



Universidad Latina de Panamá

Facultad de Ingeniería en Sistemas

Lic. De ingeniería en Sistemas informáticos

Materia: Programación VI

Nombre del Profesor: Oriel Cedeño

Laboratorio 7

Integrantes:

Brian Rodríguez 8-949-684

Alexander Zhuo 8-965-959

Año
2024

```

package com.example.multimedia;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    4 usages
    Button play_pause, btn_repetir;
    no usages
    MediaPlayer mp;
    18 usages
    ImageView iv;
    3 usages
    int repetir = 2, posicion = 0;

    26 usages
    MediaPlayer vectormp[] = new MediaPlayer[4]; // cantidad de canciones que se van a reproducir

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    play_pause = findViewById(R.id.button4);
    btn_repetir = findViewById(R.id.button2);
    iv = findViewById(R.id.imageView);

    //posiciones de las canciones
    vectormp[0] = MediaPlayer.create(context: this, R.raw.mario);
    vectormp[1] = MediaPlayer.create(context: this, R.raw.avengers);
    vectormp[2] = MediaPlayer.create(context: this, R.raw.marvel);
    vectormp[3] = MediaPlayer.create(context: this, R.raw.champions);
}

```

```

//Metodo para el boton PlayPause
1 usage
public void PlayPause(View v){
    if(vectormp[posicion].isPlaying()){
        vectormp[posicion].pause();
        play_pause.setBackgroundResource(R.drawable.reproducir);
        Toast.makeText( context: this, text: "Pausa",Toast.LENGTH_SHORT).show();
    }else {
        vectormp[posicion].start();
        play_pause.setBackgroundResource(R.drawable.pausar);
        Toast.makeText( context: this, text: "Play",Toast.LENGTH_SHORT).show();
    }
}
}

```

```

//Metodo para el boton Stop
1 usage
public void Stop(View view){
    if(vectormp[posicion] != null){
        vectormp[posicion].stop();
        vectormp[0] = MediaPlayer.create( context: this,R.raw.mario);
        vectormp[1] = MediaPlayer.create( context: this,R.raw.avengers);
        vectormp[2] = MediaPlayer.create( context: this,R.raw.marvel);
        vectormp[3] = MediaPlayer.create( context: this,R.raw.champions);

        //regresa a la posicion 0
        posicion = 0;
        play_pause.setBackgroundResource(R.drawable.reproducir);
        iv.setImageResource(R.drawable.mario_bross);
        Toast.makeText( context: this, text: "Stop",Toast.LENGTH_SHORT).show();
    }
}
}

```

```

//Metodo repetir una pista
1 usage
public void Repetir(View v){
    if(repetir ==1){
        btn_repetir.setBackgroundResource(R.drawable.no_repetir);
        Toast.makeText( context: this, text: "No repetir",Toast.LENGTH_SHORT).show();
        vectormp[posicion].setLooping(false);
        repetir = 3;
    }else{
        btn_repetir.setBackgroundResource(R.drawable.repetir);
        Toast.makeText( context: this, text: "Repetir",Toast.LENGTH_SHORT).show();
        vectormp[posicion].setLooping(false);
        repetir = 1;
    }
}
}

```

```

//Metodo para saltar a la siguiente cancion
1 usage
public void Siguiente(View v){
    if(posicion< vectormp.length -1){
        if(vectormp[posicion].isPlaying()){
            vectormp[posicion].stop();
            posicion++;
            vectormp[posicion].start();

            if (posicion == 0) {
                iv.setImageResource(R.drawable.mario_bross);
            } else if(posicion==1){
                iv.setImageResource(R.drawable.avengerss);
            }else if(posicion==2){
                iv.setImageResource(R.drawable.marvell);
            }else if(posicion==3){
                iv.setImageResource(R.drawable.championss);
            }
        }else{
            posicion++;
        }
    }
}

```

```

        if (posicion == 0) {
            iv.setImageResource(R.drawable.mario_bross);
        } else if(posicion==1){
            iv.setImageResource(R.drawable.avengerss);
        }else if(posicion==2){
            iv.setImageResource(R.drawable.marvell);
        }else if(posicion==3){
            iv.setImageResource(R.drawable.championss);
        }
    }
} else{
    Toast.makeText( context: this, text: "No hay mas canciones",Toast.LENGTH_SHORT).show();
}
}

```

```

//Metodo para regresar a la cancion anterior
1 usage
public void Anterior(View v){
    if(posicion >=1){
        if(vectormp[posicion].isPlaying()){
            vectormp[posicion].stop();
            vectormp[0] = MediaPlayer.create( context: this,R.raw.mario);
            vectormp[1] = MediaPlayer.create( context: this,R.raw.avengers);
            vectormp[2] = MediaPlayer.create( context: this,R.raw.marvel);
            vectormp[3] = MediaPlayer.create( context: this,R.raw.champions);
            posicion--;

            if (posicion == 0) {
                iv.setImageResource(R.drawable.mario_bross);
            } else if(posicion==1){
                iv.setImageResource(R.drawable.avengerss);
            }else if(posicion==2){
                iv.setImageResource(R.drawable.marvell);
            }else if(posicion==3){
                iv.setImageResource(R.drawable.championss);
            }

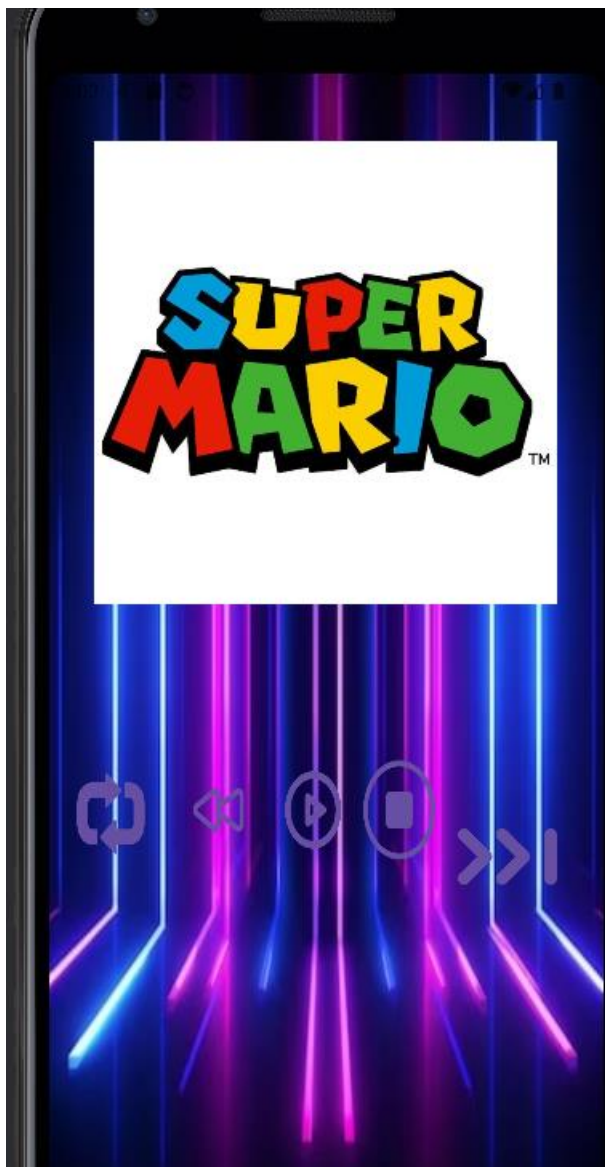
            vectormp[posicion].start();
        }else{
            posicion--;
        }
    }
}

```

```

        if (posicion == 0) {
            iv.setImageResource(R.drawable.mario_bross);
        } else if(posicion==1){
            iv.setImageResource(R.drawable.avengerss);
        }else if(posicion==2){
            iv.setImageResource(R.drawable.marvell);
        }else if(posicion==3){
            iv.setImageResource(R.drawable.championss);
        }
    }
} else{
    Toast.makeText( context: this, text: "No hay mas canciones",Toast.LENGTH_SHORT).show();
}
}
}

```

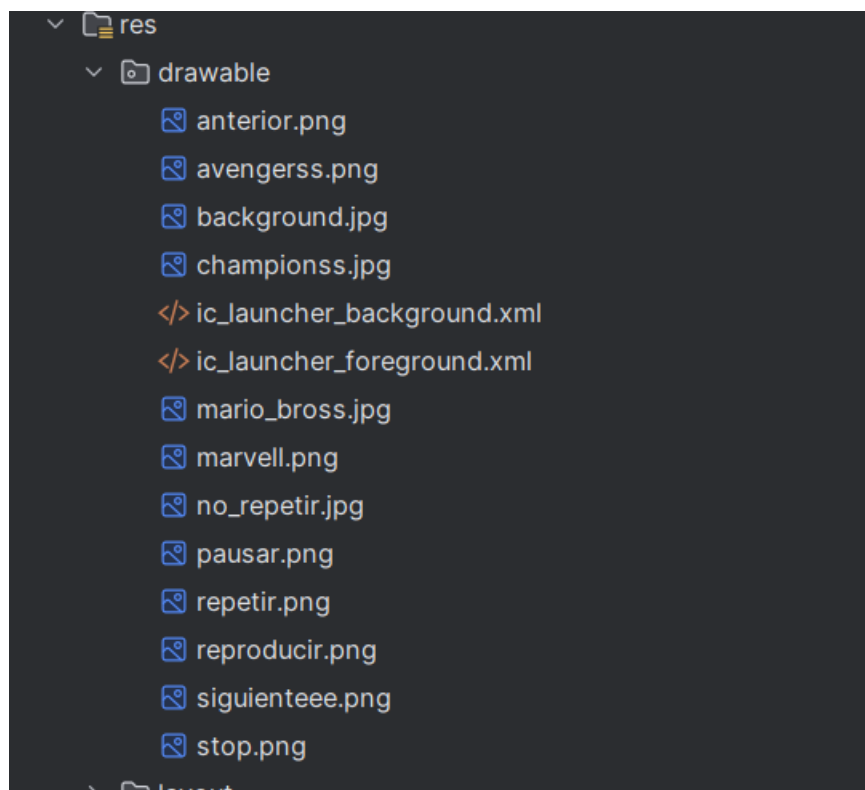


Variables de instancia

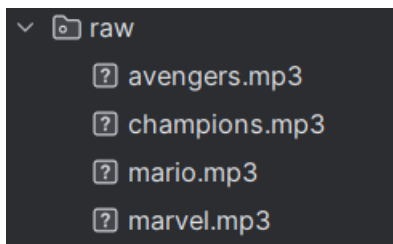
- Button play_pause: Botón para reproducir/pausar la música.
- Button btn_repetir: Botón para controlar la repetición de la pista.
- MediaPlayer mp: Objeto MediaPlayer para reproducir archivos de audio.
- ImageView iv: ImageView para mostrar la imagen asociada con la pista de música.
- int repetir: Variable para controlar el estado de repetición.
- int posicion: Variable para rastrear la posición actual de la canción que se está reproduciendo.
- MediaPlayer vectormp[]: Un array de objetos MediaPlayer para almacenar las instancias de las canciones a reproducir.

Métodos

- PlayPause(View v): Método llamado cuando se hace clic en el botón de reproducción/pausa. Alterna entre reproducir y pausar la canción actual
- Stop (View view): Método llamado cuando se hace clic en el botón de detener. Detiene la reproducción de la canción actual y restablece las canciones en el array de MediaPlayer y muestra un mensaje de notificación.
- Repetir (View v): Método llamado cuando se hace clic en el botón de repetición. Alterna entre repetir y no repetir la pista actual.
- Siguiente (View v): Método llamado cuando se hace clic en el botón de siguiente pista. Cambia a la siguiente canción en la lista de reproducción, comienza a reproducir la nueva canción si la canción anterior estaba en reproducción.
- Anterior (View v): Método llamado cuando se hace clic en el botón de pista anterior. Cambia a la canción anterior en la lista de reproducción y comienza a reproducir la nueva canción si la canción anterior estaba en reproducción.



En la carpeta res/drawable, se anexaron las imágenes necesarias para el reproductor de música.



Se creó una carpeta con el nombre de “raw”, esto nos permite realizar lo siguiente:

La aplicación maneja la carga de las canciones desde recursos de Android (R.raw.*).

Se utiliza un array de objetos MediaPlayer para manejar múltiples canciones.

Actualiza la canción que se esté reproduciendo y muestra una imagen asociada con la canción.