



MediaPlayer

M08 Programación multimedia y dispositivos móviles

Actividad 03. UF2

Elsa Fernández Prado

Antes de empezar...	
Nombre	Elsa
Apellidos	Fernández Prado
Módulo/Crédito	DAM
UF (solo ciclos LOE)	01
Título de la actividad	MediaPlayer

INDICE

Objetivos: APP Reproductor de archivos de audio	2
Códigos principales	2-22
JAVA.....	2-14
MainActivity.java.....	2-4
Pistas.java.....	5
Adaptador.java.....	6-7
ReproductorActivity.java	7-14
XML.....	15-22
AndroidManifest.xml.	15
activity_main.xml.	16
elementalista.xml.....	17
activity_reproductor.xml.....	18-22
.....Carpeta drawable.....	22
Capturas de Pantalla.....	23-27
Carpeta musica y archivos de audio.....	23
ListView (ActivityMain)	23
Explicación elementos(activity_reproductor.xml)	24
Comprobación de los botones y controles	25-27

Objetivos:

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

CÓDIGOS PRINCIPALES

JAVA

MainActivity.java

```
package com.efp.reproducto30;
```

```
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import android.Manifest;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;
```

```

import android.widget.AdapterView;
import android.widget.ListView;
import java.io.File;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    //Permisos para acceder al almacenamiento externo
    private static String[] PERMISSIONS_STORAGE = {
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    };
    private static final int REQUEST_EXTERNAL_STORAGE = 1;

    //ELEMENTOS DEL ACTIVITY_MAIN.XML
    ListView lv_PlayList;

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

        //ELEMENTOS DEL ACTIVITY_MAIN.XML
        lv_PlayList= (ListView) findViewById(R.id.lv_PlayList);

        //Pedimos PERMISOS

        ActivityCompat.requestPermissions(MainActivity.this,PERMISSIONS_STORAGE,REQUEST_EXTERNAL_STORAGE);

        File directorio; // definimos el directorio del que queremos sacar las canciones
        directorio = new File(System.getenv("EXTERNAL_STORAGE") + "/musica");

        //creamos ArrayList de File para pasar a la 2ªActivity
        final ArrayList<File> canciones_efp = EncontrarCanciones(directorio); //
        Buscamos las canciones en nuestro directorio

        //creamos ArrayList de Pistas para el adaptador personalization
        ArrayList<Pistas> pistas = new ArrayList<Pistas>(); // creamos un arrayList para
        almacenar
        for (int pos = 0; pos < canciones_efp.size(); pos++) { //hacemos bucle a ficheros
        e ir insertando las pistas en n
            pistas.add(new Pistas(pos,
            canciones_efp.get(pos).getName().toString().replace(".mp3",""),canciones_efp.get(
            pos)));
        }
        // Definimos el adaptador para crear la lista

```

```

Adaptador adapter = new Adaptador(pistas, this);
ListView lv_PlayList = (ListView) findViewById(R.id.lv_PlayList);
lv_PlayList.setAdapter(adapter);

lv_PlayList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long
l) {
        startActivity(new Intent(getApplicationContext(),
ReproductorActivity.class).putExtra("pos", i).putExtra("canciones",
canciones_efp));
    }
});
}

/*
METODO ENCONTRAR CANCIONES. Busca los archivos mp3 dentro del
directorio que le hemos pasado y en las carpetas dentro de este
*/
public ArrayList<File> EncontrarCanciones(File directorio) { //le pasamos un
Path
    ArrayList<File> canciones_efp = new ArrayList<File>(); // Array donde
almacenaremos nuestros ficheros mp3
    File[] archivos = directorio.listFiles(); // creamos un array de ficheros que hay
en el directorio que le pasamos
    for (File lista : archivos) {
        if (lista.isDirectory() && !lista.isHidden()) { //si es una carpeta y no esta
oculta
            canciones_efp.addAll(EncontrarCanciones(lista)); //Recursividad,
volvemos llamar a nuestra funcion para que recorrer esa carpeta
        } else { //si no es carpeta
            if (lista.getName().endsWith(".mp3")) { //si es archivo mp3
                canciones_efp.add(lista); // y lo añadimos a nuestro array
            }
        }
    }
    return canciones_efp; // retorna arrayLis canciones
}

```

Pistas.java

```
package com.efp.reproducto30;

import java.io.File;

public class Pistas {
    String Nombre;
    File Audio;
    long Id;
    // Constructor
    public Pistas(long Id, String Nombre, File audio) {
        this.Id=Id;
        this.Nombre= Nombre;
        this.Audio= audio;
    }
    //Getters
    public String getNombre() {
        return Nombre;
    }
    public File getAudio() {
        return Audio;
    }
    public long getId() {
        return Id;
    }
}
```

Adaptador.java

```
package com.efp.reproducto30;

import android.annotation.SuppressLint;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.ArrayList;

public class Adaptador extends BaseAdapter {

    ArrayList<Pistas> pistas;
    Context c;

    public Adaptador(ArrayList<Pistas> pistas, Context c) {
        this.c = c;
        this.pistas = pistas;
    }

    @Override
    public int getCount() {
        return pistas.size();
    }

    @Override
    public Object getItem(int i) {
        return pistas.get(i);
    }

    @Override
    public long getItemId(int i) {
        return pistas.get(i).getId();
    }

    @Override
    public View getView(final int i, View view, ViewGroup viewGroup) {
```

```

LayoutInflater inflater = (LayoutInflater)
    c.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
@SuppressWarnings("ViewHolder") final View vista_elemento =
    inflater.inflate(R.layout.elementolista, viewGroup, false);
// Pongo el texto
TextView texto = (TextView)
vista_elemento.findViewById(R.id.tv_titCancion_efp);
texto.setText(pistas.get(i).Nombre);

return vista_elemento;
}
}

```

ReproductorActivity.java

```

package com.efp.reproductor30;

import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.io.File;
import java.util.ArrayList;

public class ReproductorActivity extends AppCompatActivity implements
View.OnClickListener {

    //DECLARACION DE VARIABLES
    //Elementos de activity_reproductor.xml
    ImageButton btn_PlayList_efp, imgBtn_anterior_efp, imgBtn_PlayPause_efp,
        imgBtn_atras_efp, imgBtn_adelante_efp, imgBtn_siguiente_efp;
    SeekBar sB_progresoCancion_efp, sB_volumen_efp;
    TextView tv_titCancion_efp, tv_tiempoQLleva_efp, tv_duracion_efp;
    ImageView iv_cancion_efp;

```


//Otras

```
Thread actualizarProgresoSeekBar;  
static MediaPlayer mp_efp;  
String aux="";  
ArrayList<File> canciones_efp;  
int posicion;  
Uri uri_efp;  
AudioManager audioManager_efp;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_reproductor);
```

//INICIALIZAMOS VARIABLES

//Elementos de activity_reproductor.xml

//Botones

```
    imgBtn_adelante_efp = (ImageButton)  
    findViewById(R.id.imgBtn_adelante_efp);  
    imgBtn_anterior_efp = (ImageButton) findViewById(R.id.imgBtn_anterior_efp);  
    imgBtn_atras_efp = (ImageButton) findViewById(R.id.imgBtn_atras_efp);  
    imgBtn_siguiente_efp = (ImageButton)  
    findViewById(R.id.imgBtn_siguiente_efp);  
    imgBtn_PlayPause_efp = (ImageButton)  
    findViewById(R.id.imgBtn_PlayPause_efp);  
    btn_PlayList_efp = (ImageButton) findViewById(R.id.btn_PlayList_efp);
```

//SeekBar

```
    sB_progresoCancion_efp = (SeekBar)  
    findViewById(R.id.sB_progresoCancion_efp);  
    sB_volumen_efp = (SeekBar) findViewById(R.id.sB_volumen_efp);
```

//TextView

```
    tv_titCancion_efp = (TextView) findViewById(R.id.tv_titCancion_efp);  
    tv_tiempoQLleva_efp = (TextView) findViewById(R.id.tv_tiempoQLleva_efp);  
    tv_duracion_efp = (TextView) findViewById(R.id.tv_duracion_efp);
```

//Imageview

```
    iv_cancion_efp = (ImageView) findViewById(R.id.iv_cancion_efp);
```

// implementamos el metodo OnClik a cada Botón

```
    imgBtn_adelante_efp.setOnClickListener(this);  
    imgBtn_anterior_efp.setOnClickListener(this);  
    imgBtn_atras_efp.setOnClickListener(this);  
    imgBtn_siguiente_efp.setOnClickListener(this);
```

```

imgBtn_PlayPause_efp.setOnClickListener(this);
btn_PlayList_efp.setOnClickListener(this);

/* HILO PARA SEEKBAR PROGRESO
//Creamos hilo para que nos muestre el progreso de la seekbar al mismo
tiempo que reproduce la canción
*/
actualizarProgresoSeekBar = new Thread() {
    @Override
    public void run() {
        int duracion = mp_efp.getDuration();// para obtener la durecion de la
canción
        sB_progresoCancion_efp.setMax(duracion);// pasamos la duración de la
cancion a la seekbar
        int posicionActual = 0; // la inicializamos a cero para que la barra de
progreso empiece desde cero
        int ejecucion = 0;
        while (posicionActual < duracion) { //Para calcular el progreso de nuestra
seekbar
            try {
                sleep(500);
                posicionActual = mp_efp.getCurrentPosition();//Obtiene la posición
de reproducción actual en milisegundos
                sB_progresoCancion_efp.setProgress(posicionActual);//Establece
el progreso actual en el valor especificado.
                ejecucion = sB_progresoCancion_efp.getProgress();//Obtén el nivel
actual de progreso de la barra de progreso.
                aux = conversorUnidadesTiempo(ejecucion);

                //para obtener el progreso de nuestra cancion comenzando desde
0
                tv_tiempoQLleva_efp.setText(aux.toString().trim());
                //trim()= Método que elimina los caracteres blancos iniciales y
finales de la cadena, devolviendo una copia de la misma.

            } catch (InterruptedException e) {
                tv_tiempoQLleva_efp.setText(aux); // Para mostrar un texto vacio
            }
        }
    }
};

if (mp_efp != null) { // verificamos que la mediaplayer no esta vacia
    mp_efp.stop();
} // si no esta vacia la paramos

try {
    //Recogemos los datos de la ACTivity Principal

```

```

Intent i = getIntent(); // llamamos a al intent que hicimos en el Main activity
Bundle b = i.getExtras(); // para capturar los extras que enviamos
canciones_efp = (ArrayList) b.getParcelableArrayList("canciones"); //
recogemos el valor del ArrayList enviada
posicion = (int) b.getInt("pos", 0); // recogemos la posicion enviada

//Creamos el MediaPlayer e iniciamos la reproduccion de la cancion
uri_efp = Uri.parse(canciones_efp.get(posicion).toString()); // para conseguir
la cancion que queremos reproducir
mp_efp = MediaPlayer.create(getApplication(), uri_efp); // le indicamos la
cancion a reproducir

//Rellenamos los datos a mostrar en la vista

tv_titCancion_efp.setText(canciones_efp.get(posicion).getName().toString()
    .replace(".mp3", "")); // mostramos el nombre de la cancion en la vista

tv_duracion_efp.setText(convertidorUnidadesTiempo(mp_efp.getDuration())); //
mostramos el tiempo que dura la cancion

//
actualizarProgresoSeekBar.start(); // para que comience a correr la seekbar
mp_efp.start(); // para que comience la cancion
Volumen(); // Activamos el control del volumen
}catch (Exception e){}

/*METODO SETONSEEKBARCHANGEListener DE LA SEEKBAR DE
PROGRESO
Es para que cuando se mueva la seekbar, tambien se mueva el progreso de
la cancion
*/
sB_progresoCancion_efp.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SearchBar seekBar, int i, boolean b) { // cuando
la barra de progreso se mueve
        mp_efp.seekTo(seekBar.getProgress()); // Busca la posición de tiempo
especificada en la seekbar en el mp_efp
    }

    @Override
    public void onStartTrackingTouch(SearchBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SearchBar seekBar) {

```

```

    }
    });
}

/*
METODO ONCLICK DE TODOS LO BOTONES
*/
@Override
public void onClick(View view) {
    int id = view.getId();
    switch(id){
        case R.id.imgBtn_PlayPause_efp: // cuando pulseamos el botón PlayPause
            if(mp_efp.isPlaying()) { // si la cancion se esta reproduciendo
                imgBtn_PlayPause_efp.setImageResource(R.drawable.play); // boton
con imagen pause
                iv_cancion_efp.setImageResource(R.drawable.pauseimg); // imagen
                Toast.makeText(this,"Pause", Toast.LENGTH_SHORT).show();
                mp_efp.pause(); //pausar
            }else{//si no se esta reproduciendo
                imgBtn_PlayPause_efp.setImageResource(R.drawable.pause); // boton
con imagen play
                iv_cancion_efp.setImageResource(R.drawable.ecualizador); //imagen
                Toast.makeText(this,"Play", Toast.LENGTH_SHORT).show();
                mp_efp.start(); //reproducir
            }
            break;
        case R.id.imgBtn_adelante_efp:
            mp_efp.seekTo(mp_efp.getCurrentPosition() + 5000); // sumamos a la
posicion de la seekbar 5 segundos
            break;
        case R.id.imgBtn_atras_efp:
            mp_efp.seekTo(mp_efp.getCurrentPosition() - 5000); // le restamos 5
segundos a la posicion de la seekbar
            break;
        case R.id.imgBtn_siguiente_efp:
            SiguienteCacion();
            break;
        case R.id.imgBtn_anterior_efp:
            AnteriorCacion();
            break;
        case R.id.btn_PlayList_efp:
            Toast.makeText(this,"Volvemos a la PlayList",
Toast.LENGTH_SHORT).show();
            startActivity(new
Intent(getApplicationContext(),MainActivity.class).putExtra("pos",posicion).putExtra("canciones",canciones_efp));
            break;
        default:
    }
}

```

```

        throw new IllegalStateException("Unexpected value: " + id);
    }
}
/*
METODO SIGUIENTE CANCION para pasar a la siguiente canción del ArrayList
cancions_efp
*/
public void SiguienteCacion(){
    mp_efp.stop(); // Paramos la pista actual
    posicion=(posicion+1) % canciones_efp.size(); // sumamos una a la posicion
    actual

    tv_titCancion_efp.setText(canciones_efp.get(posicion).getName().toString().replac
e(".mp3", "")); //recogemos el nombre de la cancion
    tv_duracion_efp.setText(conversorUnidadesTiempo(mp_efp.getDuration()));

    uri_efp=Uri.parse(canciones_efp.get(posicion).toString()); //recogemos la Uri
    de la cancion
    mp_efp=MediaPlayer.create(getApplicationContext(),uri_efp); //creamo el
    Media Palaer con la nueva uri

    Toast.makeText(this,"Siguiente canción", Toast.LENGTH_SHORT).show();

    mp_efp.start(); //Conmenzamos la reproduccion de la cancion
    sB_progresoCancion_efp.setMax(0); // le indicamos el maximo soportado de
    la seek bar para que empiece de cero

    try{
        sB_progresoCancion_efp.setMax(mp_efp.getDuration()); //le indicamos el
        maximo soportado por la seekbar
    }catch (Exception e){}

}
/*
METODO ANTERIOR CANCION para ir a la canción anterior en el ArrayList
canciones_efp
*/
public void AnteriorCacion(){
    mp_efp.stop(); // Paramos la pista actual

    if (posicion-1<0) { // si la posicion es menor que cero ya estamos en la ultima
    posicion
        posicion = canciones_efp.size() - 1; // por lo que volvemos a la ultima de la
    lista
    }else{//si no es menos que cero
        posicion=posicion-2; // restamos una a la posicion
    }
}

```

```

tv_titCancion_efp.setText(canciones_efp.get(posicion).getName().toString().replace(
e(".mp3", ""));//recogemos el nombre de la cancion
tv_duracion_efp.setText(conversorUnidadesTiempo(mp_efp.getDuration()));

uri_efp=Uri.parse(canciones_efp.get(posicion).toString());//recogemos la Uri
de la cancion
mp_efp=MediaPlayer.create(getApplicationContext(),uri_efp);//creamo el
Media Palaer con la nueva uri

Toast.makeText(this,"Anterior canción", Toast.LENGTH_SHORT).show();

mp_efp.start();//Conmenzamos la reproduccion de la cancion
sB_progresoCancion_efp.setMax(0);// le indicamos el maximo soportado de
la seek bar para que empiece de cero

try{
    sB_progresoCancion_efp.setMax(mp_efp.getDuration());//le indicamos el
maximo soportado por la seekbar
}catch (Exception e){}

}

/*METODO CONVERSOR UNIDADES TIEMPO
//para calcular el tiempo que dura nuestra cancion, ya que el metadata nos la
pasa el dato en milisegundo
//retorna un String
*/
private String conversorUnidadesTiempo(int milliseconds) {
    int seconds = (int) (milliseconds/1000) % 60;
    int minutes = (int) (milliseconds/ (1000*60) %60);
    int hours = (int) ((milliseconds / (1000*60*60)) % 24);
    String aux="";
    aux=((hours<10)?"0"+hours:hours)+ ":" +
((minutes<10)?"0"+minutes:minutes)+ ":"
    + ((seconds<10)?"0"+seconds:seconds);
    return aux;
}
/*
METODO VOLUMEN para controlar el volumen desde nuestra aplicacion
*/
public void Volumen(){
    try{
        sB_volumen_efp = (SeekBar)findViewById(R.id.sB_volumen_efp);//
capturamos la seekbar creada para volumen
        AudioManager_efp=(AudioManager)
getSystemService(Context.AUDIO_SERVICE);// audiomanager es la variable que
nos conecta con los servicios de audio de nuestro telefono

```

```
sB_volumen_efp.setMax(audioManager_efp.getStreamMaxVolume(AudioManager
.STREAM_MUSIC));//el volumen max que soporta nuestro telefono
```

```
sB_volumen_efp.setProgress(audioManager_efp.getStreamVolume(AudioManager
.STREAM_MUSIC));// recoegemos el volumen que tiene nuestro telefono
actualmente
```

```
        sB_volumen_efp.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int i, boolean b) {//
cuando la barra de progreso se mueve
```

```
audioManager_efp.setStreamVolume(AudioManager.STREAM_MUSIC,i,0); //
        }
    }
```

```
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {

        }
    }
```

```
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {

        }
    }
```

```
    });
} catch (Exception e){
    e.printStackTrace();
}
}
}
```

XML

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.efp.reproducto30">
    <!--AÑADIMOS PERMISOS-->
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Reproducto30">
        <!--Incluimos la segunda activity-->
        <activity
            android:name=".ReproductorActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:background="#09AEBD"
tools:context=".MainActivity">

    <!--Delegable para la playList-->
    <ListView
        android:id="@+id/lv_PlayList"
        android:layout_width="match_parent"
        android:layout_height="532dp"
        android:layout_marginTop="84dp"
        android:layout_marginBottom="24dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!--Titulo de la PlayList-->
    <TextView
        android:id="@+id/tv_titPlayList_efp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/aclonica"
        android:gravity="center"
        android:text="@string/playlist_de_elsa"
        android:textColor="#FFE57F"
        android:textSize="34sp"
        app:layout_constraintBottom_toTopOf="@+id/lv_PlayList"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

elementolista.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/id_imagen"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:orientation="horizontal">

    <!--Layout que contendra los elementos de cada una de las lineas de nuestra
    Playlist-->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <!--Imagen-->
        <ImageView
            android:id="@+id/iv_imagen_efp"
            android:layout_width="60dp"
            android:layout_height="60dp"
            android:src="@drawable/lv1"
            android:layout_marginTop="10dp"
            android:layout_marginBottom="10dp"
            android:layout_marginRight="10dp"/>
        <!--Titulo de la canción-->
        <TextView
            android:id="@+id/tv_titCancion_efp"
            android:layout_width="282dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp"
            android:fontFamily="@font/amita"
            android:text="Titulo Canción"
            android:textAlignment="inherit"
            android:textColor="@color/white"
            android:textSize="20dp"
            android:textStyle="bold"
            android:typeface="normal" />
    </LinearLayout>

</LinearLayout>
```

activity_reproductor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:background="#000000">

    <!--Layout que contendra los elementos volumen icono + seekbar-->
    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="401dp"
        android:layout_height="53dp"
        android:layout_marginStart="3dp"
        android:layout_marginEnd="7dp"
        android:layout_marginBottom="8dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toTopOf="@+id/tv_titCancion_efp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.495"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <!--icono Volumen-->
        <ImageView
            android:id="@+id/iv_cancion_efp_volumen_efp"
            android:layout_width="110dp"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:background="#000000"
            app:srcCompat="@drawable/volumenok"
            tools:ignore="SpeakableTextPresentCheck" />

        <!--control Volumen-->
        <SeekBar
            android:id="@+id/sB_volumen_efp"
            android:layout_width="377dp"
            android:layout_height="match_parent"

            android:layout_weight="1"
            app:layout_constraintTop_toTopOf="parent" />
```

```
</LinearLayout>
```

```
<!--Título de la canción-->
```

```
<TextView
```

```
    android:id="@+id/tv_titCancion_efp"  
    android:layout_width="403dp"  
    android:layout_height="56dp"  
    android:layout_marginTop="8dp"  
    android:layout_marginBottom="16dp"  
    android:fontFamily="@font/architects_daughter"  
    android:gravity="center"  
    android:text="TextView"  
    android:textColor="#FFFFFFF"  
    android:textSize="24sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toTopOf="@+id/iv_cancion_efp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
```

```
<!--Imagen canción-->
```

```
<ImageView
```

```
    android:id="@+id/iv_cancion_efp"  
    android:layout_width="180dp"  
    android:layout_height="180dp"  
    android:layout_marginTop="28dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.497"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/tv_titCancion_efp"  
    app:srcCompat="@drawable/ecualizador" />
```

```
<!--Botón para volver a al playList(MainActivity)-->
```

```
<ImageButton
```

```
    android:id="@+id/btn_PlayList_efp"  
    android:layout_width="60dp"  
    android:layout_height="60dp"  
    android:layout_marginStart="61dp"  
    android:layout_marginTop="208dp"  
    android:layout_marginEnd="28dp"  
    android:layout_marginBottom="8dp"  
    android:background="#000000"  
    app:layout_constraintBottom_toTopOf="@+id/sB_progresoCancion_efp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="1.0"  
    app:layout_constraintStart_toEndOf="@+id/iv_cancion_efp"  
    app:layout_constraintTop_toBottomOf="@+id/tv_titCancion_efp"
```

```

        app:srcCompat="@drawable/playlist" />

<!--control progreso canción-->
<SeekBar
    android:id="@+id/sB_progresoCancion_efp"
    style="@style/Widget.AppCompat.SeekBar"
    android:layout_width="370dp"
    android:layout_height="31dp"
    android:layout_marginStart="13dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="28dp"
    android:layout_marginBottom="24dp"
    android:progressTint="#1E94C8"
    android:thumb="@android:drawable/presence_online"
    app:layout_constraintBottom_toTopOf="@+id/tv_tiempoQLleva_efp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_PlayList_efp" />

<!--Muestra el tiempo que lleva la reproduccion de la cancion-->
<TextView
    android:id="@+id/tv_tiempoQLleva_efp"
    android:layout_width="113dp"
    android:layout_height="31dp"
    android:layout_marginStart="26dp"
    android:layout_marginTop="12dp"
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="23dp"
    android:text="TextView"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
    app:layout_constraintEnd_toStartOf="@+id/tv_duracion_efp"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/sB_progresoCancion_efp" />

<!--Muestra la duracion de la cancion -->
<TextView
    android:id="@+id/tv_duracion_efp"
    android:layout_width="95dp"
    android:layout_height="30dp"
    android:layout_marginStart="149dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="28dp"
    android:layout_marginBottom="37dp"
    android:text="TextView"
    android:textAlignment="viewEnd"
    android:textColor="#FFFFFF"

```

```

app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/tv_tiempoQLleva_efp"
app:layout_constraintTop_toBottomOf="@+id/sB_progresoCancion_efp" />

<!--layout que contiene los controles de
PLAY+PAUSE+ATRAS+ANTERIOR+ADELANTE+SIGUIENTE-->
<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="416dp"

    android:layout_height="75dp"
    android:layout_marginTop="12dp"
    android:layout_marginBottom="141dp"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv_tiempoQLleva_efp">

    <!--Boton cancion anterior-->
    <ImageButton
        android:id="@+id/imgBtn_anterior_efp"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_weight="1"
        android:background="#000000"
        app:srcCompat="@drawable/anterior"
        tools:ignore="SpeakableTextPresentCheck" />

    <!--Boton atras -->
    <ImageButton
        android:id="@+id/imgBtn_atras_efp"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_weight="1"
        android:backgroundTint="#000000"
        app:srcCompat="@drawable/atras" />

    <!--Boton Play+Pause-->
    <ImageButton
        android:id="@+id/imgBtn_PlayPause_efp"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_weight="1"
        android:background="#000000"
        app:srcCompat="@drawable/pause"

```

```

        tools:ignore="SpeakableTextPresentCheck" />

<!--Boton adelante-->
<ImageButton
    android:id="@+id/imgBtn_adelante_efp"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_weight="1"
    android:background="#000000"
    app:srcCompat="@drawable/adelante"
    tools:ignore="SpeakableTextPresentCheck" />

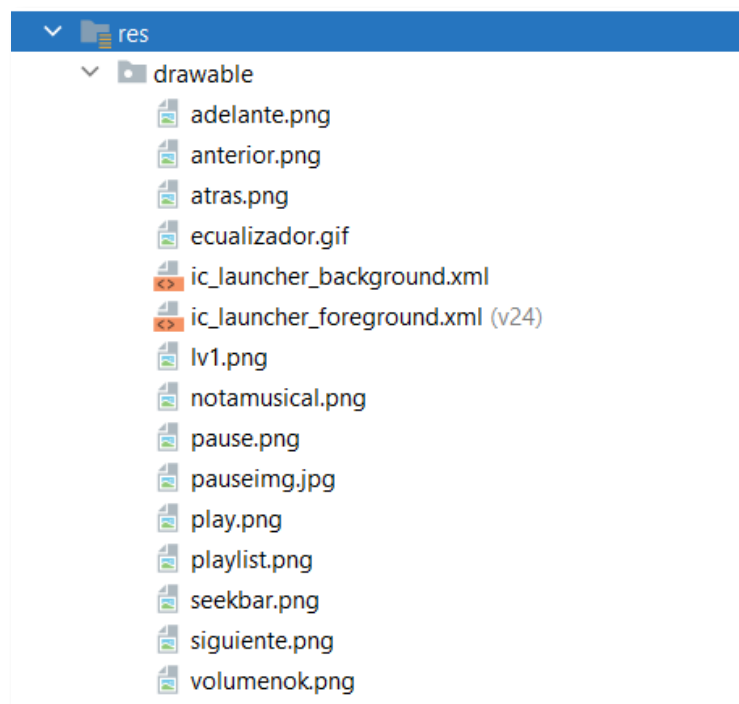
<!--Boton cancion siguiente-->
<ImageButton
    android:id="@+id/imgBtn_siguiente_efp"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_weight="1"
    android:background="#000000"
    app:srcCompat="@drawable/siguiente"
    tools:ignore="SpeakableTextPresentCheck" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

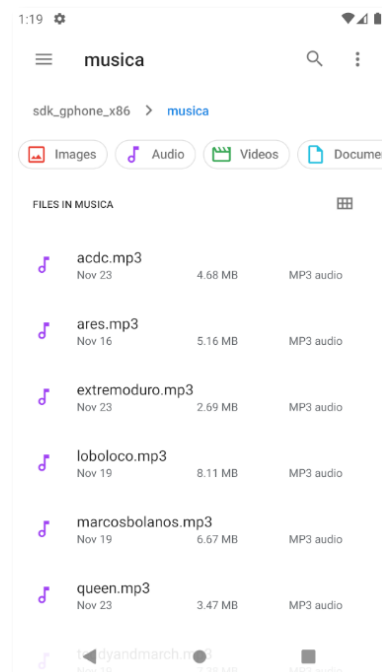
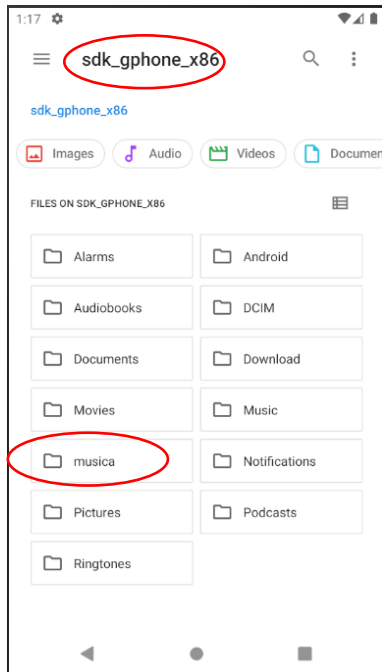
Otros elementos importantes:



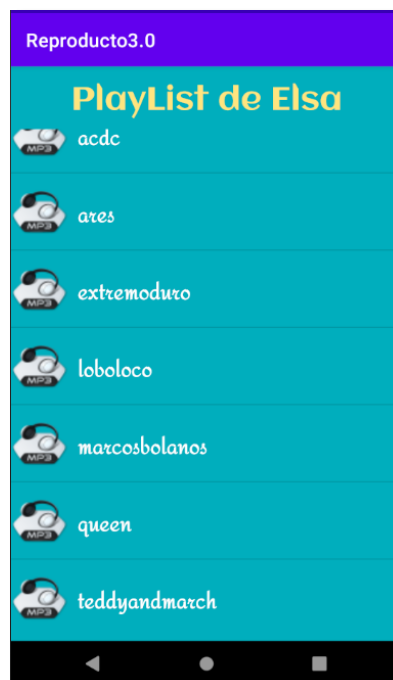
CAPTURAS DE PANTALLA

EMULADOR USADO PARA LAS CAPTURAS: *Nexus 5X API 30*

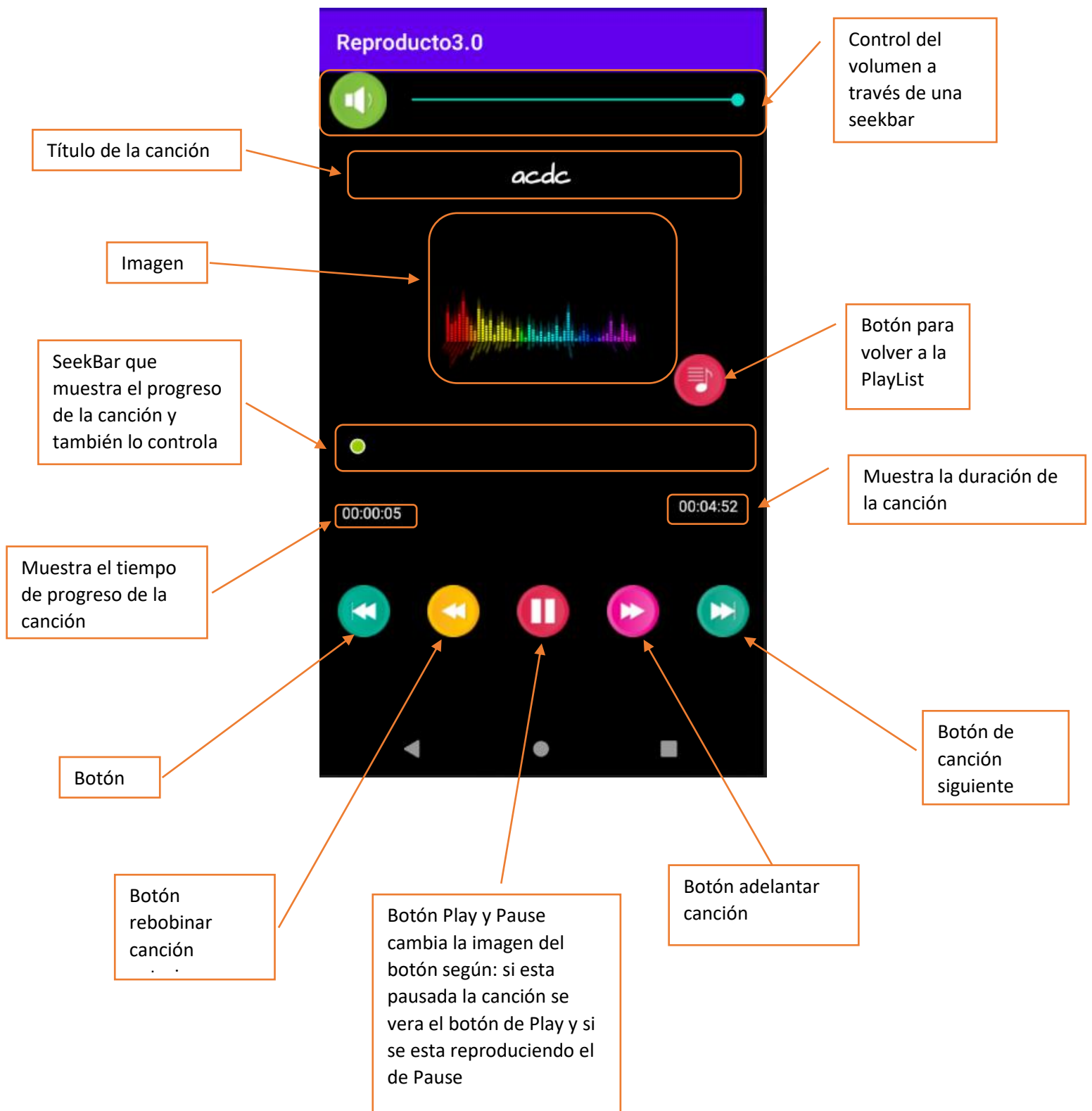
- Introducimos las canciones en el emulador siguiendo los siguientes pasos dentro de *File / sdk_gphone_x86 / música*. Esta última carpeta creada por mí.



- Comprobamos que las listview que hemos adaptado de manera personalizada muestra las canciones introducidas en la carpeta música del emulador.

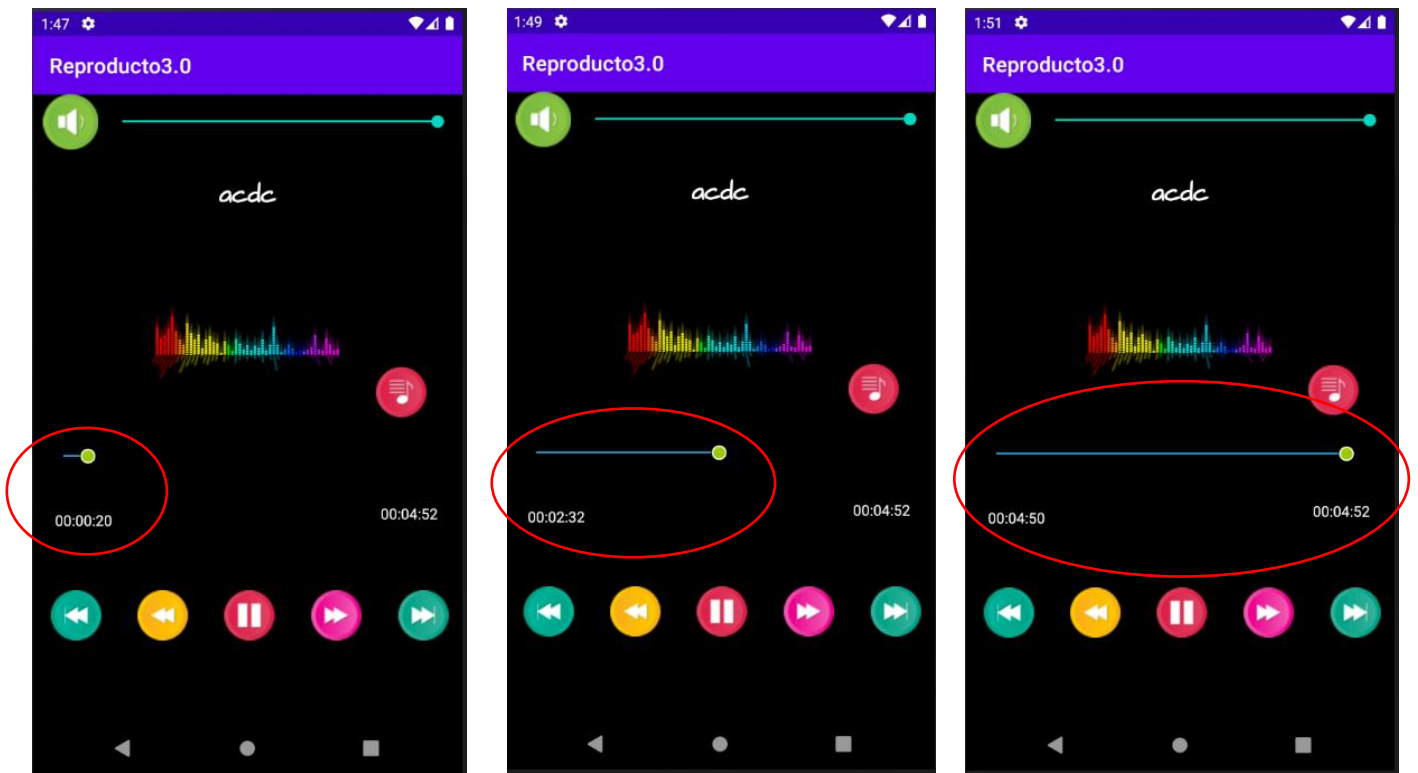


- Al pulsar cualquiera de las canciones que muestra la PlayList nos abre una segunda activity, en la que encontramos los siguientes elementos

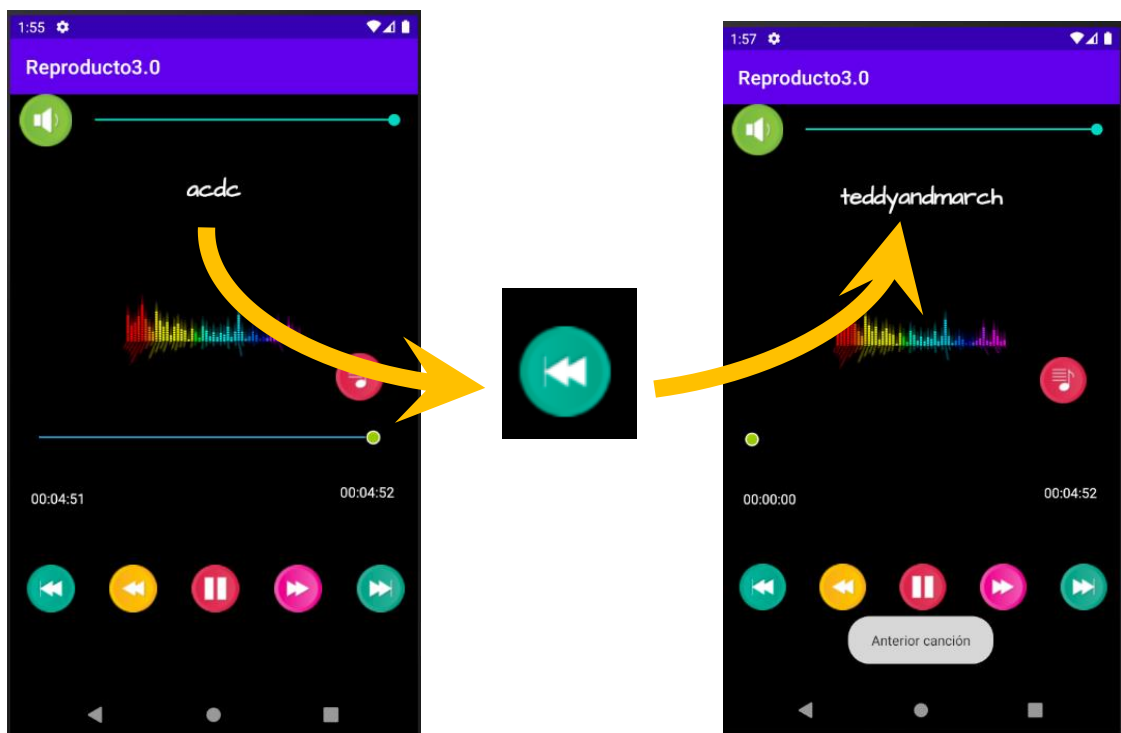


COMPROBACIONES DE LOS BOTONES Y CONTROLES:

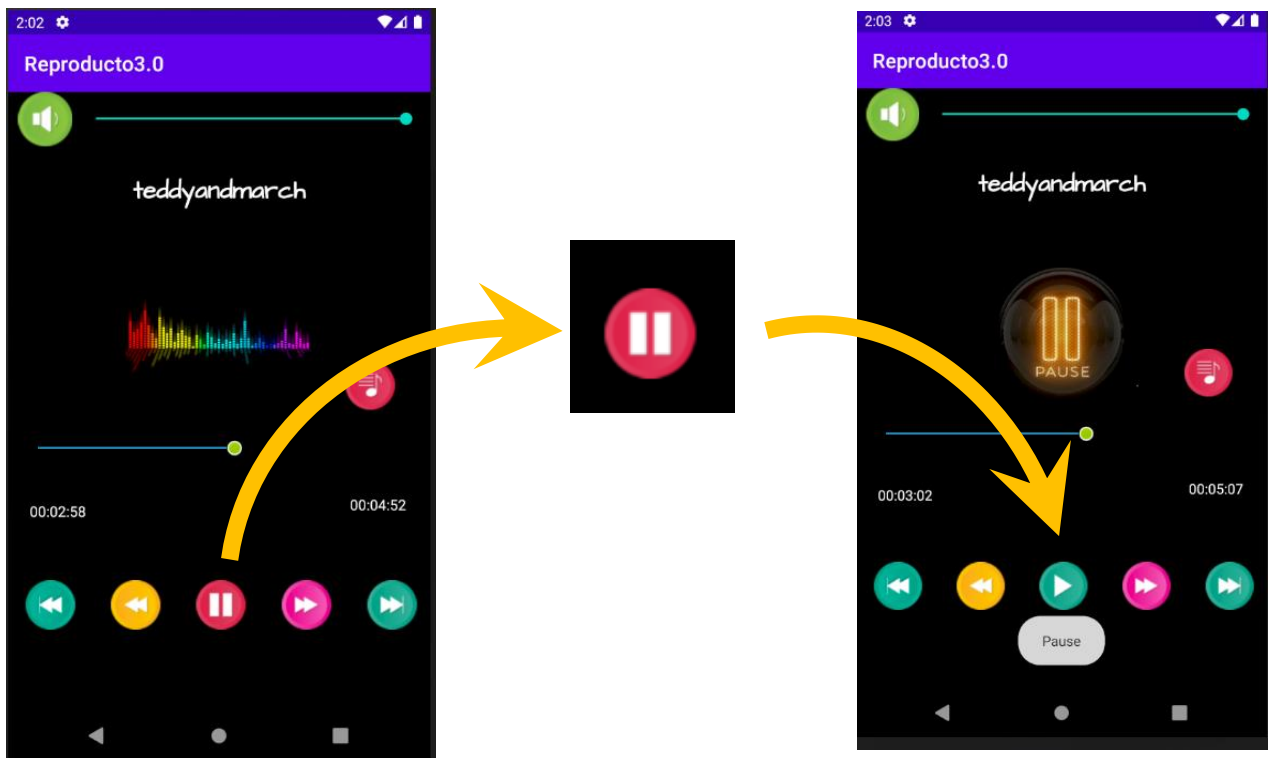
- seekbar de progreso: comprobamos como se mueve según el tiempo de la canción.



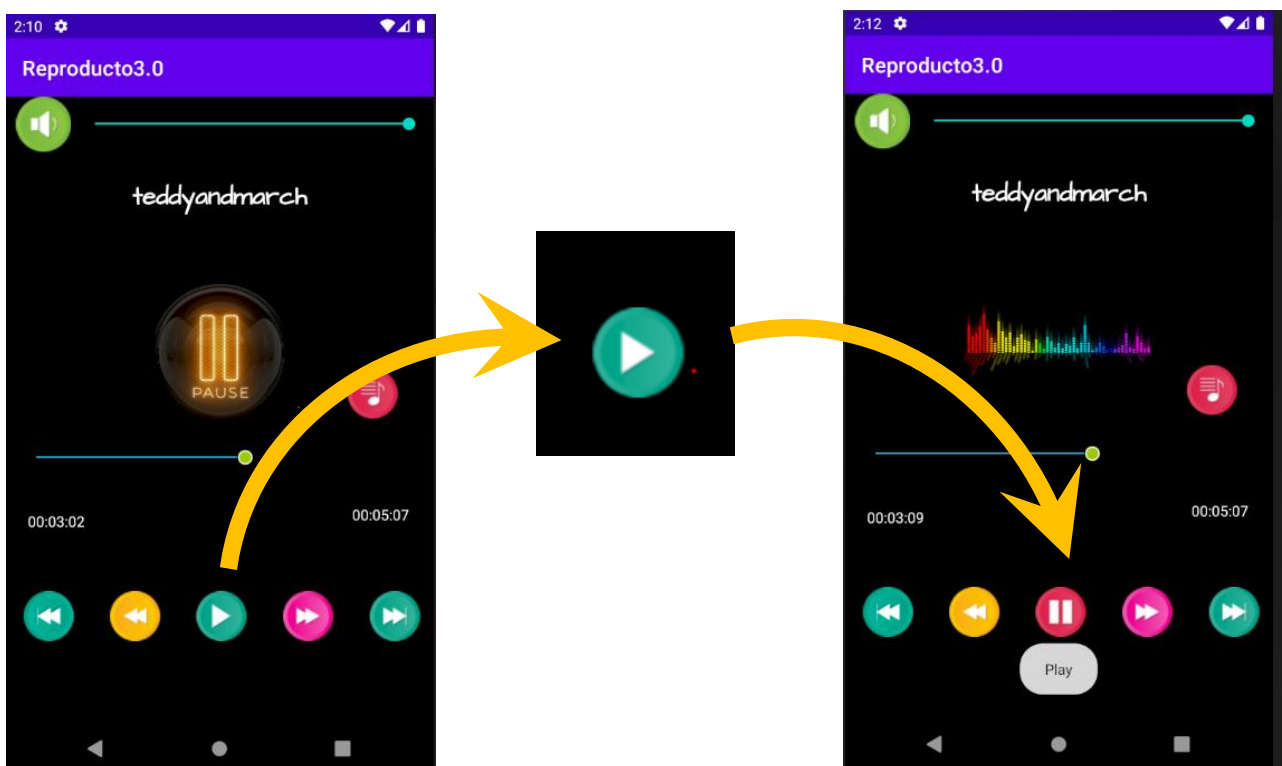
- Comprobamos el botón de canción anterior: en este caso estamos reproduciendo "acdc", al ser la primera de la lista al dar al botón pasara a la última de la lista que es "teddyandmarch".



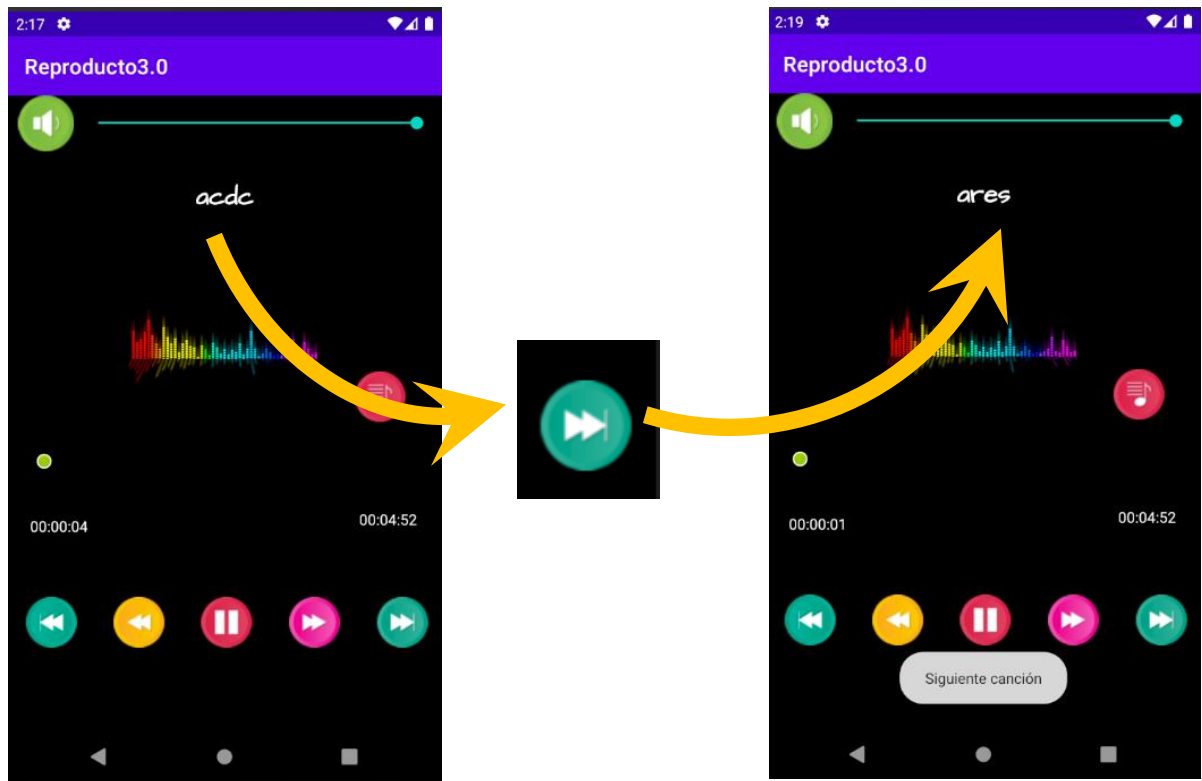
- Botón Play y Pause:
Comprobamos que, al estar reproduciendo una canción, si pulsamos pause nos cambia la imagen principal y también el botón cambia a Play



Comprobamos que al estar pausada si le da al Play vuelve a cambiar la imagen principal y el botón cambia de nuevo a Pause



- Botón siguiente canción:
Reproducimos la primera canción de la PlayList que es “acdc” y al pulsar el botón nos pasa a la canción “ares” que es la segunda en nuestra PlayList.



- Botón volver a la PlayList.

