

Actividad [#1] - [Instalación de Android Studio y Diseño de Aplicación] [Desarrollo de Aplicaciones Móviles 2] Ingeniería en Desarrollo de Software

Tutor: Eduardo Israel Castillo García

Alumno: Ricardo Rivas Rocha

Fecha: 29-noviembre-2024

Índice

Portada Página 1

Índice Página 2

Introducción Página 3

Descripción Página 4

Justificación Página 5

Desarrollo: Página 6 a 21

- Interfaz
- Codificación
- Prueba de la Aplicación

Conclusión Página 22

Referencias Página 23

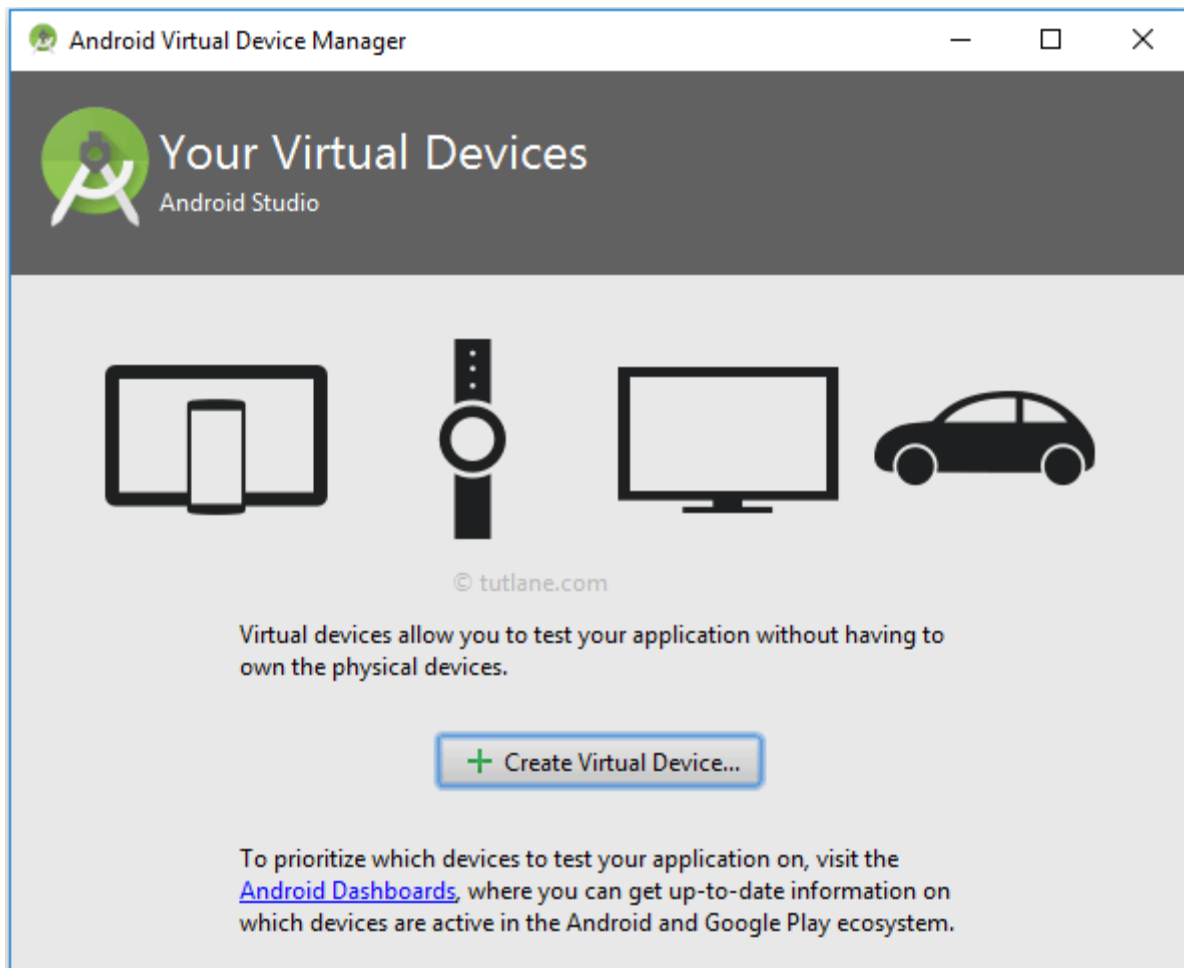
Introducción

En este primer ejercicio lo que vamos a realizar es instalar una herramienta de uso móvil que nos permita crear aplicaciones ya que en esta misma la vamos a ocupar para la temporada del buen fin y requerimos hacer una aplicación sencilla en donde podamos darle una galería de estos mismos sonidos para que los escuchen y descarguen este momento lo que vamos a configurar es la herramienta de desarrollo de aplicaciones ya configurada empezaremos a trabajar diseñando esta misma y añadiéndole algunas funcionalidades que nos pide este primer trabajo ya que hoy en día los clientes necesitan que una aplicación se adapte a cada uno de ellos y que no les moleste algún diseño o función de estas mismas ya que si no estos mismos no ocuparían dicha aplicación.



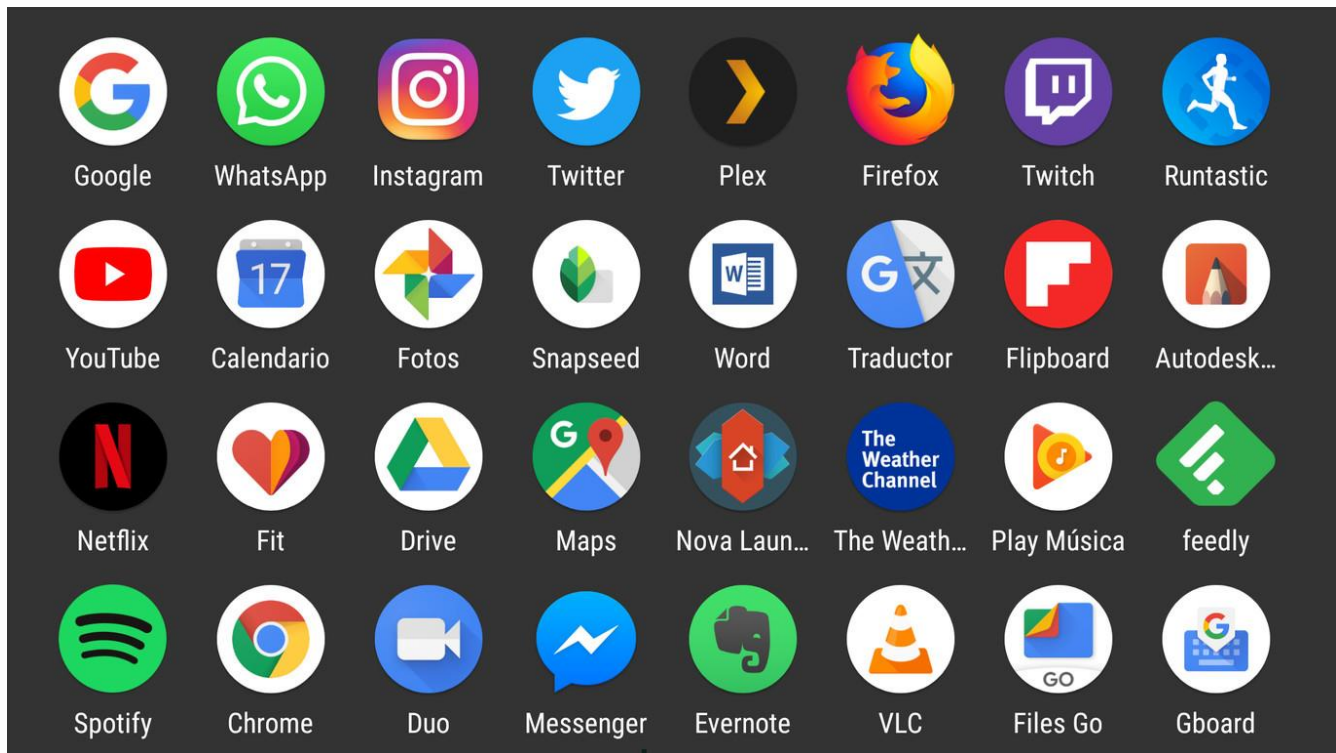
Descripción

En esta primera actividad lo que vamos a realizar es instalar la aplicación de Android Studio esta nos permitirá crear aplicaciones móviles que hoy vemos en día en la Play store ya que vamos a requerir desarrollar un proyecto en donde los clientes puedan tener una galería de sonidos y que estos mismos puedan escucharlos por que es muy importante que la aplicación funcione correcto por esto mismo configuraremos la aplicación de Android Studio para que mejor se adapte a esta actividad en esta aplicación tendremos 10 sonidos y estos mismos los vamos a anear programándolos en el lenguaje de java y en un archivo de diseño de XML la interfaz grafica de esta misma tomaremos capturas de todo el procedimiento de principio a fin en donde nos daremos cuenta de algunas funciones que tiene esta gran herramienta de desarrollador al finalizar también vamos a probar que funciona correctamente y el diseño sea el adecuado para que los clientes se sientan cómodos con este mismo acompañado de igual manera con las funcionalidades que por el momento son unas pocas.



Justificación

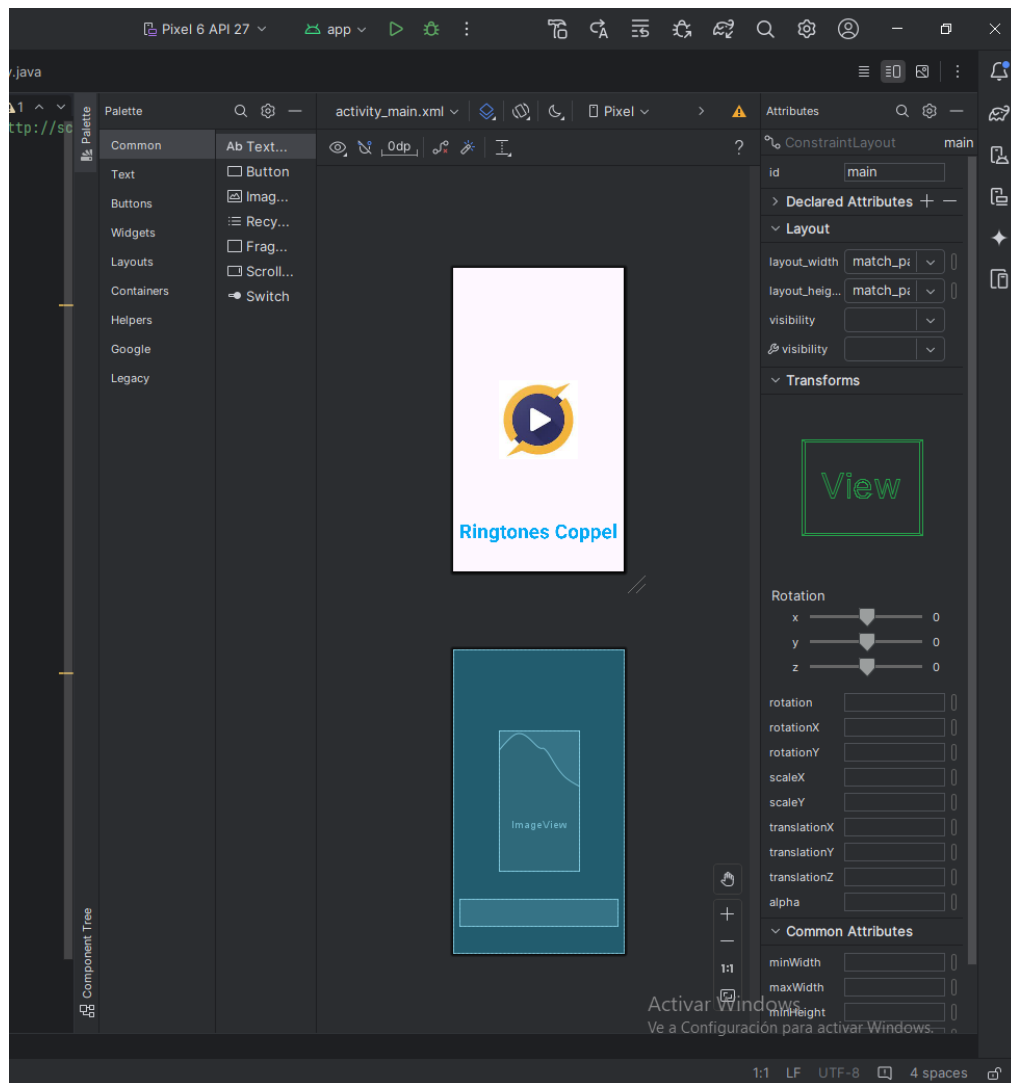
Esta actividad nos va a servir para conocer mas el entorno de desarrollo de aplicaciones móviles y explorar la variedad de funcionalidades que nos permite tener Android Studio ya que hoy en día es muy importante tener una aplicación en donde una persona pueda interactuar con esta misma aportándole un valor para ella y no solo las personas si no grandes empresas o pequeños negocios ya que en la actualidad existen infinidad de estas mismas que nos ayudan a facilitar algunos procesos como por ejemplo el inventario de alguna tienda, búsquedas de clientes o consultar tu saldo o hasta poder reservar o hacer compras desde una aplicación móvil siento que esto revoluciono la industria de las telecomunicaciones aparte de esto se han adaptado y desarrollado juegos de igual manera para estos mismos aparatos portátiles si es que la persona desea estar entretenida de alguna manera e informada como en este caso es el Facebook, WhatsApp o una aplicación para que las organizaciones compartan ideas en reuniones ya es vital tener de cajón todas estas herramientas.



Desarrollo:

- **Interfaz**

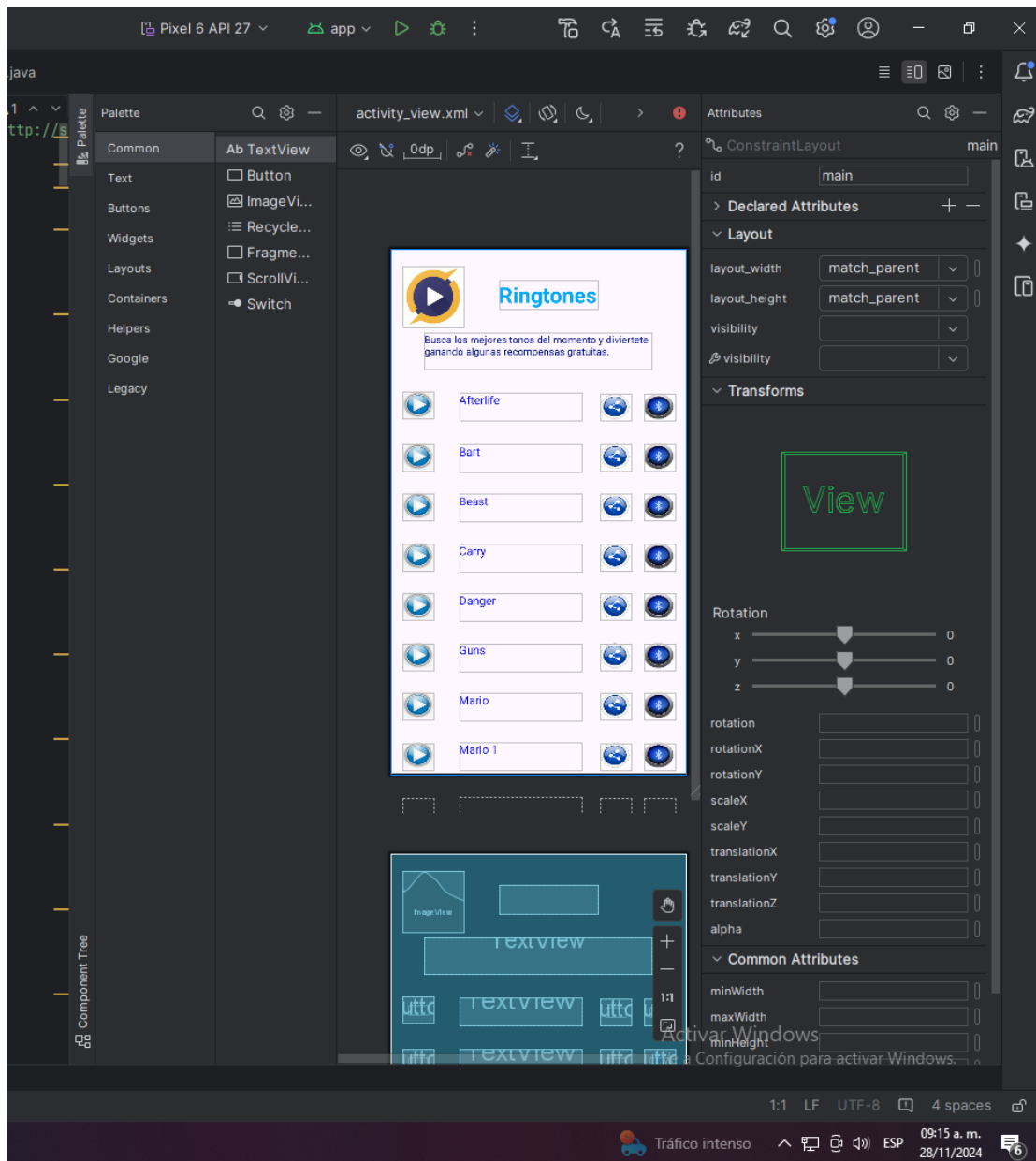
Como podemos observar en la interfaz hemos creado el `activity_main` de la pantalla de carga se podría decir esta va a hacer el efecto en inicial al entrar a la aplicación junto con la ayuda de una imagen y un cuadro de texto podremos observar la misma interfaz.



Desarrollo:

- Interfaz

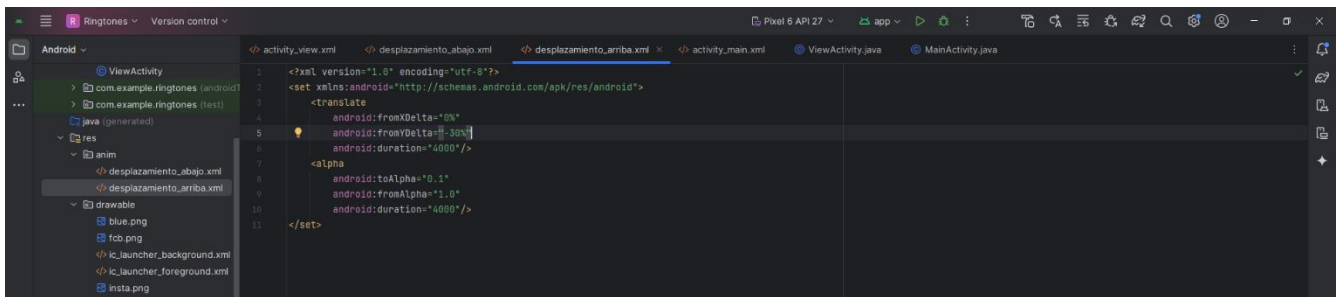
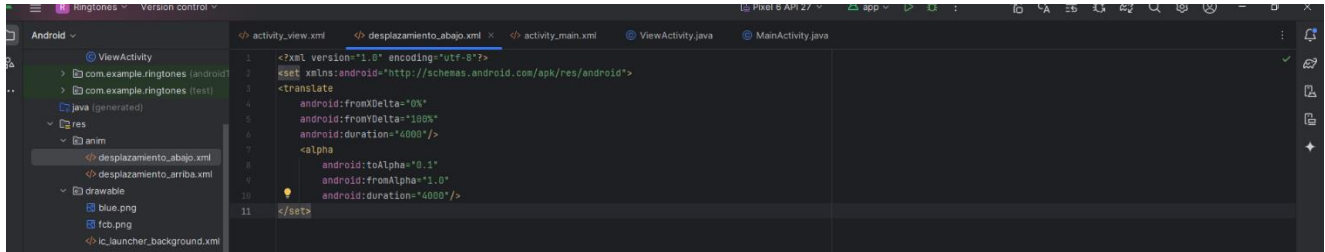
En esta otra interfaz llamada viewActivity es donde vamos a hacer el diseño de nuestra página de ringtones en donde agregamos los recursos que creamos en drawable que son las imágenes.png en este caso las que importe fueron el logo lo puse en la esquina superior izquierda le puse un título de una caja de texto llamado Ringtones después agregue otra caja de texto con la descripción de la aplicación después hice un botón de play, caja de texto y otros 2 botones más importando las imágenes de compartir y bluetooth así le hice a mis 10 canciones.



Desarrollo:

- **Codificación**

En esta parte les mostrare el código de las animaciones de la pantalla de carga esto lo hice creando un nuevo archivo llamado animación y elabore 2 que se llamaron animación desplazamiento arriba y desplazamiento abajo ambos de tipo animación.

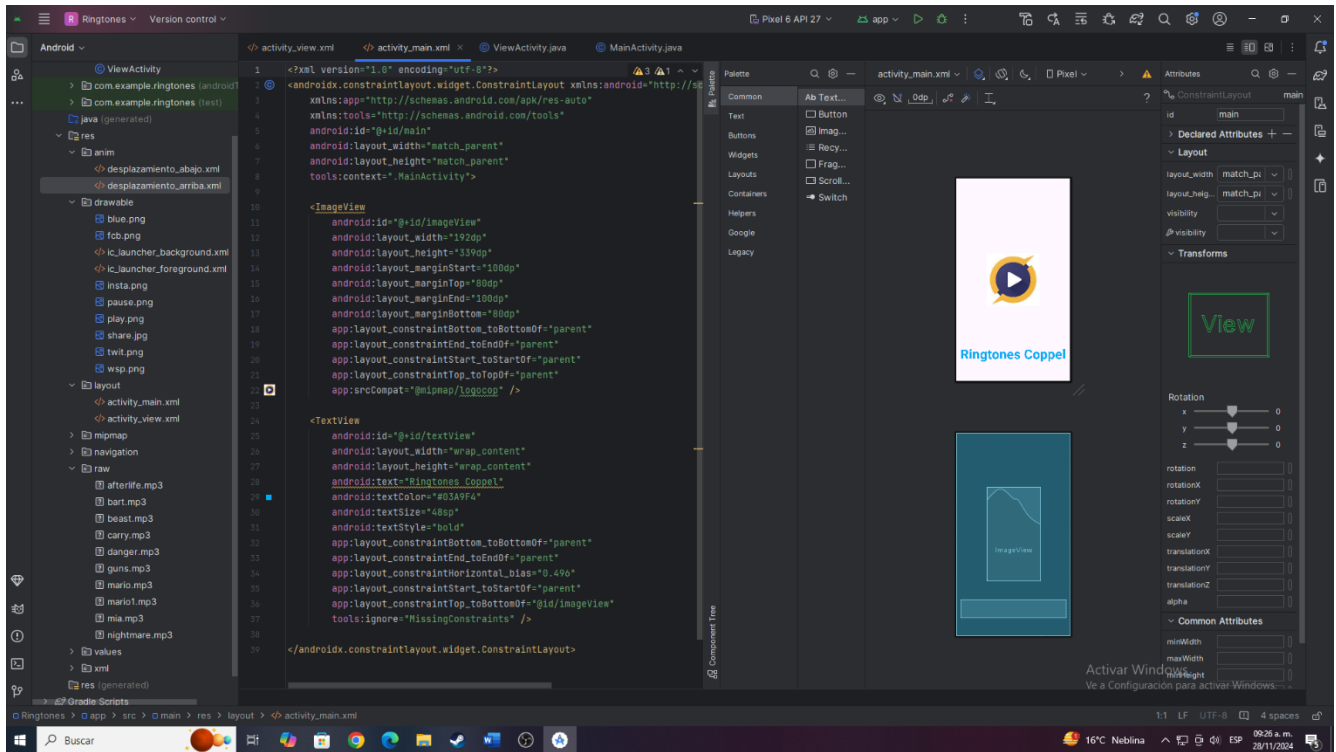


Estos los creamos desde la carpeta anim esto nos ayudará mismamente con la animación de la pantalla de inicio le insertamos los valores de duración y de donde a donde se estará moviendo esta misma.

Desarrollo:

- Codificación

Luego nos vamos al archivo de diseño que es el XML añadiremos el botón y el cuadro de vista de texto.



Después nos dirigimos al archivo de java de `MainActivity` que en este le vamos a dar la funcionalidad de la animación e invocar lo del XML.

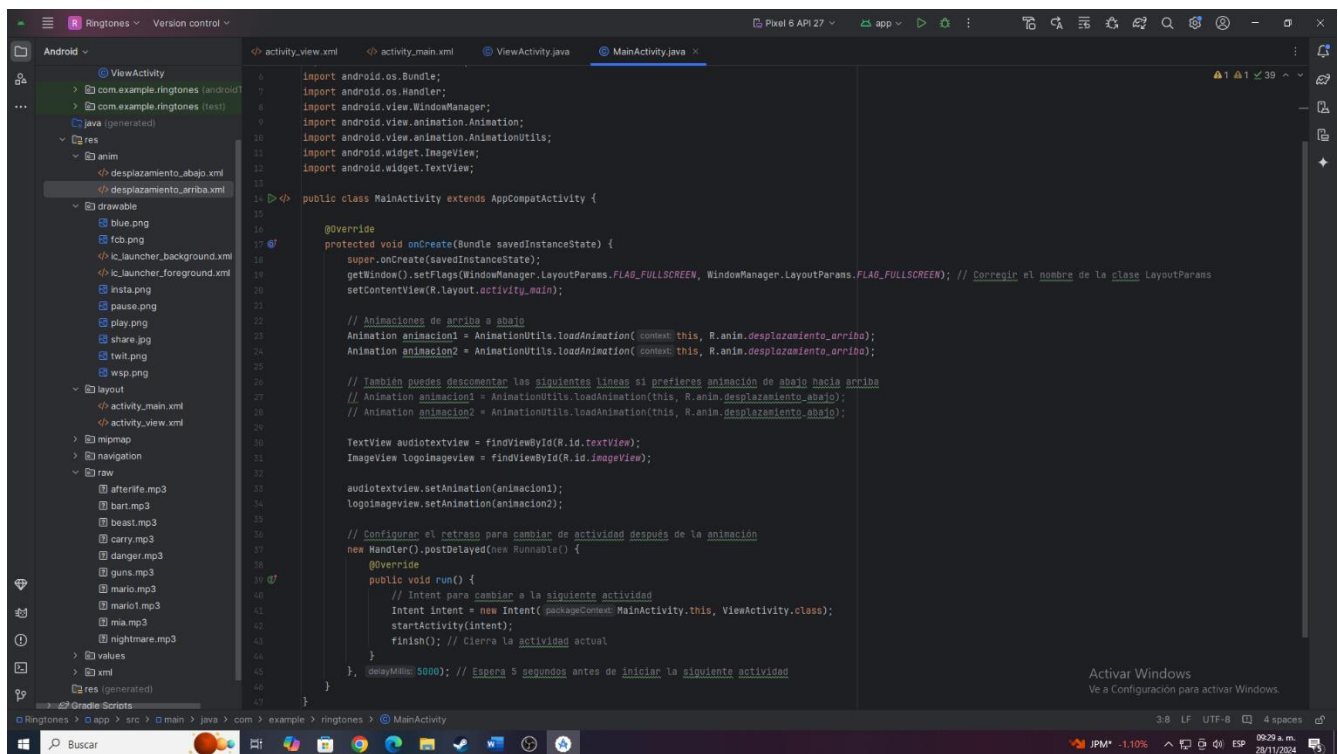
Desarrollo:

- Codificación

Aquí vamos a añadir las librerías necesarias para hacer las funciones del MainActivity vamos referenciar las 2 animaciones creadas anteriormente y la vista de la imagen y el cuadro de texto esto para que se cree la funcionalidad de la animación los cuales son llamados:

```
TextView audiotextview = findViewById(R.id.textView);
ImageView logoimageview = findViewById(R.id.imageView);
```

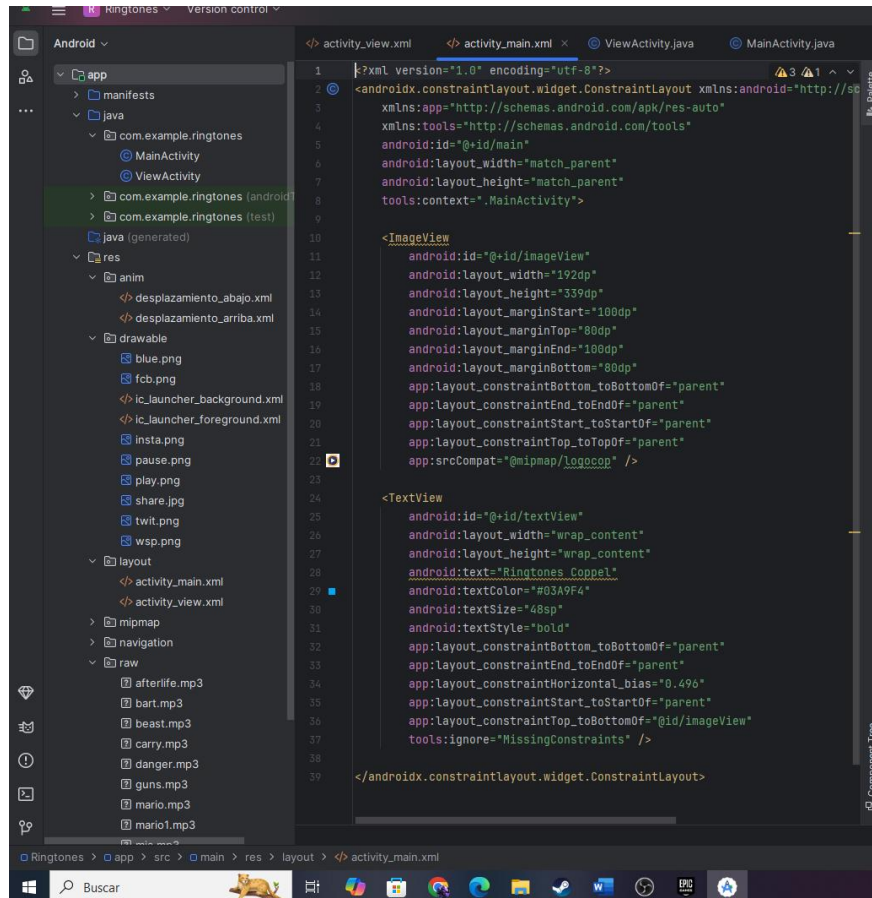
Después vamos a crear un temporizador de la animación que va a tener una duración de 5 milisegundos y nos va a ayudar a que se vea bien la animación en lo que carga el menú de ringtones.



Desarrollo:

- Codificación

Por otra parte, les mostrare la codificación del diseño de la pantalla de activitymain ya que en esta misma agregamos lo que son la codificación de la imagen referenciando esta misma desde el palette de Android Studio y la cajita de texto llamada ringtones Coppel.

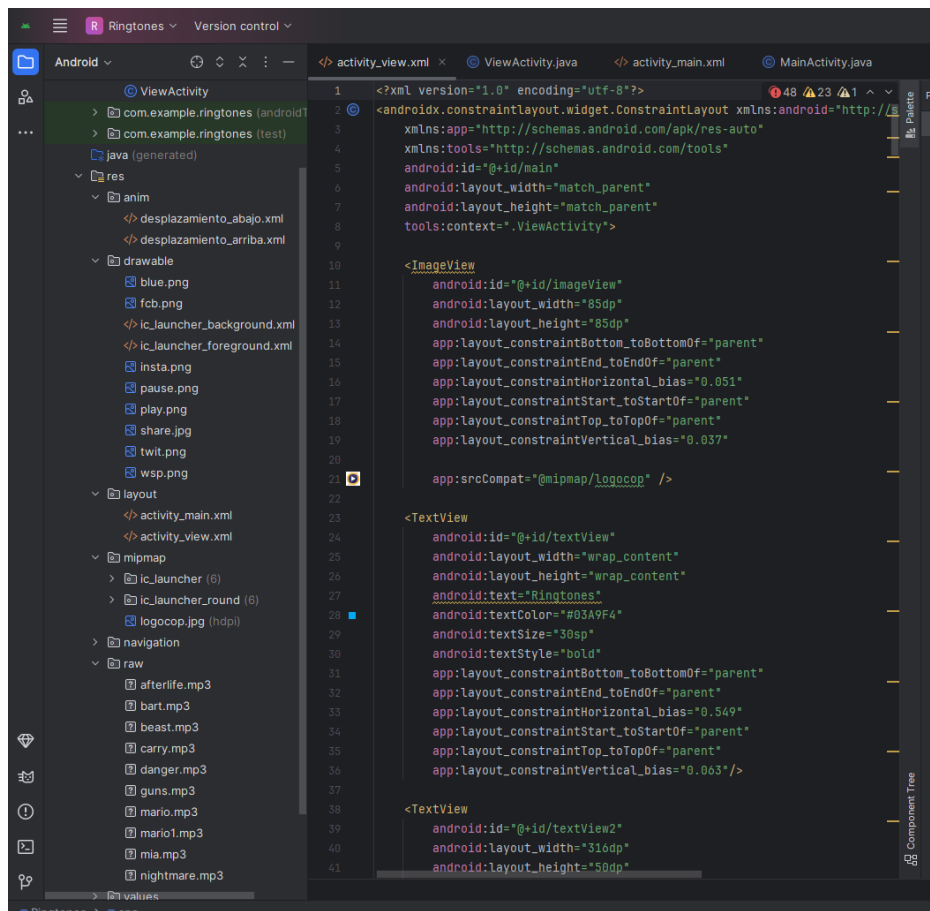


Desarrollo:

- **Codificación**

Después de esto vamos a crear una nueva actividad desde la opción app esta se llamara activityView en donde se alojara nuestro menú principal de la aplicación de ringtones se nos crearan 2 archivos uno en formato XML como lo mencionamos es el diseño de la aplicación y un archivo java que le añada funcionalidad a nuestra aplicación.

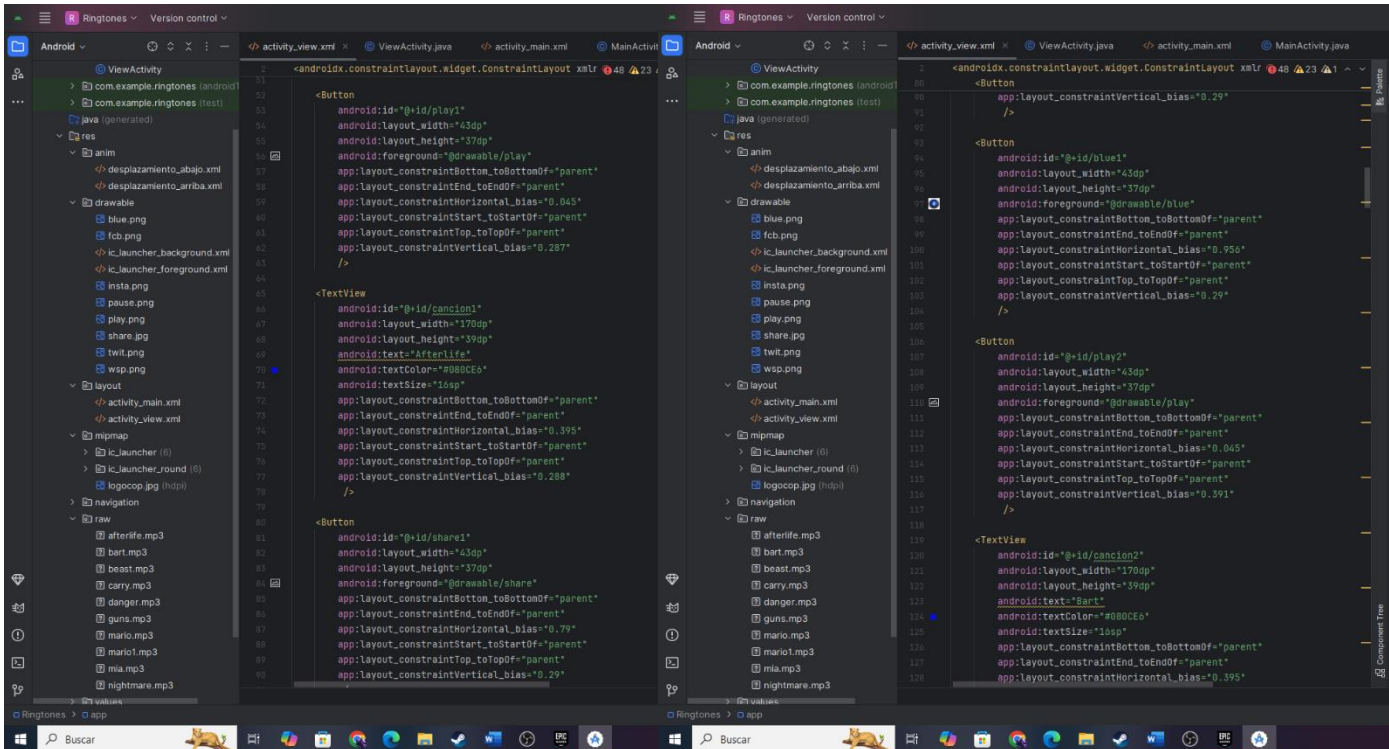
Primero que nada vamos a configurar lo que es el código en XML en donde se va a agregar el contenido grafico de la aplicación lo primero que realizaremos es insertar una imagen que desde mipmap en esta ocasión se llamara logo claro ya exportándola dentro de esta carpeta agregando una Imageview, luego de esto creamos un text view en donde añadiremos lo que es el nombre de Ringtones, y otra con la descripción de la aplicación.



Desarrollo:

• Codificación

Después de esto empezaremos codificando y agregando las respectivas imágenes de recursos de la carpeta drawable en este caso se estará codificando un botón con el nombre de play1, luego un TextView en donde ira nuestro nombre de la canción enseguida agregamos otro botón llamado share1 y otro llamado blue 1 en total repetiremos todo el código a excepción de la ubicación que es la que le dará forma a nuestra aplicación.

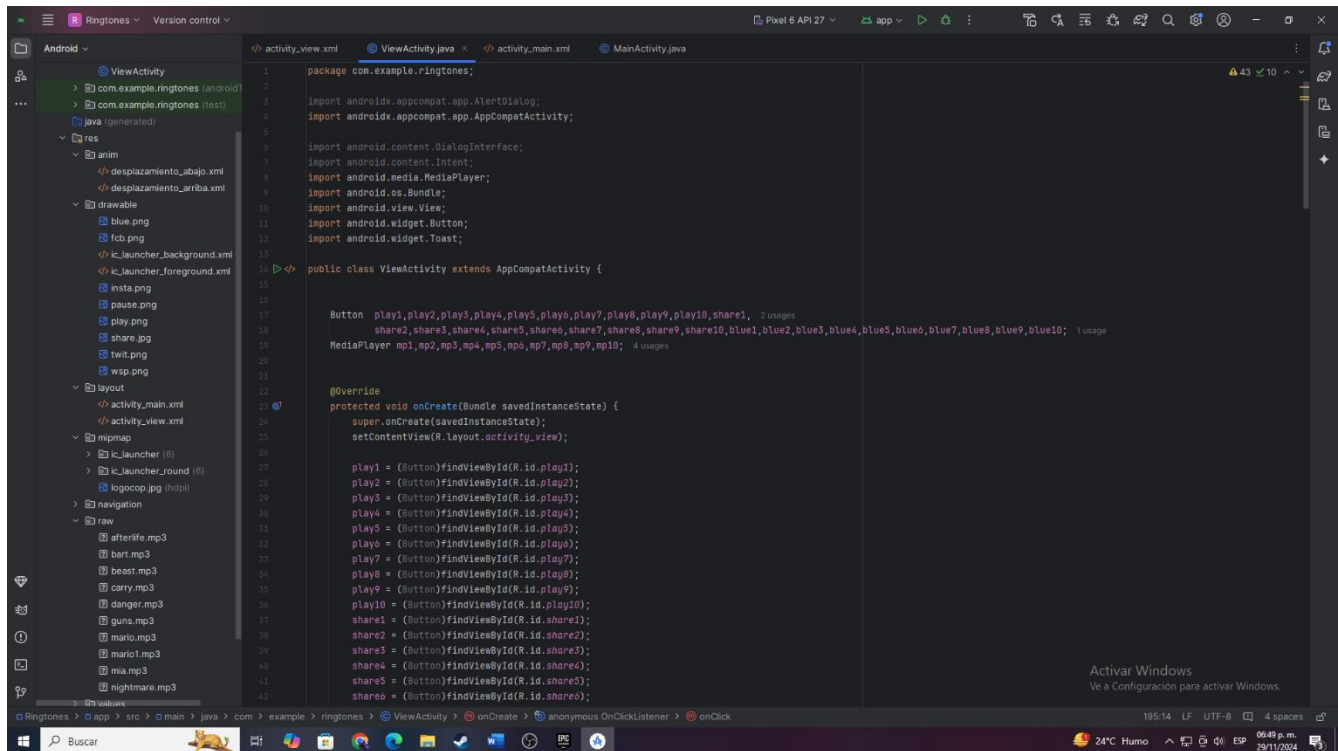


Desarrollo:

- Codificación

Lo primero que realizaremos es importar las librerías con los respectivos imports en java y declaramos las variables Button que en este caso va a hacer play1, play2, play3 hasta el 10, seguido de esto declaramos las variables de share1, share2, share3 igual hasta el 10, seguido de esto las variables de blue1, blue2, blue3 hasta el 10.

Seguido declaramos las variables de MediaPlayer que en este caso va desde mp1 hasta el mp10.



```
package com.example.ringtones;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class ViewActivity extends AppCompatActivity {

    Button play1,play2,play3,play4,play5,play6,play7,play8,play9,play10,share1,
    share2,share3,share4,share5,share6,share7,share8,share9,share10,blue1,blue2,blue3,blue4,blue5,blue6,blue7,blue8,blue9,blue10;
    MediaPlayer mp1,mp2,mp3,mp4,mp5,mp6,mp7,mp8,mp9,mp10;

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

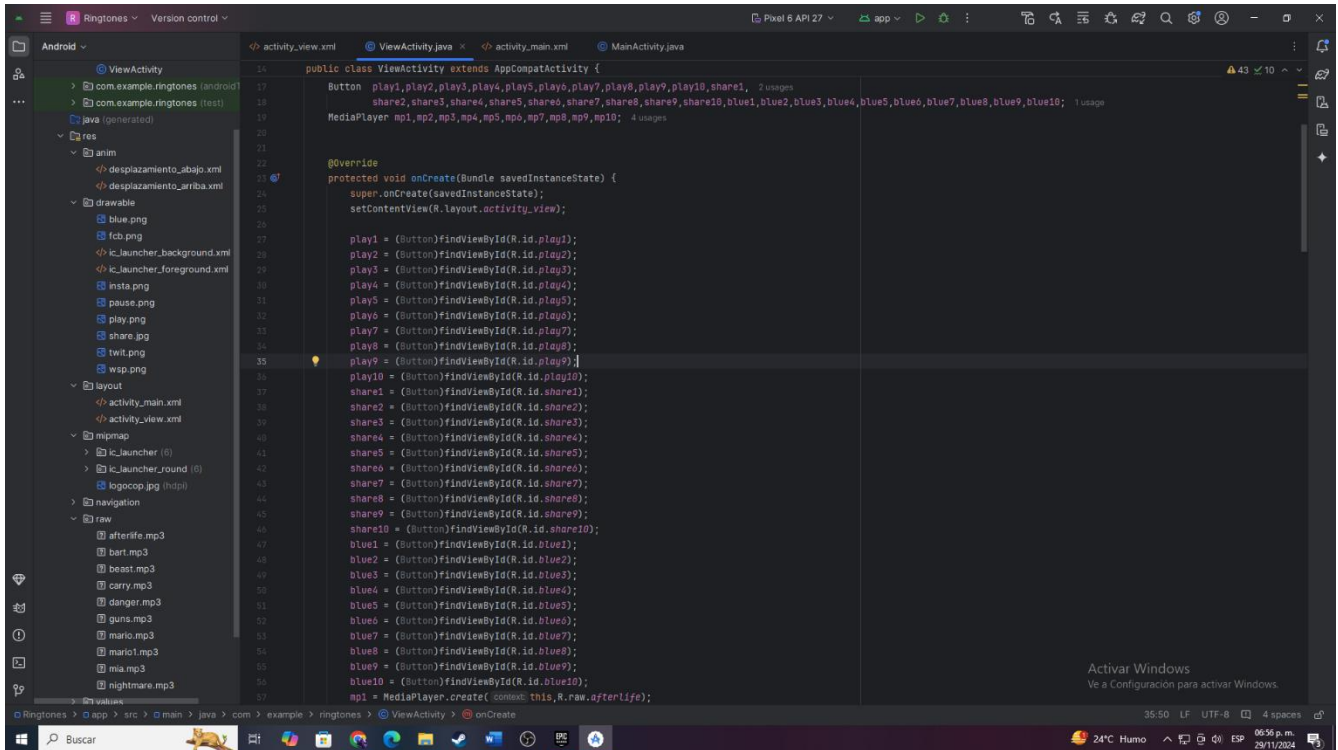
        play1 = (Button)findViewById(R.id.play1);
        play2 = (Button)findViewById(R.id.play2);
        play3 = (Button)findViewById(R.id.play3);
        play4 = (Button)findViewById(R.id.play4);
        play5 = (Button)findViewById(R.id.play5);
        play6 = (Button)findViewById(R.id.play6);
        play7 = (Button)findViewById(R.id.play7);
        play8 = (Button)findViewById(R.id.play8);
        play9 = (Button)findViewById(R.id.play9);
        play10 = (Button)findViewById(R.id.play10);
        share1 = (Button)findViewById(R.id.share1);
        share2 = (Button)findViewById(R.id.share2);
        share3 = (Button)findViewById(R.id.share3);
        share4 = (Button)findViewById(R.id.share4);
        share5 = (Button)findViewById(R.id.share5);
        share6 = (Button)findViewById(R.id.share6);
```


Desarrollo:

- Codificación

En este caso vamos asignarles sus respectivas funcionalidades como estas donde el botón play1 se va a referenciar con play1 y así mismo para share1 y blue1 de nuevo hasta asignarles a todas las variables:

```
play1 = (Button)findViewById(R.id.play1);
```



Desarrollo:

- Codificación

Luego en las variables de media player vamos a instanciar los tonos que descargamos en los recursos desde la carpeta que creamos que fue raw en este caso puse en orden la codificación del XML de acuerdo a cada canción ya que si no se me haría difícil identificarlas y asignarles sus respectivas funciones en este caso se almacenaran las variables de las canciones en una variable mp1, mp2, mp3 hasta el mp10 con los tonos que añadimos.

```
14 public class MainActivity extends AppCompatActivity {
15     protected void onCreate(Bundle savedInstanceState) {
16         // ...
17         share9 = (Button)findViewById(R.id.share9);
18         share10 = (Button)findViewById(R.id.share10);
19         blue1 = (Button)findViewById(R.id.blue1);
20         blue2 = (Button)findViewById(R.id.blue2);
21         blue3 = (Button)findViewById(R.id.blue3);
22         blue4 = (Button)findViewById(R.id.blue4);
23         blue5 = (Button)findViewById(R.id.blue5);
24         blue6 = (Button)findViewById(R.id.blue6);
25         blue7 = (Button)findViewById(R.id.blue7);
26         blue8 = (Button)findViewById(R.id.blue8);
27         blue9 = (Button)findViewById(R.id.blue9);
28         blue10 = (Button)findViewById(R.id.blue10);
29
30         mp1 = MediaPlayer.create(context, this.R.raw.afterlife);
31         mp2 = MediaPlayer.create(context, this.R.raw.bart);
32         mp3 = MediaPlayer.create(context, this.R.raw.beast);
33         mp4 = MediaPlayer.create(context, this.R.raw.beyonce);
34         mp5 = MediaPlayer.create(context, this.R.raw.danger);
35         mp6 = MediaPlayer.create(context, this.R.raw.guns);
36         mp7 = MediaPlayer.create(context, this.R.raw.mario);
37         mp8 = MediaPlayer.create(context, this.R.raw.mario1);
38         mp9 = MediaPlayer.create(context, this.R.raw.mia);
39         mp10 = MediaPlayer.create(context, this.R.raw.nightmare);
40
41         // ...
42         // Extract Surround # ...
43         // ...
44         // ...
45         // ...
46         // ...
47         // ...
48         // ...
49         // ...
50         // ...
51         // ...
52         // ...
53         // ...
54         // ...
55         // ...
56         // ...
57         // ...
58         // ...
59         // ...
60         // ...
61         // ...
62         // ...
63         // ...
64         // ...
65         // ...
66         // ...
67         // ...
68         // ...
69         // ...
70         // ...
71         // ...
72         // ...
73         // ...
74         // ...
75         // ...
76         // ...
77         // ...
78         // ...
79         // ...
80         // ...
81         // ...
82         // ...
83         // ...
84         // ...
85         // ...
86         // ...
87         // ...
88         // ...
89         // ...
90         // ...
91         // ...
92         // ...
93         // ...
94         // ...
95         // ...
96         // ...
97         // ...
98         // ...
99         // ...
100        // ...
101    }
102 }
```

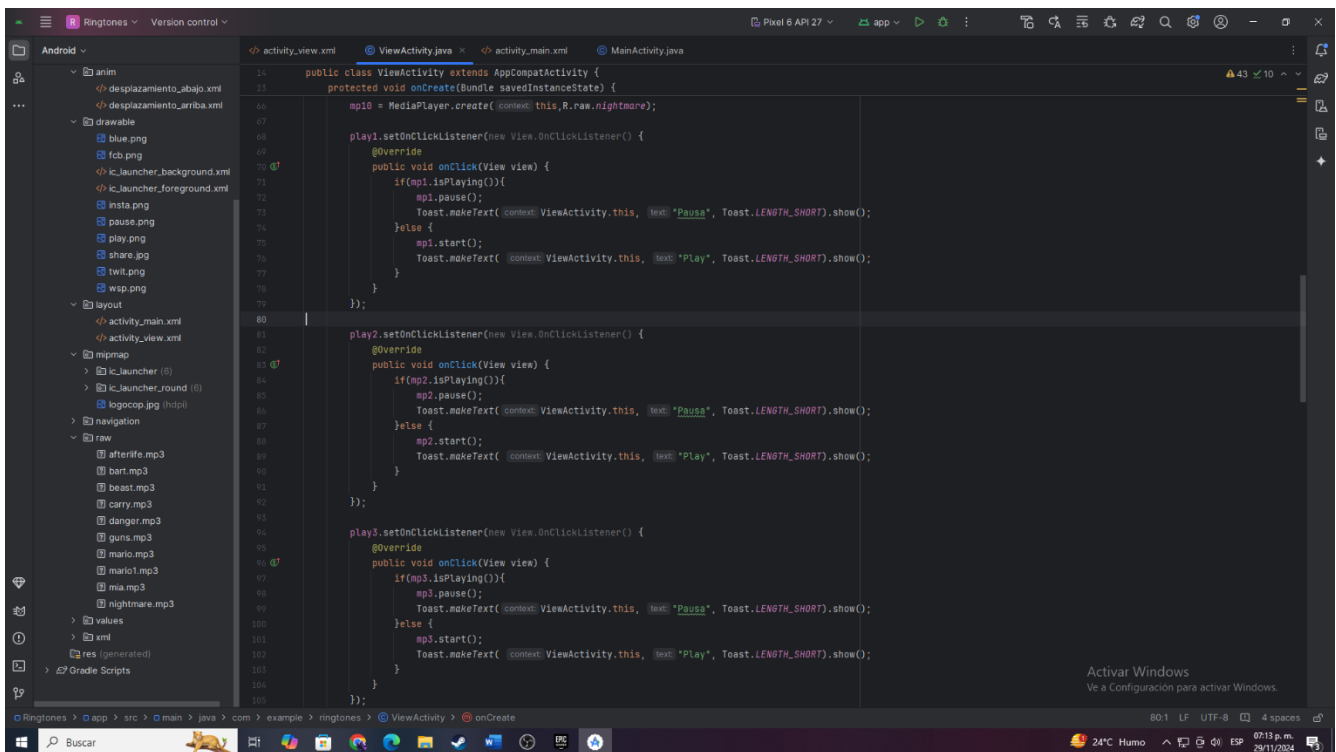
Desarrollo:

- Codificación

Después añadiremos la función OneClick esta nos permitirá que al accionar el botón de play se pueda pausar el tono y las configuraremos de la siguiente forma:

```
play1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if(mp1.isPlaying()){  
            mp1.pause();  
            Toast.makeText(ViewActivity.this, "Pausa", Toast.LENGTH_SHORT).show();  
        }else {  
            mp1.start();  
            Toast.makeText( ViewActivity.this, "Play", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

Este pedacito de código nos indica que si se esta reproduciendo el sonido lo pausa en este caso en el botón de play1 estará esta variable y por otra parte la otra condición que mp1 en donde accionaremos el botón de play, así le vamos a hacer con el botón play2 le añadiremos la funcionalidad de pausa con la variable mp2 y así sucesivamente hasta llegar al play10 función mp10.



Desarrollo:

- **Codificación**

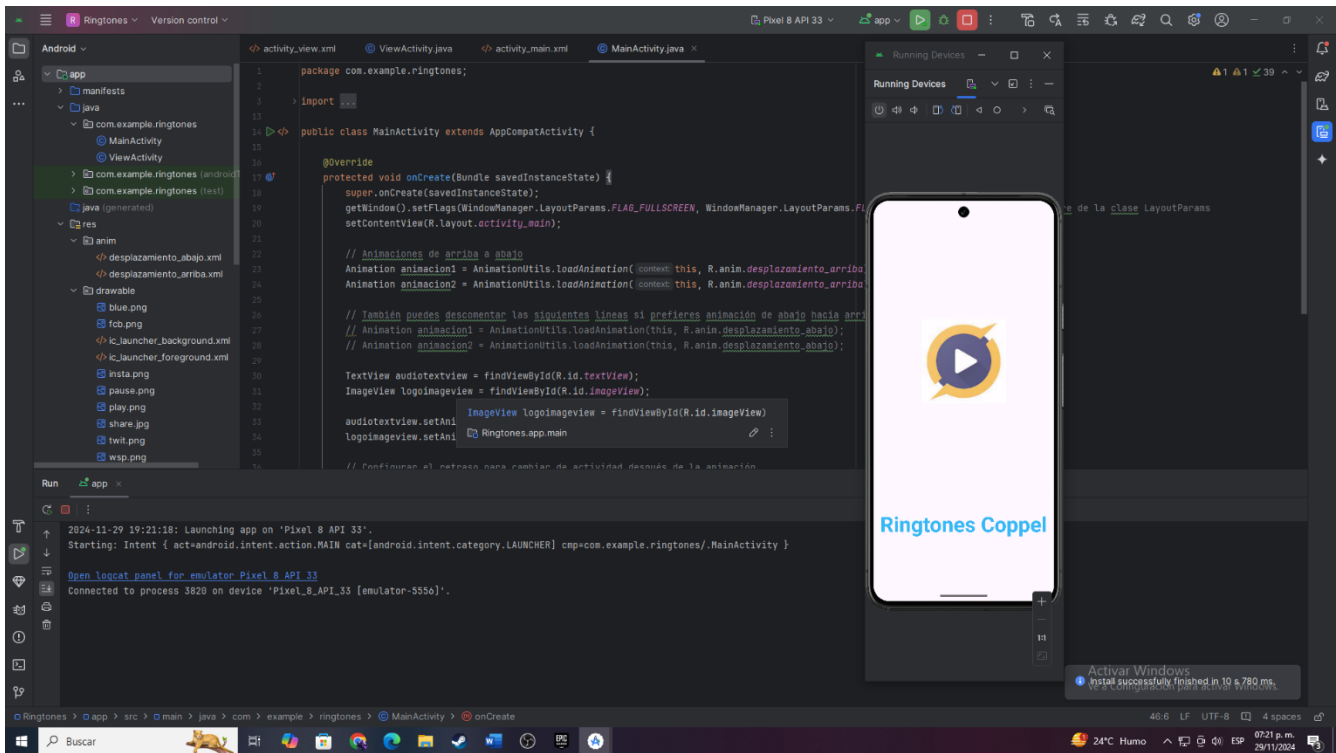
Otra cosa importante para crear el puente de una pantalla a otra referenciamos en este caso `ViewActivity` esto nos permitirá que después de la animación nos mande a la pagina principal de la aplicación de Ringtones.

```
new Handler().postDelayed(new Runnable() {  
    @Override  
    public void run() {  
        // Intent para cambiar a la siguiente actividad  
        Intent intent = new Intent(MainActivity.this, ViewActivity.class);  
        startActivity(intent);  
        finish(); // Cierra la actividad actual  
    }  
}, 5000); // Espera 5 segundos antes de iniciar la siguiente actividad
```

Desarrollo:

- Prueba de la Aplicación

En este caso probaremos la aplicación desde un dispositivo virtual en este caso ya lo tengo desde Android Studio de igual manera configurado.

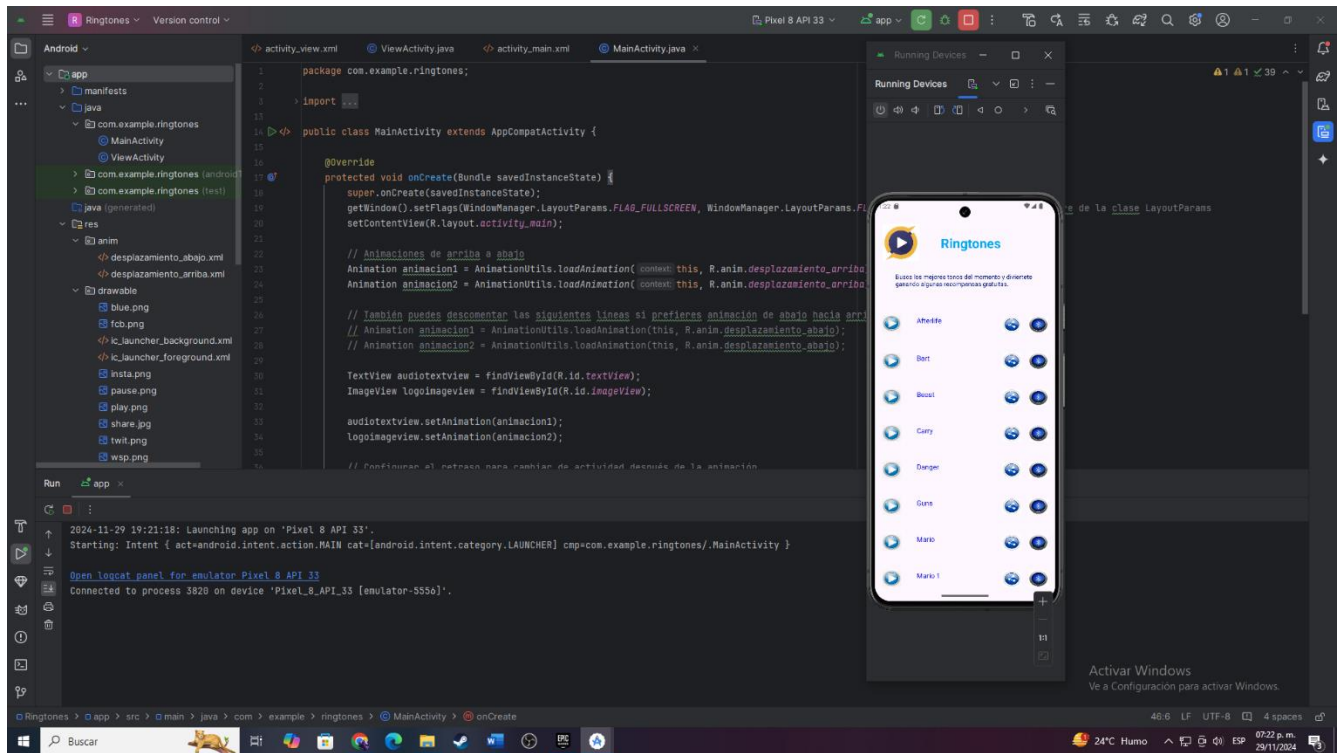


En esta primera captura observamos la animación de la aplicación del MainActivity.

Desarrollo:

- Prueba de la Aplicación

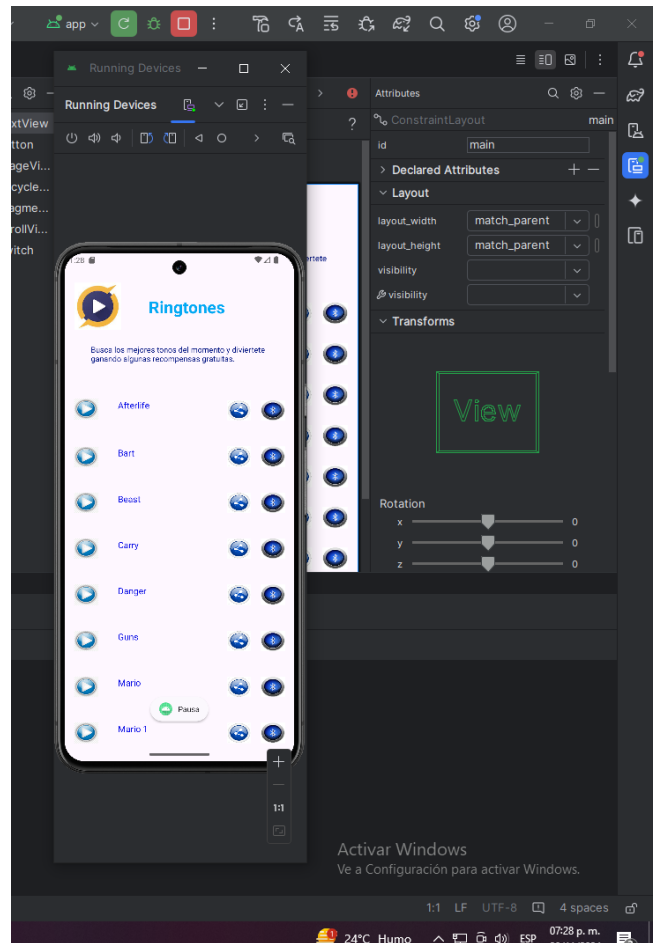
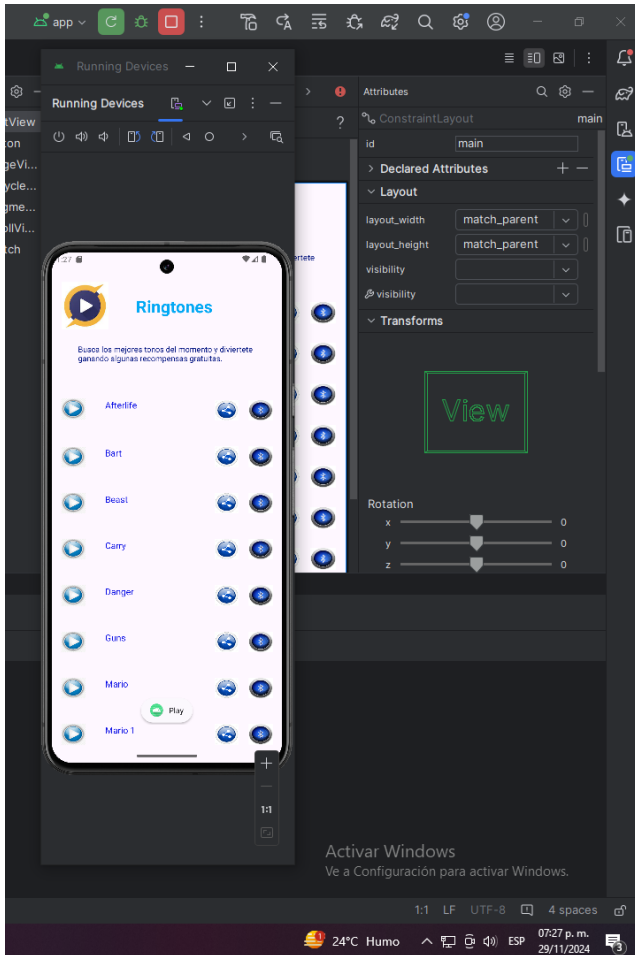
En esta segunda captura de pantalla observamos la pantalla de ActivityView que nos muestra el menú principal de los tonos descargados anteriormente.



Desarrollo:

- Prueba de la Aplicación

En esta última captura le vamos a dar clic en el botón de play y nos reproducirá el tono y si le volvemos a dar clic este pausará este mismo.



Conclusión

Esta actividad fue un reto la verdad por que no sabía cómo crearía el diseño de esta misma ya que no tenía el conocimiento de las funcionalidades en Android Studio ya que en la anterior materia que fue la de desarrollo de aplicaciones 1 solo maneje lo que fue el diseño esta aplicación como digo fue todo un reto para mi ya que el código tiene que ir bien y funcionando adecuadamente gracias al tutor por explicarnos paso a paso y el detalle de las funcionalidades de Android Studio ya que sus tutorías fueron de gran ayuda a mi aprendizaje no importo que nos lleváramos un poco más de tiempo en estas mismas ya que nos ayudo a clarificar algunas de las funcionalidades y para qué sirven ya con esto me voy mas con la idea de cómo crear una aplicación móvil ya que no sabia en su totalidad como hacerla funcionar ya que muchos ejercicios no vienen las funciones y fue intuitivo de igual manera esta herramienta de Android ya que tiene muchas funciones y apartados que la hacen única de verdad.

Referencias

<https://github.com/Ricardorivas94cr/Parte-1-Ringtones-AndridStudio>