

## PROGRAMACION EN ANDROID CON JAVA

### PRACTICA No. 6

### SONIDOS

En Android se tienen dos componentes para manejar sonidos SoundPool y MediaPlayer.

SoundPool puede cargar en la memoria un recurso existente dentro del desarrollo (APK) o desde un archivo en el sistema de archivos, es adecuado para clips cortos y perfecto para efectos de sonido.

MediaPlayer está diseñado para archivos de sonido más largos o transmisiones, esto es más adecuado para archivos de multimedia más grandes. Los archivos se cargarán desde el disco cada vez que se llame al método create, esto ahorrará espacio en la memoria, pero introducirá un pequeño retraso (no muy notable).

#### PARTE 1.

#### CREACION DEL PROYECTO

Ejecute el programa *Android Studio*. En la ventana que se despliega haga clic sobre el menú *File* y la Opción *New Project*.

En la nueva ventana seleccione la pestaña *Phone and Table*, elija la opción *Empty Activity* y haga clic en el botón Next.

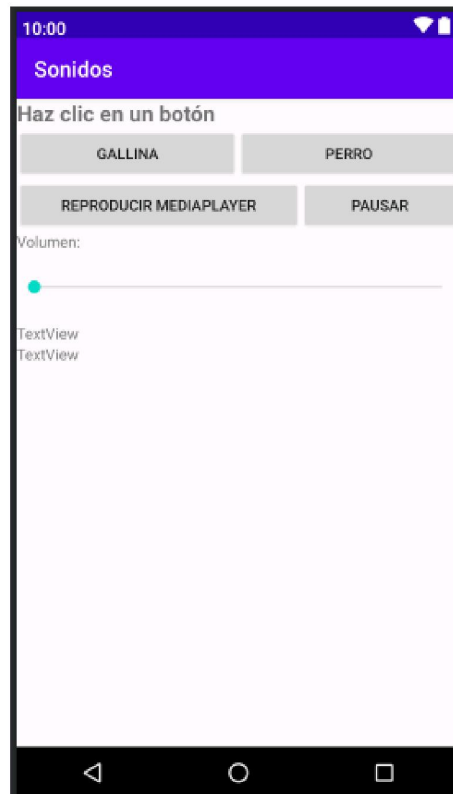
En la siguiente ventana escriba como nombre del proyecto **Sonidos** y como dominio escriba **ud.example.sonidos**, seleccione la mínima versión de android de ejecución para su nueva aplicación (se recomienda la API 23 por defecto), seleccione como lenguaje Java y haga clic sobre el botón Finish.

#### PARTE 2.

#### ADICION DE CODIGO

##### Modificación del archivo *activity\_main.xml*

Nuestra actividad al final debería lucir de la siguiente manera:



Haga doble clic sobre el archivo *activity\_main.xml* del paquete *layout*, y cambie el modo de visualización a texto, por medio de la pestaña *Text*. Modifique el código de la siguiente manera:

```
<?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"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:text="Haz clic en un botón"
        android:textSize="20dp"
        android:textStyle="bold"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="suenaboton01"
            android:text="Gallina" />

        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="suenaboton02"
            android:text="Perro" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="suenaboton03"
            android:text="Reproducir mediaPlayer" />

        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="suenaboton04"
            android:text="pausar" />
    </LinearLayout>

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Volumen:" />
```

```
<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="66dp" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView" />

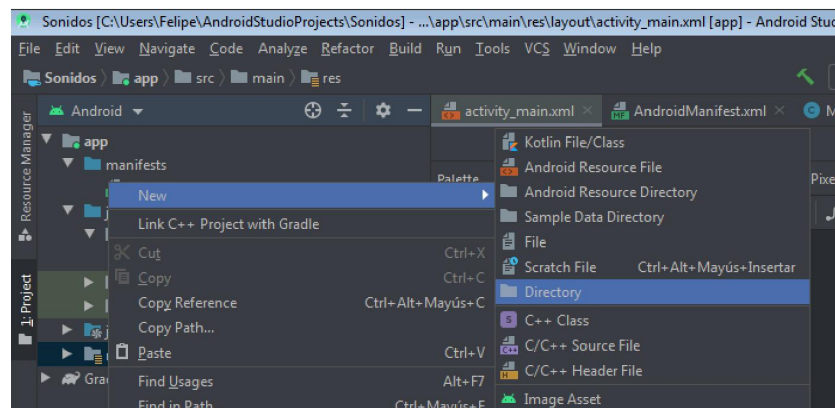
<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView" />

</LinearLayout>

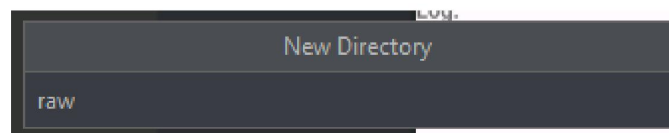
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Adición de los archivos wav al proyecto

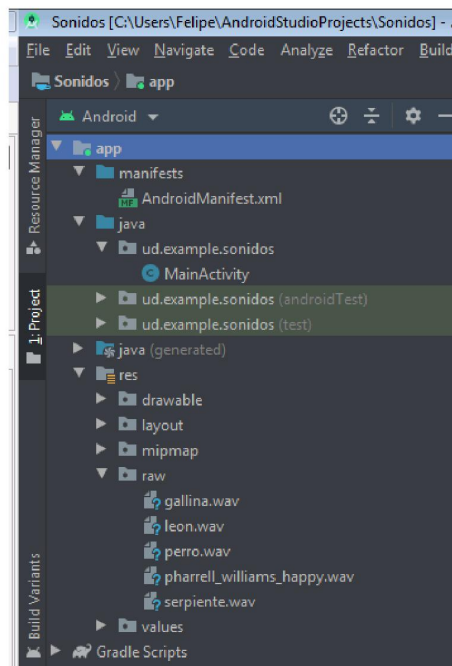
Sobre el paquete res del proyecto haga clic derecho y seleccione la opción *Directory* del submenú *New*.



Al nuevo paquete lo llamaremos raw



Por medio de un explorador de Windows o Linux buscaremos esa carpeta y copiaremos los archivos .wav adjuntos a la práctica en dicho directorio. Al realizar el anterior proceso nuestro paquete debe mostrar que contiene nuevos archivos.



### Modificación del archivo *MainActivity.java*

Haga doble clic sobre el archivo *MainActivity.java* del paquete *java*, y modifique el código de la siguiente manera:

Agregue los siguientes atributos privados a la clase

```
private SoundPool sPool;  
private MediaPlayer player;  
private AudioManager audioManager;  
private int sound1, sound2;  
private Button boton03, boton04;  
private SeekBar volumen;  
private TextView text1, text2;
```

En el método *OnCreate* de la clase agregue el siguiente código:

```
boton03 = findViewById(R.id.button3);  
boton04 = findViewById(R.id.button4);  
text1 = findViewById(R.id.textView4);  
text2 = findViewById(R.id.textView5);  
  
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
    AudioAttributes attributes = new AudioAttributes.Builder()  
        .setUsage(AudioAttributes.USAGE_ASSISTANCE_SONIFICATION)  
        .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)  
        .build();  
    sPool = new SoundPool.Builder()
```

```
        .setAudioAttributes(attributes)
        .setMaxStreams(10)
        .build();
    }else{
        sPool = new SoundPool(6, AudioManager.STREAM_MUSIC, 0);
    }
    sound1 = sPool.load(this, R.raw.gallina,1);
    sound2 = sPool.load(this, R.raw.perro,1);

    AudioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
    volumen = findViewById(R.id.seekBar);
    volumen.setMax(AudioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC));
    volumen.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
            //Log.i("Volumen", String.valueOf(i));
            AudioManager.setStreamVolume(AudioManager.STREAM_MUSIC,i,0);
        }
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
        }
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
        }
    });
});
```

Agregamos los siguientes métodos a la clase, los cuales vamos a conectar a los botones de la actividad:

```
public void suonaboton01 (View v){
    sPool.play(sound1,1,1,1,0,1);
    //Log.i("MIREMOS", "LLEGAMOS");
}
public void suonaboton02 (View v){
    sPool.play(sound2,1,1,1,1,1);
}
public void suonaboton03(View v){
    if (player != null && player.isPlaying()){
        player.stop();
        boton03.setText("Reproducir mediaplayer");
    } else {
        player = MediaPlayer.create(this,R.raw.pharrell_williams_happy);
        text1.setText(String.valueOf(player.getDuration()));
        player.start();
        boton03.setText("Detener mediaplayer");
    }
}
public void suonaboton04(View v){
    if (player != null){
        if(player.isPlaying()) {
            player.pause();
            text2.setText(String.valueOf(player.getCurrentPosition()));
            boton04.setText("REINICIAR");
        }
    }
}
```



```
    }  
    else {  
        player.start();  
        boton04.setText("PAUSAR");  
    }  
}  
}  
  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    sPool.release();  
    sPool = null;  
    player.release();  
    player = null;  
}
```

Ejecuté y Evalué la aplicación

### TRABAJO ADICIONAL

1. Adicionar dos botones que le permitan al usuario escuchar el sonido del león y la serpiente dentro del aplicativo.
2. Adicione el código necesario para que la barra de volumen inicie mostrando la posición actual del volumen del dispositivo.
3. Adicione el código y objetos necesarios que muestre una barra de progreso cuando se escucha un objeto MediaPlayer.
4. Corrija cualquier mal funcionamiento de la aplicación.