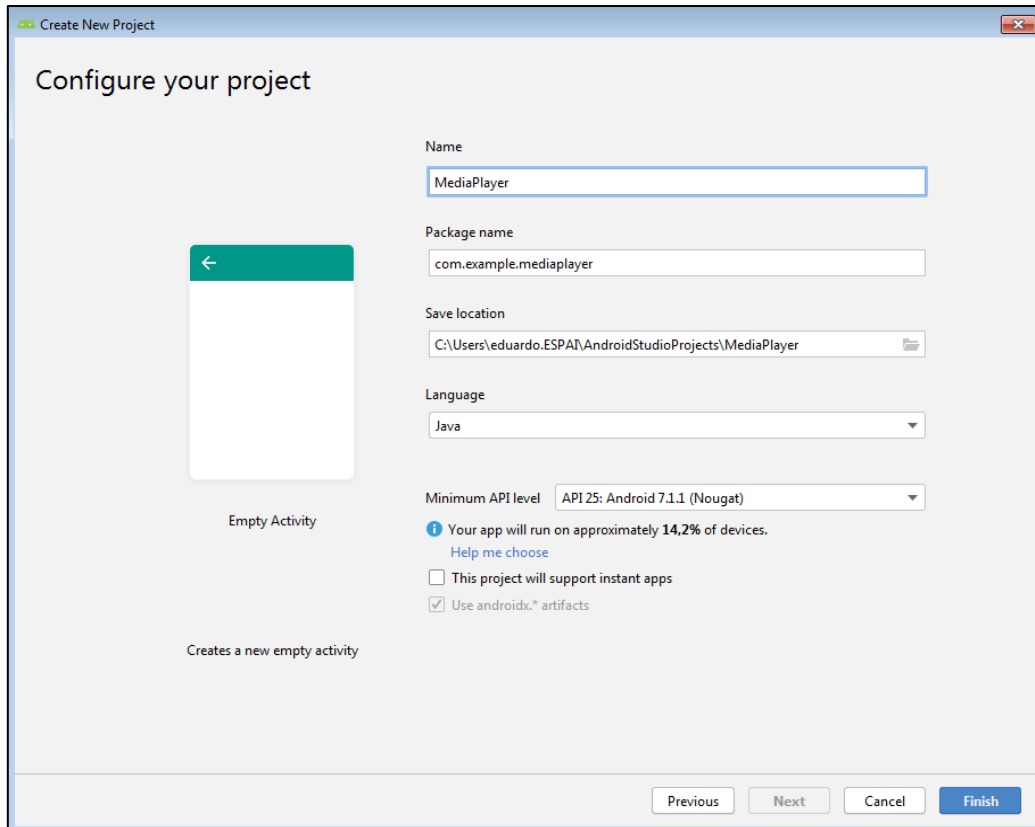


PRACTICA 11: MEDIAPLAYER

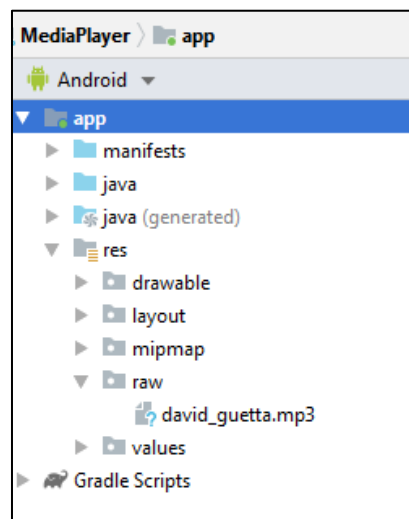
Parte I: Reproducir un vídeo con MediaPlayer

Paso 1. Crea un nuevo proyecto de nombre mediaPlayer:



Paso 2. Crea una nueva carpeta con nombre raw dentro de la carpeta res.

Paso 3. Arrastra a su interior el fichero de audio. Por ejemplo audio.mp3.



Paso 4. Añade las siguientes líneas de código en el evento onCreate:

```

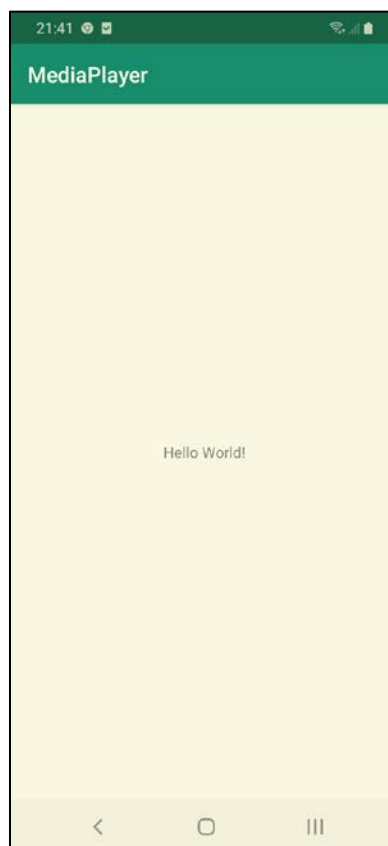
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        MediaPlayer mp = MediaPlayer.create( context: this, R.raw.david_guetta);
        mp.start();
    }
}

```

Paso 5. Ejecuta la aplicación.

Funciona correctamente la reproducción de música



Si deseas parar la reproducción tendrás que utilizar el método `stop()`. Si a continuación quieres volver a reproducirlo utiliza el método `prepare()` y a continuación `start()`. También puedes usar `pause()` y `start()` para ponerlo en pausa y reanudarlo.

Si en lugar de un recurso prefieres reproducirlo desde un fichero utiliza el siguiente código. Observa como en este caso es necesario llamar al método `prepare()`. En el caso anterior no ha sido necesario dado que esta llamada se hace desde `create()`.

```

MediaPlayer mp =new MediaPlayer();
mp.setDataSource(RUTA_AL_FICHERO);
mp.prepare();
mp.start();

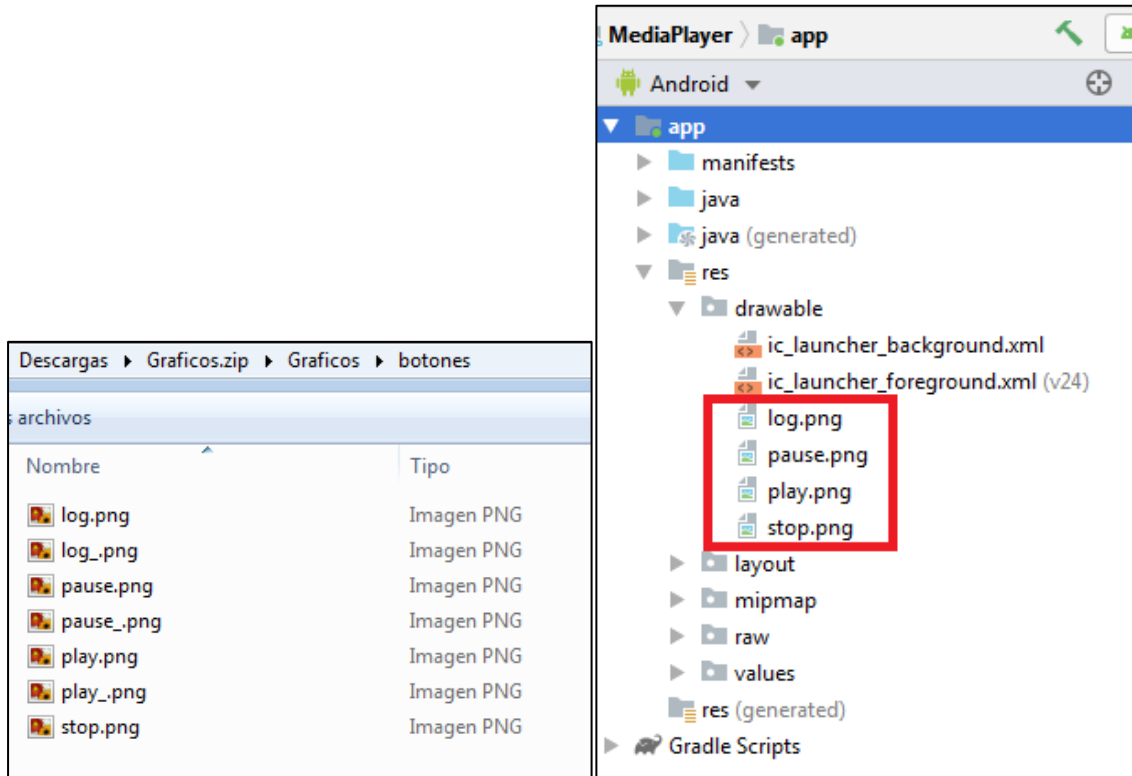
```

Paso 6. Descargar el archivo Graficos.zip de la siguiente url:

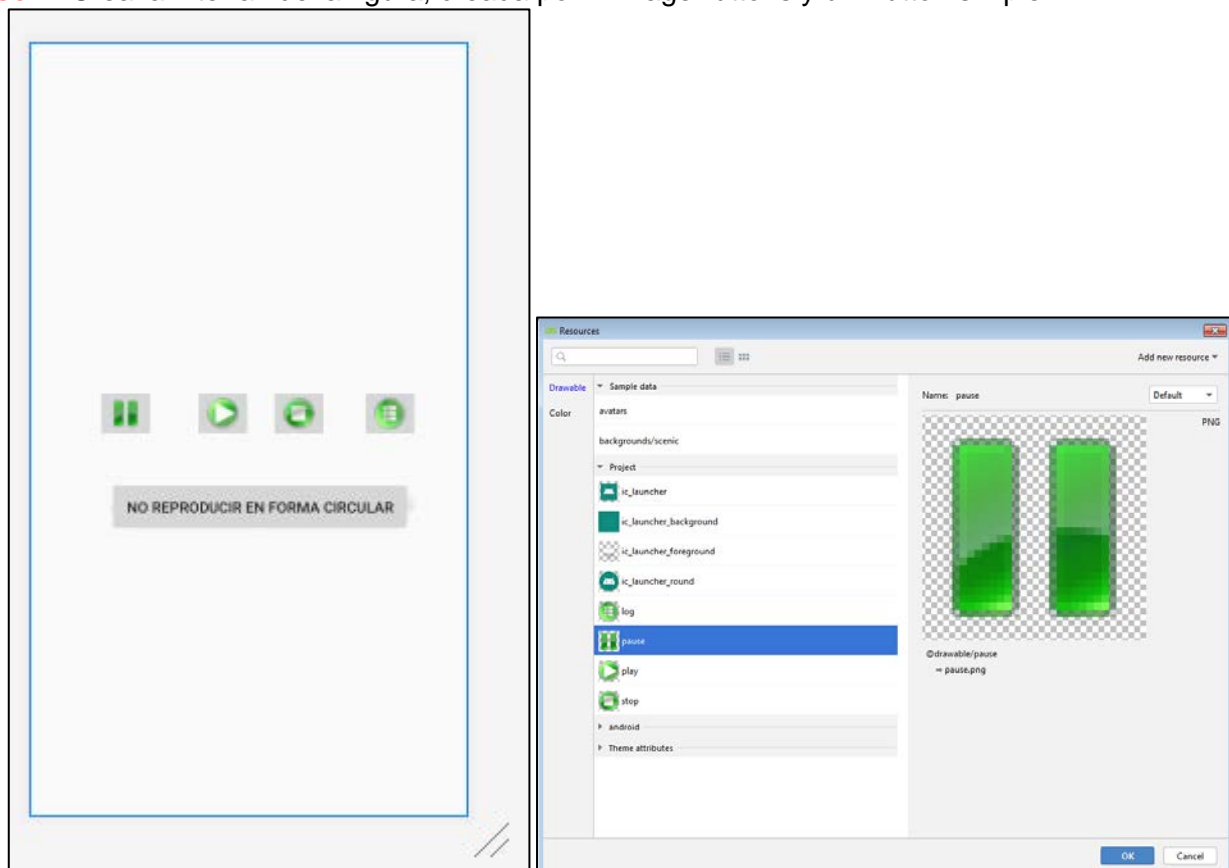
(recuerda que este archivo también está en el Moodle)

<http://www.dcomg.upv.es/~jtomas/android/ficheros/Graficos.zip>

En la carpeta botones accede a los iconos de un reproductor: play, pause, stop, etc y cópialos en el directorio res/drawable del proyecto:



Paso 7. Crea la interfaz de la figura, creada por 4 ImageButtons y un Button simple:



Paso 8. En el fichero MainActivity, definir los siguientes 3 datos miembro dentro de la Clase MainActivity:

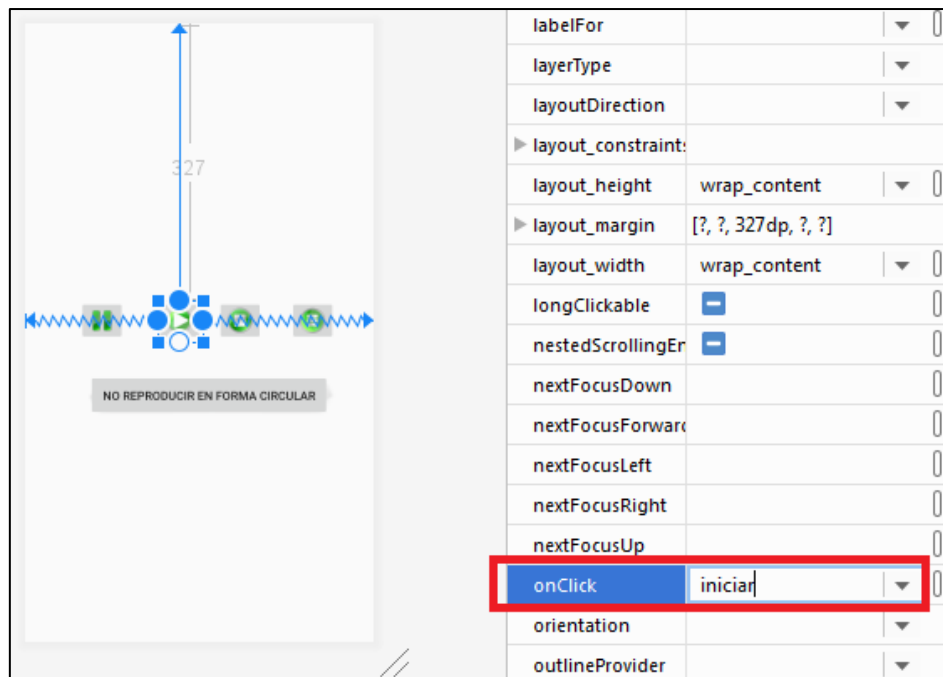
```
public class MainActivity extends AppCompatActivity {  
  
    private MediaPlayer mp;  
    private Button b1;  
    private int posicion=0;  
}
```

Paso 9. Dentro de la función onCreate comentamos la creación del objeto MediaPlayer y enlazamos el elemento button del layout, con su respectivo objeto en Java, mediante la función findViewById:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    //mp = MediaPlayer.create(this, R.raw.david_guetta);  
    //mp.start();  
    b1= (Button) findViewById(R.id.button);  
}
```

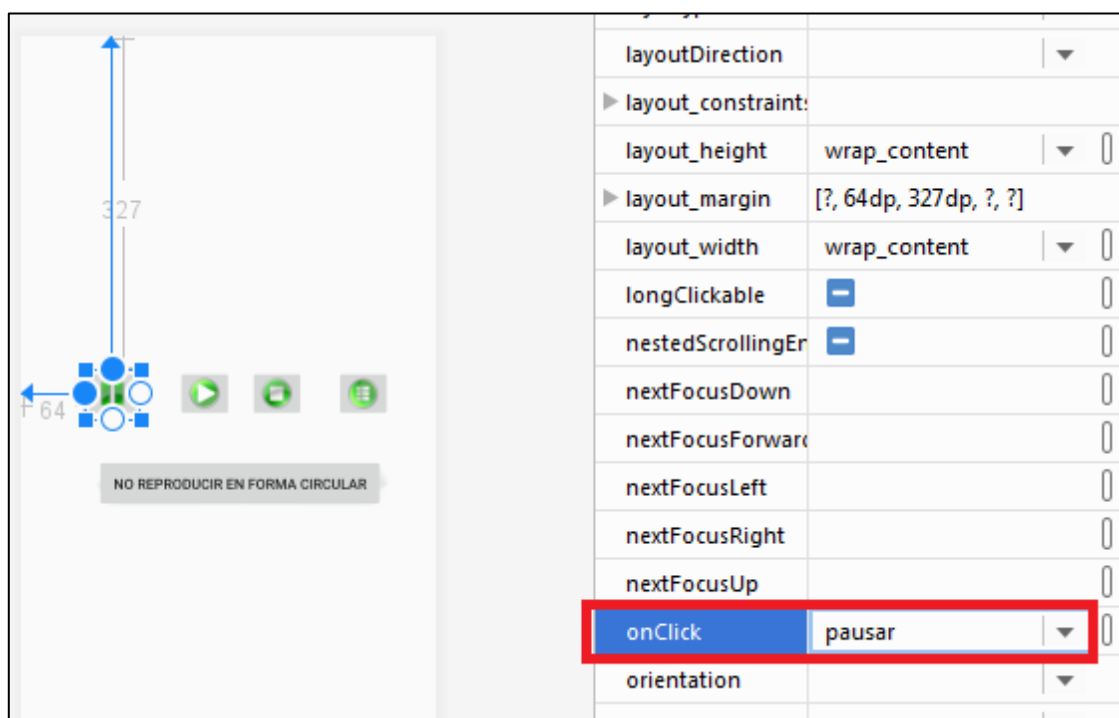
Paso 10. Crearemos la función iniciar que será llamada desde el botón play. Nos permitirá iniciar la melodía en el reproductor mediante la llamada a la función mediaPlayer.start():

```
public void iniciar(View v) {  
    destruir();  
    mp = MediaPlayer.create( context: this, R.raw.david_guetta);  
    mp.start();  
    String op = b1.getText().toString();  
    if (op.equals("No reproducir en forma circular"))  
        mp.setLooping(false);  
    else  
        mp.setLooping(true);  
}  
  
public void destruir() {  
    if (mp != null)  
        mp.release();  
}
```



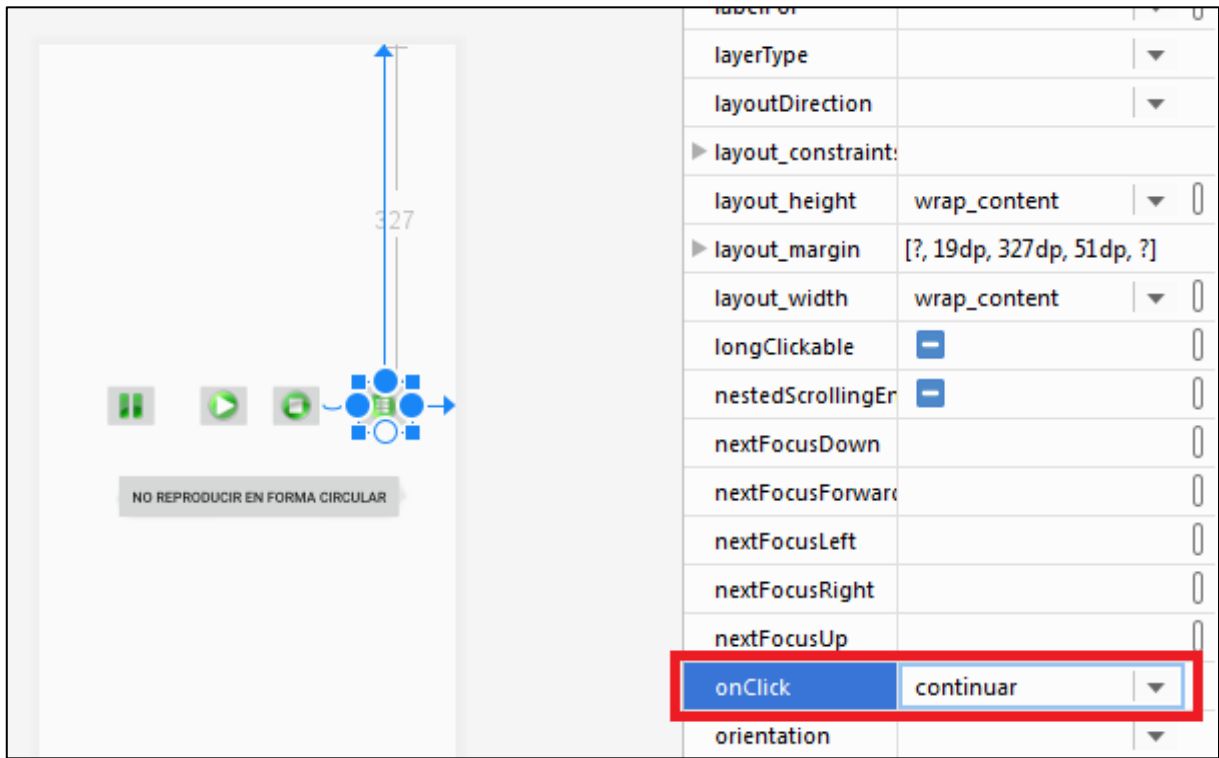
Paso 11. Crearemos la función pausar que será llamada desde el botón pause. Nos permitirá pausar la melodía en el reproductor mediante la llamada a la función `mediaPlayer.pause()`:

```
public void pausar(View v) {
    if (mp != null && mp.isPlaying()) {
        posicion = mp.getCurrentPosition();
        mp.pause();
    }
}
```



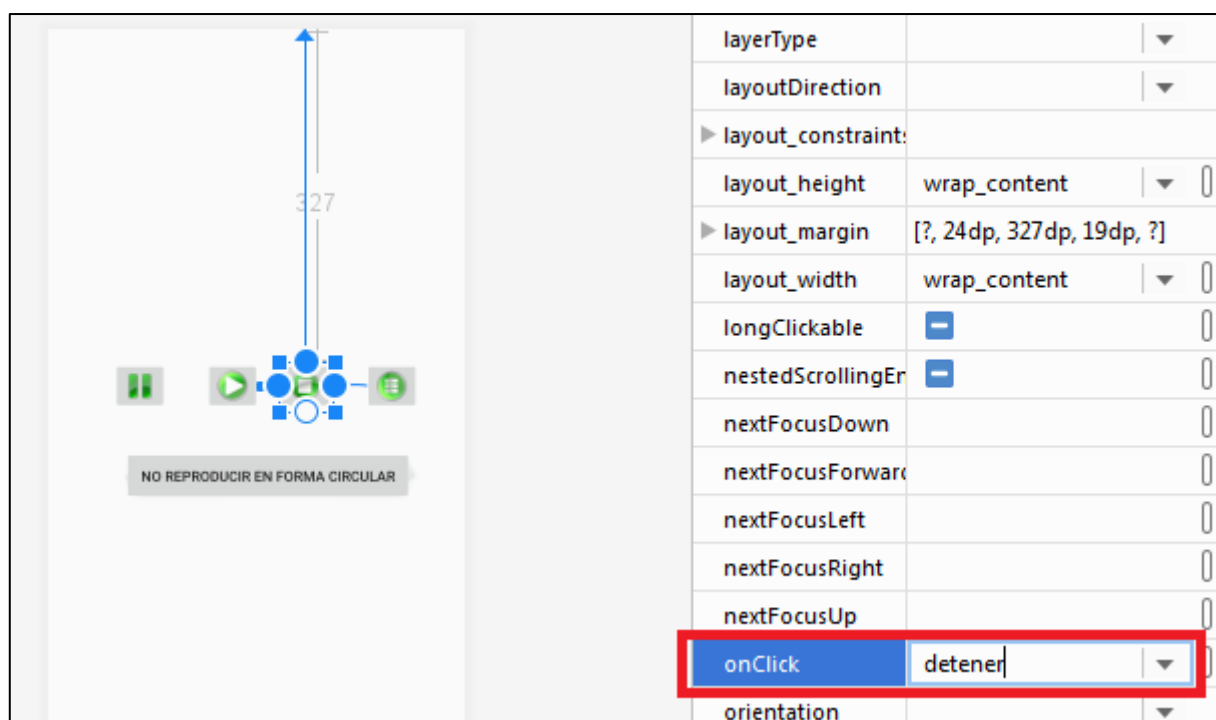
Paso 12. Crearemos la función continuar que será llamada desde el botón continuar, que continuará en la misma posición donde se pausó la canción con llamada a las funciones mediaPlayer.seekTo(pos) y mediaPlayer.start():

```
public void continuar(View v) {
    if (mp != null && mp.isPlaying() == false) {
        mp.seekTo(posicion);
        mp.start();
    }
}
```



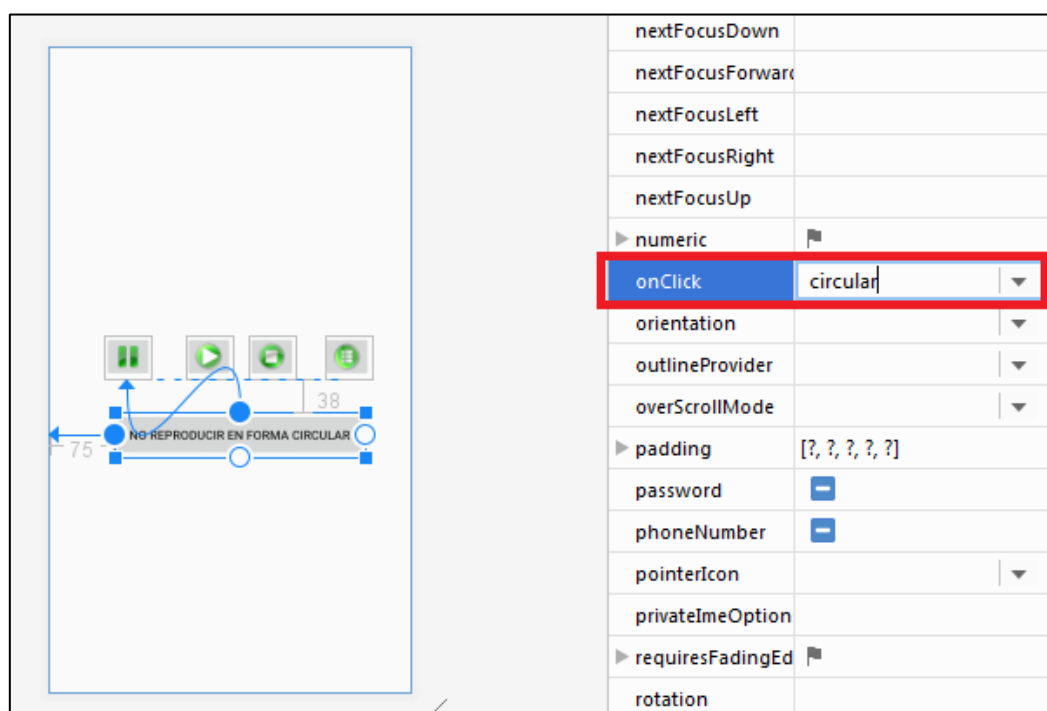
Paso 13. Crearemos la función detener que será llamada desde el botón Stop, que destruirá el objeto multimedia mediante la llamada a las funciones mediaPlayer.stop():

```
public void detener(View v) {
    if (mp != null) {
        mp.stop();
        posicion = 0;
    }
}
```



Paso 14. Crearemos la función circular que será llamada desde el botón que decide si la reproducción se realiza de forma circular o no:

```
public void circular(View v) {
    detener(v);
    String op = b1.getText().toString();
    if (op.equals("No reproducir en forma circular"))
        b1.setText("Reproducir en forma circular");
    else
        b1.setText("No reproducir en forma circular");
}
```



Paso 15. Ejecuta la aplicación

