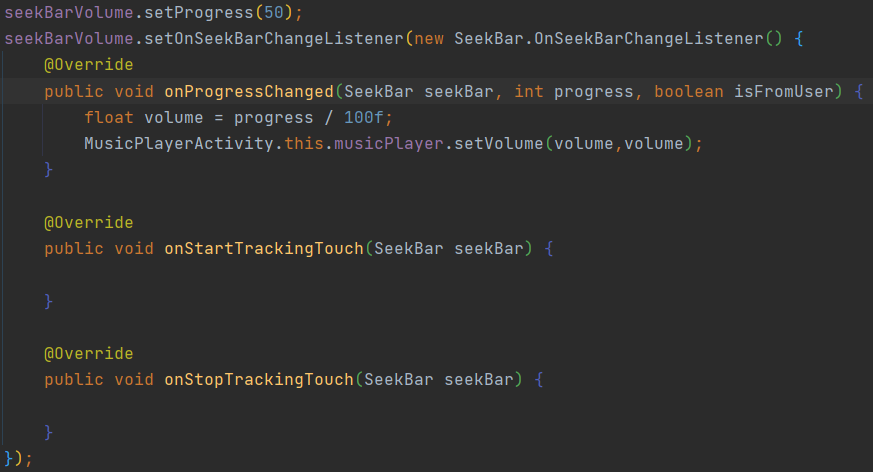
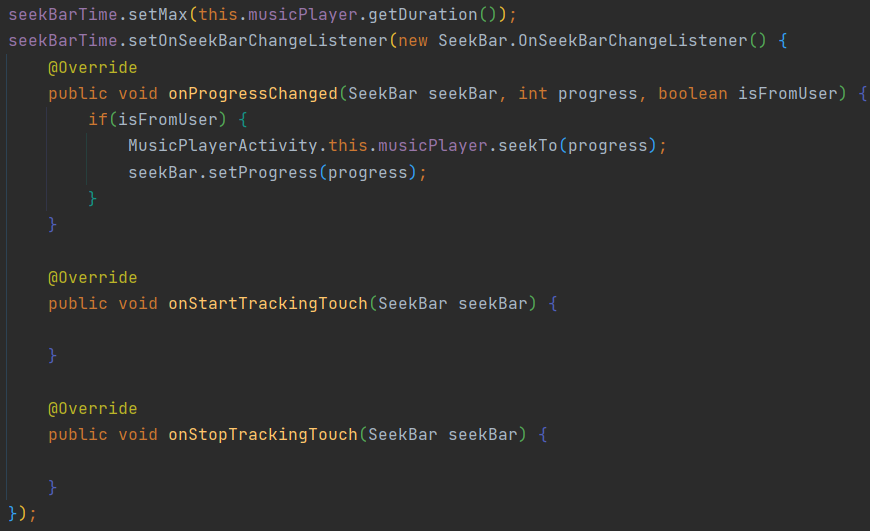
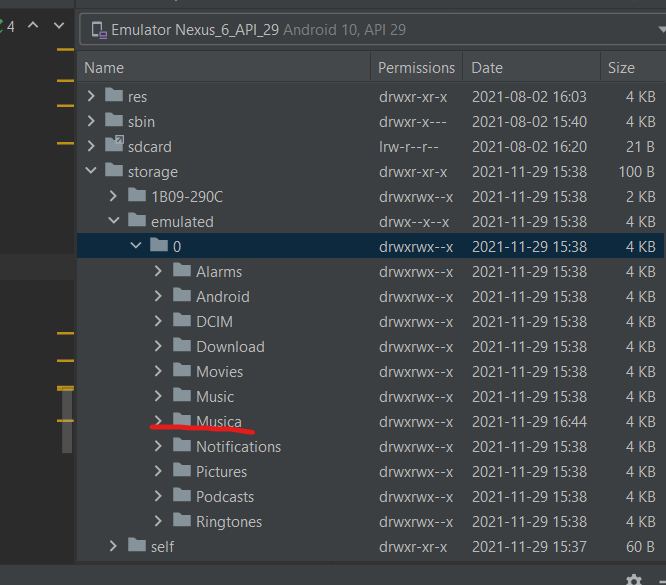
**Índice**

**Código para crear barra de sonido y barra de progreso**





**Creación de carpeta en el dispositivo virtual para probar**



Desarrollo de Aplicaciones Multiplataforma

Actividad 03

Aplicaciones multimedia

Programación multimedia y dispositivos móviles

Actividad

Aplicaciones multimedia.

**Objetivos**

Desarrollar Aplicaciones para teléfono móvil.

Usar elementos multimedia.

Reproducir audio y video.

|  |
| --- |
| **¿Cómo lo hago?** |
| 1. Rellena los datos que se piden en la tabla “Antes de empezar”. 2. Haz uso de fuentes comunes como Arial, Calibri, Times New Roman etc. 3. Utiliza el color negro para desarrollar tus respuestas y usa otros colores para destacar contenidos o palabras que creas necesario resaltar. 4. Lee detenidamente todos los pasos a continuación. Realiza todas las acciones indicadas y responde a todas las preguntas planteadas. 5. Deberás realizar una aplicación **JAVA** que incluya todas las clases y métodos necesarios para dar respuesta a todos los ítems marcados en el enunciado. 6. *La entrega de la actividad estará formada por el proyecto que hayas creado en* ***NetBeans*** *como solución de la actividad, comprimido en formato* ***.zip****.* 7. Recuerda nombrar el archivo siguiendo estas indicaciones:  * Ciclo\_Módulo o crédito\_Tema\_ACT\_número actividad\_Nombre y apellido   + Ejemplo: AF\_M01\_T01\_ACT\_01\_Maria Garcia |

|  |  |
| --- | --- |
| **Antes de empezar…** | |
| Nombre |  |
| Apellidos |  |
| Módulo/Crédito |  |
| UF (solo ciclos LOE) |  |
| Título de la actividad |  |

**MediaPlayer**

Desarrollar una aplicación Android para reproducir archivos de audio almacenados en la memoria del teléfono o dispositivo Android:

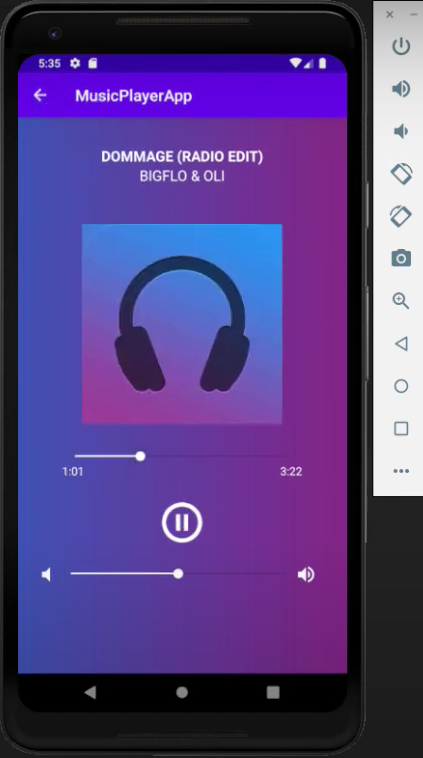
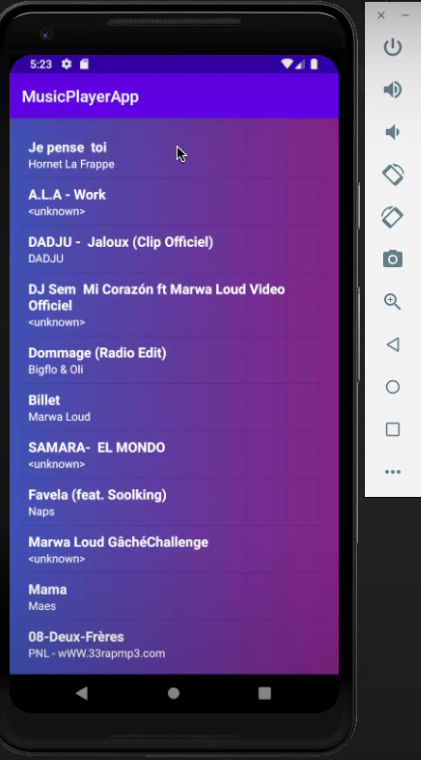
* Dependiendo del tipo de dispositivo, utilizar la memoria interna o la memoria externa (tarjeta SD card).
* Crear una carpeta llamada música y almacenar en ella varios archivos de audio (formato mp3)
* Presentar por pantalla una lista de los archivos de audio disponibles en dicha carpeta y permitir al usuario elegir uno de ellos y reproducirlo.
* Agregar los clásicos controles de Play, Pause y Stop.

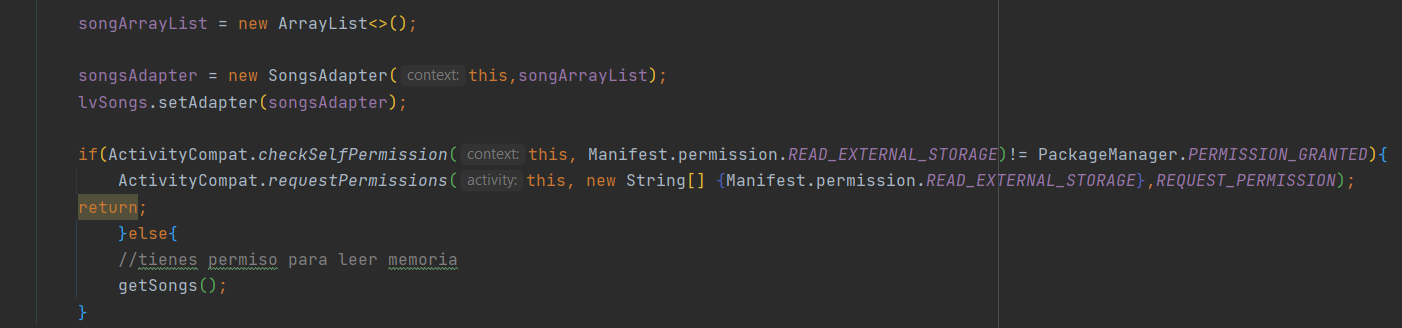
La App debe ser intuitiva y fácil de usar y debe evitar cualquier comportamiento inestable por información errónea o no proporcionada.

yo he podido acceder a la carpeta músi

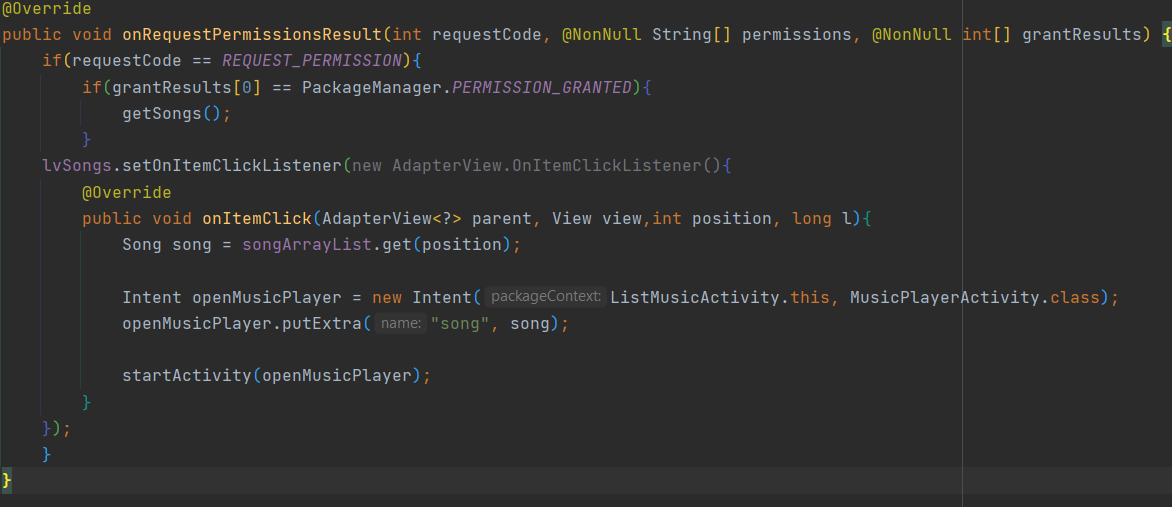
**Listado de canciones de mi dispositivo y reproductor en funcionamiento**

Finalmente la lista de canciones la he hecho con listView ya que el recyclerView me dio algunos problemas y no me mostraba bien las canciones con el adaptador.

**Arraylist con adaptador y permisos para buscar canciones en el dispositivo**

****

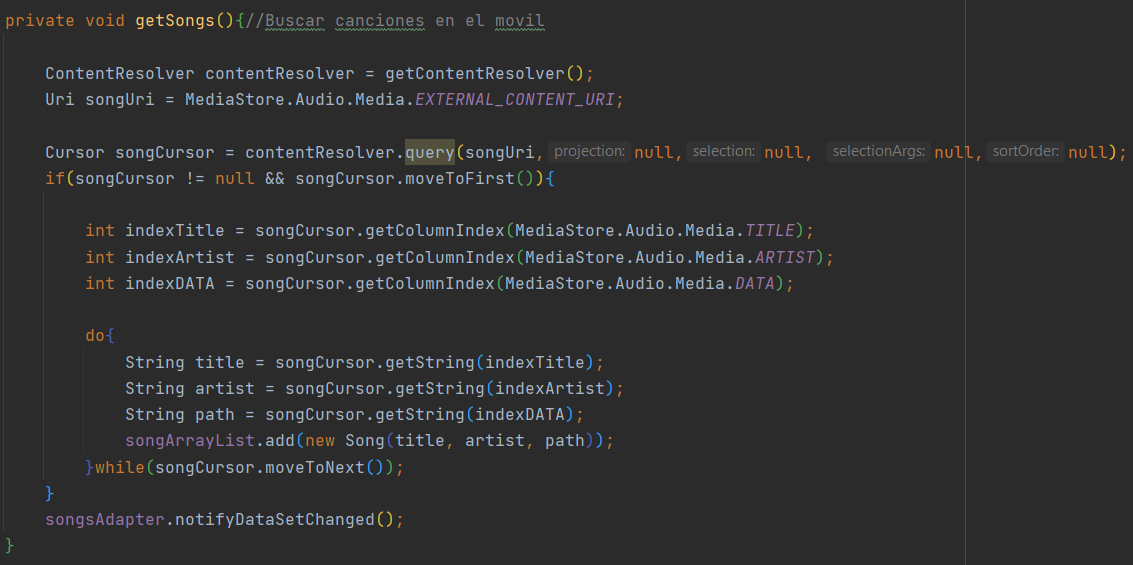
**Sobreescribo el método para iniciar la activity después de los permisos**

****

**Creo inflater para layout**

****

**Código**

**Metodo para buscar canciones en el móvil**

**Codigo Manifest**

*<?*xml version="1.0" encoding="utf-8"*?>  
<*manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.ndroid.musicplayerapp"*>  
  
 <*uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" */>  
  
 <*application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"*>  
 <*activity android:name=".MusicPlayerActivity"*>  
 <*intent-filter*>  
 <*action android:name="android.intent.action.MAIN" */>  
  
 <*category android:name="android.intent.category.LAUNCHER" */>  
 </*intent-filter*>  
 </*activity*>  
 </*application*>  
  
</*manifest*>*

**Codigo Clase ListMusic**

package com.ndroid.musicplayerapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import android.Manifest;  
import android.content.ContentResolver;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;  
import java.util.ArrayList;  
  
public class ListMusicActivity extends AppCompatActivity *{* private static final int *REQUEST\_PERMISSION* = 99;  
 ArrayList*<*Song*>* songArrayList;  
 ListView lvSongs;  
 SongsAdapter songsAdapter;  
  
 @Override  
 protected void onCreate*(*Bundle savedInstanceState*) {* super.onCreate*(*savedInstanceState*)*;  
 setContentView*(*R.layout.*activity\_list\_music)*;  
  
 lvSongs = findViewById*(*R.id.*lvSongs)*;  
  
 songArrayList = new ArrayList*<>()*;  
  
 songsAdapter = new SongsAdapter*(*this,songArrayList*)*;  
 lvSongs.setAdapter*(*songsAdapter*)*;  
  
 if*(*ActivityCompat.*checkSelfPermission(*this, Manifest.permission.*READ\_EXTERNAL\_STORAGE)*!= PackageManager.*PERMISSION\_GRANTED){* ActivityCompat.*requestPermissions(*this, new String*[] {*Manifest.permission.*READ\_EXTERNAL\_STORAGE}*,*REQUEST\_PERMISSION)*;  
 return;  
 *}*else*{* //tienes permiso para leer memoria  
 getSongs*()*;  
 *}  
 }* @Override  
 public void onRequestPermissionsResult*(*int requestCode, @NonNull String*[]* permissions, @NonNull int*[]* grantResults*) {* if*(*requestCode == *REQUEST\_PERMISSION){* if*(*grantResults*[*0*]* == PackageManager.*PERMISSION\_GRANTED){* getSongs*()*;  
 *}* lvSongs.setOnItemClickListener*(*new AdapterView.OnItemClickListener*(){* @Override  
 public void onItemClick*(*AdapterView*<*?*>* parent, View view,int position, long l*){* Song song = songArrayList.get*(*position*)*;  
  
 Intent openMusicPlayer = new Intent*(*ListMusicActivity.this, MusicPlayerActivity.class*)*;  
 openMusicPlayer.putExtra*(*"song", song*)*;  
  
 startActivity*(*openMusicPlayer*)*;  
 *}  
 })*;  
 *}  
  
 }* private void getSongs*(){*//Buscar canciones en el movil  
  
 ContentResolver contentResolver = getContentResolver*()*;  
 Uri songUri = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
  
 Cursor songCursor = contentResolver.query*(*songUri,null,null, null,null*)*;  
 if*(*songCursor != null && songCursor.moveToFirst*()){* int indexTitle = songCursor.getColumnIndex*(*MediaStore.Audio.Media.*TITLE)*;  
 int indexArtist = songCursor.getColumnIndex*(*MediaStore.Audio.Media.*ARTIST)*;  
 int indexDATA = songCursor.getColumnIndex*(*MediaStore.Audio.Media.*DATA)*;  
  
 do*{* String title = songCursor.getString*(*indexTitle*)*;  
 String artist = songCursor.getString*(*indexArtist*)*;  
 String path = songCursor.getString*(*indexDATA*)*;  
 songArrayList.add*(*new Song*(*title, artist, path*))*;  
 *}*while*(*songCursor.moveToNext*())*;  
 *}* songsAdapter.notifyDataSetChanged*()*;  
 *}  
}*

**Codigo Clase Musicplayer**

package com.ndroid.musicplayerapp;  
  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.os.Environment;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Button;  
import android.widget.SeekBar;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.io.File;  
import java.io.IOException;  
import java.sql.Array;  
  
public class MusicPlayerActivity extends AppCompatActivity implements View.OnClickListener *{* //declaracion de vistas  
 TextView tvTime, tvDuration, tvTitle, tvArtist;  
 SeekBar seekBarTime, seekBarVolume;  
 Button btnPlay;  
  
 MediaPlayer musicPlayer;  
 private MenuItem item;  
  
  
 @Override  
 protected void onCreate*(*Bundle savedInstanceState*) {* super.onCreate*(*savedInstanceState*)*;  
 setContentView*(*R.layout.*activity\_main)*;  
  
 getSupportActionBar*()*.setDisplayHomeAsUpEnabled*(*true*)*;  
  
 Song song = *(*Song*)*getIntent*()*.getSerializableExtra*(*"song"*)*;  
  
 tvTime = findViewById*(*R.id.*tvTime)*;  
 tvDuration = findViewById*(*R.id.*tvDuration)*;  
 seekBarTime = findViewById*(*R.id.*seekBarTime)*;  
 seekBarVolume = findViewById*(*R.id.*seekBarVolume)*;  
 btnPlay = findViewById*(*R.id.*btnPlay)*;  
 tvTitle = findViewById*(*R.id.*tvTitle)*;  
 tvArtist = findViewById*(*R.id.*tvArtist)*;  
  
 tvTitle.setText*(*song.getTitle*())*;  
 tvArtist.setText*(*song.getArtist*())*;  
  
 this.musicPlayer = new MediaPlayer*()*;  
 try *{* musicPlayer.setDataSource*(*song.getPath*())*;  
 musicPlayer.prepare*()*;  
 *}* catch *(*IOException e*) {* e.printStackTrace*()*;  
 *}* this.musicPlayer.setLooping*(*true*)*;  
 this.musicPlayer.seekTo*(*0*)*;  
 this.musicPlayer.setVolume*(*0.5f, 0.5f*)*;  
  
 String duration = millisecondsToString*(*this.musicPlayer.getDuration*())*;  
 tvDuration.setText*(*duration*)*;  
  
 btnPlay.setOnClickListener*(*this*)*;  
  
 seekBarVolume.setProgress*(*50*)*;  
 seekBarVolume.setOnSeekBarChangeListener*(*new SeekBar.OnSeekBarChangeListener*() {* @Override  
 public void onProgressChanged*(*SeekBar seekBar, int progress, boolean isFromUser*) {* float volume = progress / 100f;  
 MusicPlayerActivity.this.musicPlayer.setVolume*(*volume,volume*)*;  
 *}* @Override  
 public void onStartTrackingTouch*(*SeekBar seekBar*) {  
  
 }* @Override  
 public void onStopTrackingTouch*(*SeekBar seekBar*) {  
  
 }  
 })*;  
  
 seekBarTime.setMax*(*this.musicPlayer.getDuration*())*;  
 seekBarTime.setOnSeekBarChangeListener*(*new SeekBar.OnSeekBarChangeListener*() {* @Override  
 public void onProgressChanged*(*SeekBar seekBar, int progress, boolean isFromUser*) {* if*(*isFromUser*) {* MusicPlayerActivity.this.musicPlayer.seekTo*(*progress*)*;  
 seekBar.setProgress*(*progress*)*;  
 *}  
 }* @Override  
 public void onStartTrackingTouch*(*SeekBar seekBar*) {  
  
 }* @Override  
 public void onStopTrackingTouch*(*SeekBar seekBar*) {  
  
 }  
 })*;  
  
  
 new Thread*(*new Runnable*() {* @Override  
 public void run*() {* while *(*MusicPlayerActivity.this.musicPlayer != null*) {* if*(*MusicPlayerActivity.this.musicPlayer.isPlaying*()) {* try *{* final double current = MusicPlayerActivity.this.musicPlayer.getCurrentPosition*()*;  
 final String elapsedTime = millisecondsToString*((*int*)* current*)*;  
  
 runOnUiThread*(*new Runnable*() {* @Override  
 public void run*() {* tvTime.setText*(*elapsedTime*)*;  
 seekBarTime.setProgress*((*int*)* current*)*;  
 *}  
 })*;  
  
 Thread.*sleep(*1000*)*;  
 *}*catch *(*InterruptedException e*) {}  
 }  
 }  
 }  
 })*.start*()*;  
  
 *}*   
  
  
 public String millisecondsToString*(*int time*) {* String elapsedTime = "";  
 int minutes = time / 1000 / 60;  
 int seconds = time / 1000 % 60;  
 elapsedTime = minutes+":";  
 if*(*seconds < 10*) {* elapsedTime += "0";  
 *}* elapsedTime += seconds;  
  
 return elapsedTime;  
 *}* @Override  
 public void onClick*(*View view*) {* if*(*view.getId*()* == R.id.*btnPlay) {* if*(*musicPlayer.isPlaying*()) {* // reproduciendo  
 musicPlayer.pause*()*;  
 btnPlay.setBackgroundResource*(*R.drawable.*ic\_play)*;  
 *}* else *{* // en pausa  
 musicPlayer.start*()*;  
 btnPlay.setBackgroundResource*(*R.drawable.*ic\_pause)*;  
 *}  
 }  
 }* @Override  
 public boolean onOptionsItemSelected*(*@NonNull MenuItem item*) {* if*(*item.getItemId*()* == android.R.id.*home){* finish*()*;  
 if*(*musicPlayer.isPlaying*()){* musicPlayer.stop*()*;  
 *}  
 }* return super.onOptionsItemSelected*(*item*)*;  
 *}  
  
  
}*

**Codigo clase Song**

package com.ndroid.musicplayerapp;  
  
import java.io.Serializable;  
  
public class Song implements Serializable *{* private String title;  
 private String artist;  
 private String path;  
  
 public Song*(*String title, String artist, String path*) {* this.title = title;  
 this.artist = artist;  
 this.path = path;  
 *}* public String getTitle*() {* return title;  
 *}* public void setTitle*(*String title*) {* this.title = title;  
 *}* public String getArtist*() {* return artist;  
 *}* public void setArtist*(*String artist*) {* this.artist = artist;  
 *}* public String getPath*() {* return path;  
 *}* public void setPath*(*String path*) {* this.path = path;  
 *}  
}*

**Codigo Clase SongsAdapter**

package com.ndroid.musicplayerapp;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
  
import java.util.List;  
  
public class SongsAdapter extends ArrayAdapter*<*Song*> {* private int resource;  
 public SongsAdapter*(*@NonNull Context context, @NonNull List*<*Song*>* objects*) {* super*(*context, 0, objects*)*;  
 this.resource = resource;  
 *}* @NonNull  
 @Override  
 public View getView*(*int position, @Nullable View convertView, @NonNull ViewGroup parent*){* convertView = LayoutInflater.*from(*parent.getContext*())*.inflate*(*R.layout.*item\_song*, null*)*;  
  
 TextView tvTitle = convertView.findViewById*(*R.id.*tvTitle)*;  
 TextView tvArtist = convertView.findViewById*(*R.id.*tvArtist)*;  
 Song song = getItem*(*position*)*;  
 tvArtist.setText*(*song.getArtist*())*;  
 tvTitle.setText*(*song.getTitle*())*;  
  
 return convertView;  
 *}  
}*

**Codigo layout activityListMusic**

*<?*xml version="1.0" encoding="utf-8"*?>  
<*LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp"  
 android:background ="@drawable/background"  
 tools:context=".ListMusicActivity" *>  
  
 <*ListView  
 android:id="@+id/lvSongs"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" */>  
  
</*LinearLayout*>*

**Codigo layout activity\_main**

*<?*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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/background"  
 android:gravity="center\_horizontal"  
 android:orientation="vertical"  
 android:padding="16dp"  
 tools:context=".MusicPlayerActivity"*>  
  
 <*TextView  
 android:id="@+id/tvArtist"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="Someone like you"  
 android:textAllCaps="true"  
 android:textColor="#fff"  
 android:textSize="18sp"  
 android:textStyle="bold" */>  
  
 <*TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:text="Adele"  
 android:textAllCaps="true"  
 android:textColor="#fff"  
 android:textSize="18sp" */>  
  
 <*ImageView  
 android:layout\_width="250dp"  
 android:layout\_height="250dp"  
 android:layout\_marginTop="30dp"  
 android:src="@drawable/icon\_music\_player" */>  
  
 <*SeekBar  
 android:id="@+id/seekBarTime"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:backgroundTint="#fff"  
 android:progressTint="#fff"  
 android:thumbTint="#fff" */>  
  
 <*RelativeLayout  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"*>  
  
 <*TextView  
 android:id="@+id/tvTime"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="00:00"  
 android:textColor="#fff" */>  
  
 <*TextView  
 android:id="@+id/tvDuration"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:text="00:00"  
 android:textColor="#fff" */>  
  
 </*RelativeLayout*>  
  
 <*Button  
 android:id="@+id/btnPlay"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="30dp"  
 android:background="@drawable/ic\_play" */>  
  
 <*LinearLayout  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:orientation="horizontal"*>  
  
 <*ImageView  
 android:layout\_width="20dp"  
 android:layout\_height="20dp"  
 android:src="@drawable/ic\_volume\_down" */>  
  
 <*SeekBar  
 android:id="@+id/seekBarVolume"  
 android:layout\_width="300dp"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#fff"  
 android:progressTint="#fff"  
 android:thumbTint="#fff" */>  
  
 <*ImageView  
 android:layout\_width="20dp"  
 android:layout\_height="20dp"  
 android:src="@drawable/ic\_volume\_up" */>  
 </*LinearLayout*>  
  
  
</*LinearLayout*>*

**Codigo layout itemSong**

*<?*xml version="1.0" encoding="utf-8"*?>  
<*LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/itsong"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="8dp"*>  
  
 <*TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Song 1"  
 android:textColor="#fff"  
 android:textStyle="bold"  
 android:textSize="17sp"*/>  
  
 <*TextView  
 android:id="@+id/tvArtist"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Song 1"  
 android:textColor="#fff"*/>  
  
</*LinearLayout*>*

ca y music de la SD con: Uri = MediaStore.Audio.Media.EXTERNAL\_CONTENT\_URI