

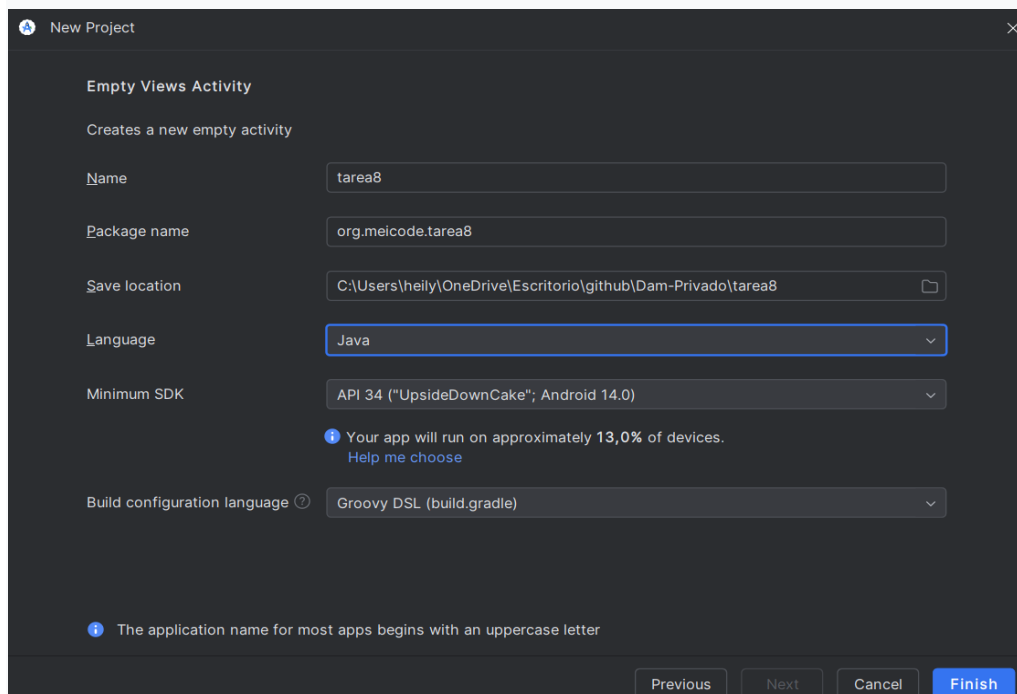
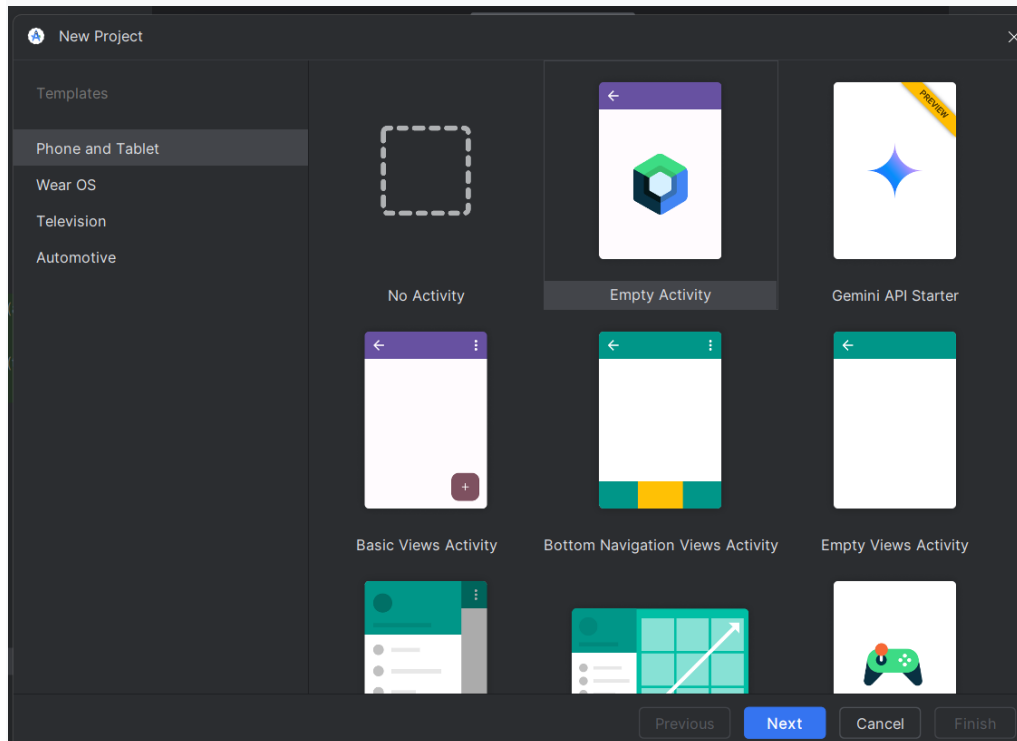
Tarea Individual 8 - Reproduccion audio y video

Objetivo

Crear una aplicación con un menú que permita al usuario reproducir un archivo de audio y un archivo de video utilizando MediaPlayer y VideoView respectivamente.

Instrucciones

1. Crea una actividad vacía.



- Añade dos botones en el diseño principal: uno con el texto "Reproducir audio" y otro con el texto "Reproducir video".

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

    <Button
        android:id="@+id/buttonActivity1"
        android:layout_width="match_parent"
        android:layout_height="68dp"
        android:layout_alignParentTop="true"
        android:layout_marginTop="16dp"
        android:layout_marginHorizontal="16dp"
        android:text="@string/buttonActivity1" />

    <Button
        android:id="@+id/buttonActivity2"
        android:layout_width="match_parent"
        android:layout_height="68dp"
        android:layout_below="@id/buttonActivity1"
        android:layout_marginTop="16dp"
        android:layout_marginHorizontal="16dp"
        android:text="@string/buttonActivity2" />

</RelativeLayout>
```

```
public class MainActivity extends AppCompatActivity {

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

        Button buttonActivity1 = findViewById(R.id.buttonActivity1);
        Button buttonActivity2 = findViewById(R.id.buttonActivity2);

        // Intent para el primer botón
        buttonActivity1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Intent para iniciar la Actividad 1
                Intent intent1 = new Intent( packageContext: MainActivity.this, Activity1.class);
                startActivity(intent1);
            }
        });

        // Intent para el segundo botón
        buttonActivity2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Intent para iniciar la Actividad 2
                Intent intent2 = new Intent( packageContext: MainActivity.this, Activity2.class);
                startActivity(intent2);
            }
        });
    }
}
```

RESULTADO:



- Configura el botón "Reproducir audio" para que reproduzca un archivo de audio utilizando MediaPlayer. El archivo de audio puede estar en la carpeta **res/raw**.

Script de JAVA:

```
public class Activity1 extends AppCompatActivity {

    private MediaPlayer mediaPlayer; // 7 usages
    private String url = "https://www.learningcontainer.com/wp-content/uploads/2020/02/Kalimba.mp3";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity1);

        Button playbut = findViewById(R.id.button_play);
        Button stopbut = findViewById(R.id.button_stop);

        initializeMediaPlayer();

        playbut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!mediaPlayer.isPlaying()) {
                    mediaPlayer.start();
                }
            }
        });

        stopbut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mediaPlayer.isPlaying()) {
                    mediaPlayer.stop();
                    initializeMediaPlayer(); // Reinicializa el MediaPlayer después de detenerlo
                }
            }
        });
    }
}
```

```

private void initializeMediaPlayer() { 2 usages
    Uri parseurl = Uri.parse(url);
    mediaPlayer = MediaPlayer.create(context: this, parseurl);
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (mediaPlayer != null) {
        mediaPlayer.release(); // Libera los recursos del Medi
    }
}
}

```

#Aquí hemos realizado una modificación para que si tú después de que has parado el audio quieres volver a reproducir el audio se siga reproduciendo, ya que en clase el código que hicimos no podía. La modificación consiste en reinicializar el objeto MediaPlayer después de detenerlo. Esto se hace utilizando el método initializeMediaPlayer dentro del listener del botón stop. Después de llamar a mediaPlayer.stop() en el listener del botón "STOP", se invoca al método initializeMediaPlayer(). Este método crea un nuevo objeto MediaPlayer configurado con la misma URL del audio. De esta forma, el objeto vuelve al estado inicial y está listo para reproducirse nuevamente. Funciona porque crea un nuevo MediaPlayer con la URL proporcionada, asegurando que este esté en el estado PREPARED. Al presionar el botón "PLAY", el nuevo objeto MediaPlayer puede reproducirse desde el inicio del audio.

Layout:

```

<?xml version="1.0" encoding="utf-8"?>
<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=".Activity1">

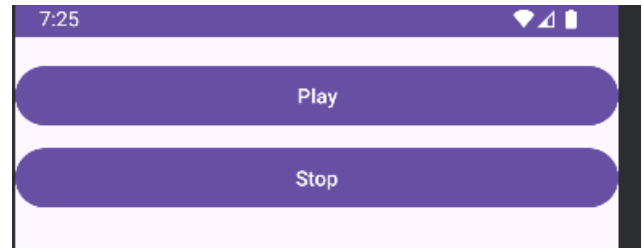
    <Button
        android:id="@+id/button_play"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Play"
        android:layout_alignParentTop="true"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/button_stop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Stop"
        android:layout_below="@id/button_play"
        android:layout_marginTop="8dp" />

</RelativeLayout>

```

RESULTADO:



4. Configura el botón "Reproducir video" para que reproduzca un archivo de video utilizando `VideoView`. El archivo de video puede estar en la carpeta `res/raw` o en un URI accesible.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <VideoView
        android:id="@+id/VideoView"
        android:layout_width="match_parent"
        android:layout_height="250dp"/>

</LinearLayout>
```

```
public class Activity2 extends AppCompatActivity {

    private MediaController mediaController; 3 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2);

        String url="android.resource://" + getPackageName() + "/" + R.raw.sample_video;

        Uri urlparse=Uri.parse(url);

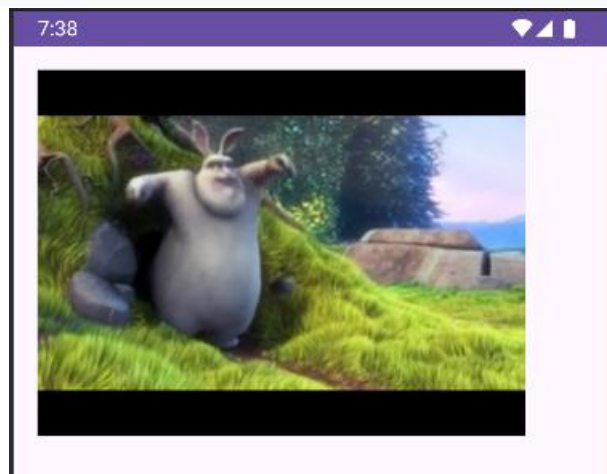
        VideoView videoView=findViewById(R.id.VideoView);

        mediaController=new MediaController( context: this);
        mediaController.setAnchorView(videoView);
        videoView.setMediaController(mediaController);

        videoView.setVideoURI(urlparse);

        videoView.start();
    }
}
```

RESULTADO:



5. Prueba la aplicación para garantizar que ambos botones funcionan correctamente y reproducen los archivos correspondientes.

He ido adjuntando imágenes a lo largo de la explicación.