

Actividad | 2# | Instalación

Android Studio y Diseño de Aplicación(Parte2)

Desarrollo de Aplicaciones Móviles II

Ingeniería en Desarrollo de Software



TUTOR: Eduardo Israel Castillo García

ALUMNO: Casandra Montserrat Ortiz Cortes

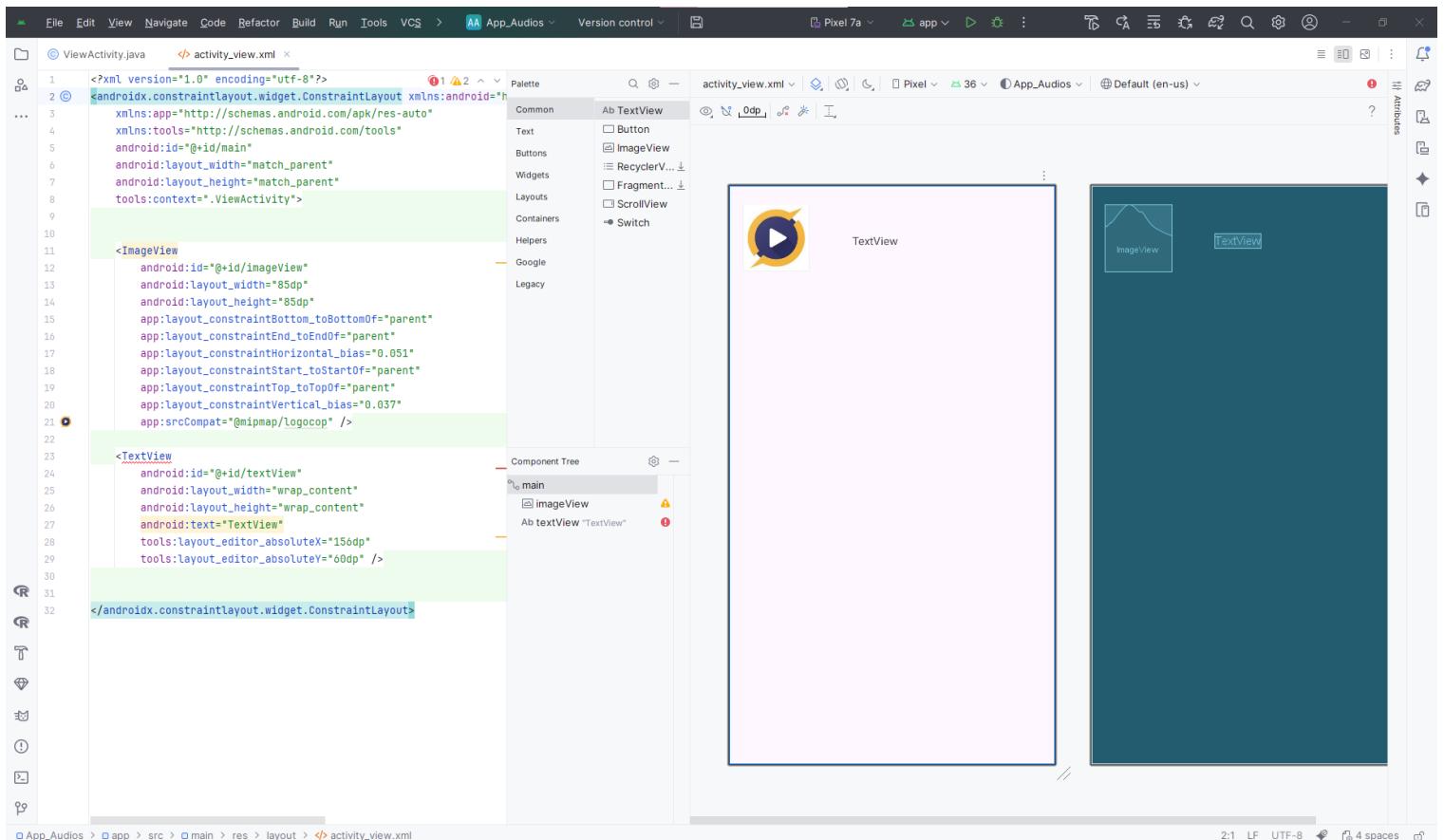
FECHA: 28/11/2025



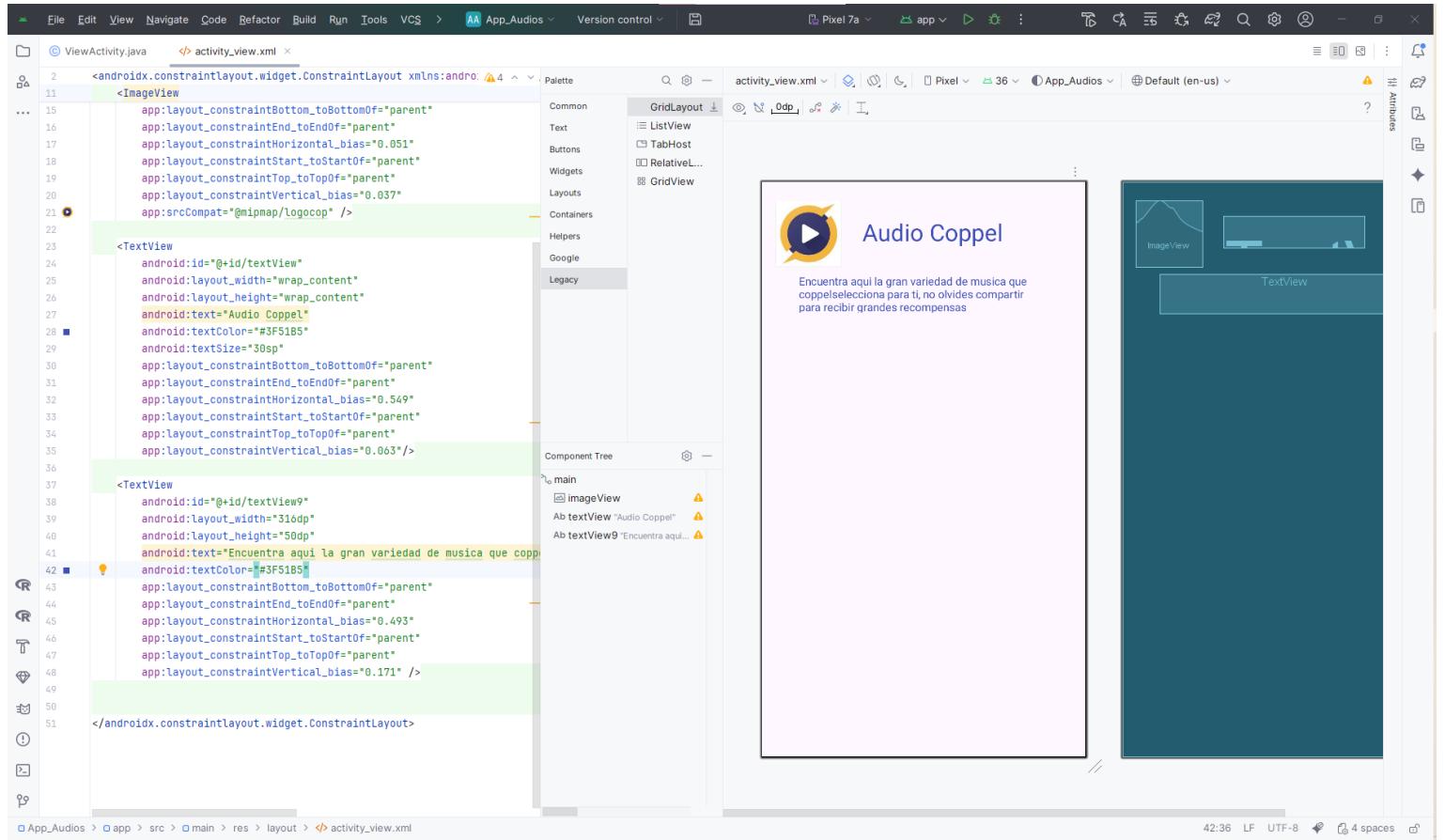
Índice

Desarrollo.....	5
○ Interfaz	
○ Codificación	
○ Prueba de la aplicación	
Conclusión.....	6
Referencia	

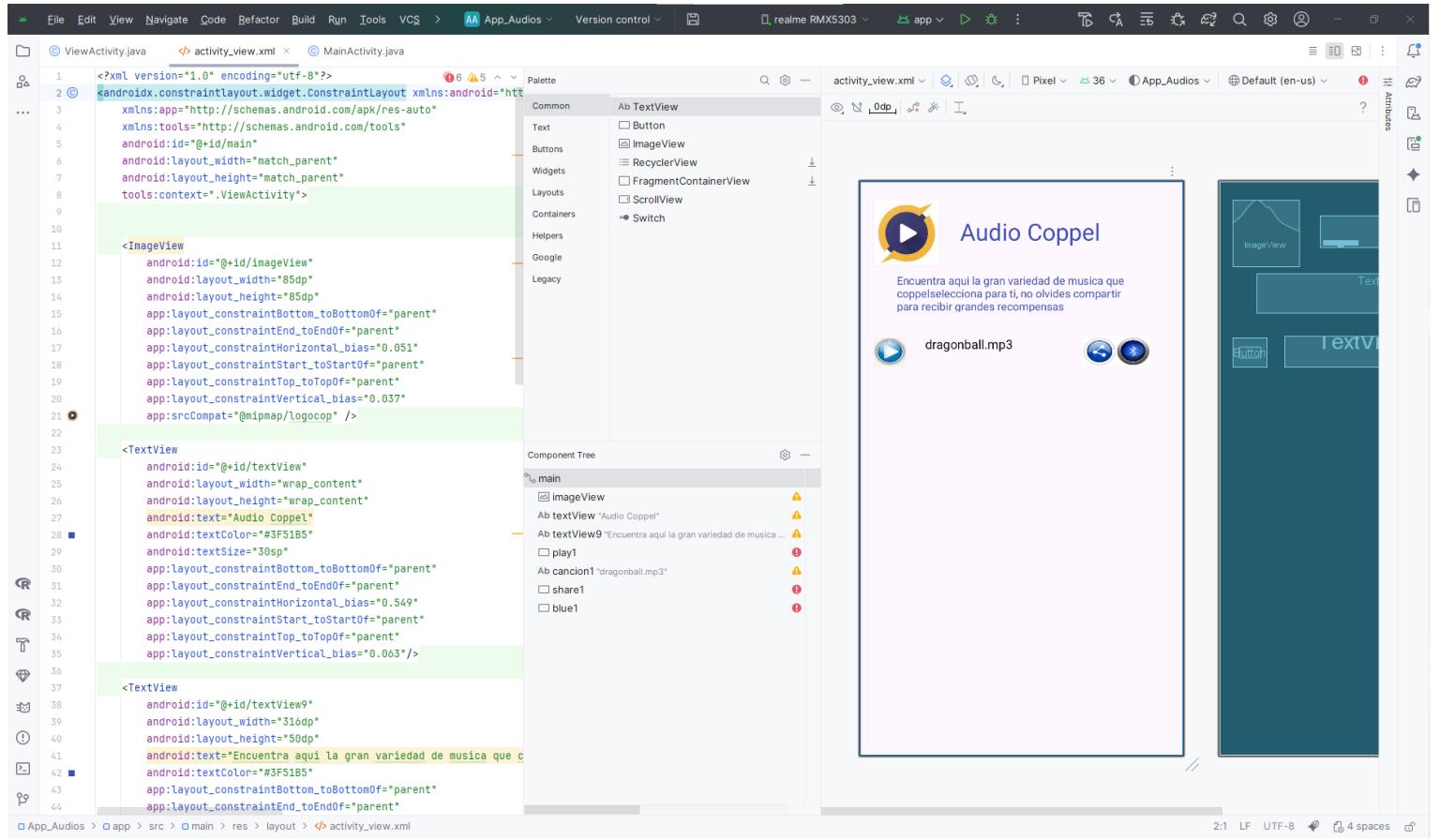
○ Interfaz



Lado Derecho: puse la imagen que llevaría y el texto que llevara de “Audio Coppel” donde el lado izquierdo nos muestra los códigos que se van agregando y sus funciones que están realizando.



Lado Derecho: Se muestra otro detalle en la imagen de cómo va agregándose ahora un mensaje que se mostrara en la parte de abajo del Audio Coppel.



Lado Izquierdo: en esta parte comenzaremos agregar las primera canción y así haremos con los demás agregando y del lado izquierdo veremos los códigos de Button, Share, Blue con sus funciones y el nombre de cada música que serán (como Star Wars, Mario Bros, Titanic, etc.) y a cada una le agregué tres botones: uno de reproducir play. El share y el bluetoot, aun no tendrá una función todavía hasta la próxima actividad.

○ Codificación

The screenshot shows the Android Studio interface with the ViewActivity.java file open in the main editor. The code defines a class ViewActivity that extends AppCompatActivity. It contains 15 lines of Java code, including imports for various Android components like View, Button, EdgeToEdge, AppCompatActivity, Insets, ViewCompat, WindowInsetsCompat, MediaPlayer, and Toast. Below the code, there are two sections of annotations: '2 usages' and '4 usages'. The right side of the screen displays the 'Running Devices' panel, which includes a diagram of a physical device connected via USB and instructions for mirroring to a virtual device or connecting over WiFi. The bottom status bar shows the current time as 17:45, the file type as LF, the encoding as UTF-8, and the code style as 4 spaces.

```
package com.example.app_audios;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.media.MediaPlayer;
import android.widget.Toast;

public class ViewActivity extends AppCompatActivity {

    2 usages ⚡ Rename usages
    Button play1, play2, play3, play4, play5, play6, play7, play8, play9, play10, share1, share2, share3, share4, share5, share6, share7, share8, share9, share10;

    4 usages
    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);
        share7 = (Button) findViewById(R.id.share7);
        share8 = (Button) findViewById(R.id.share8);
    }
}
```

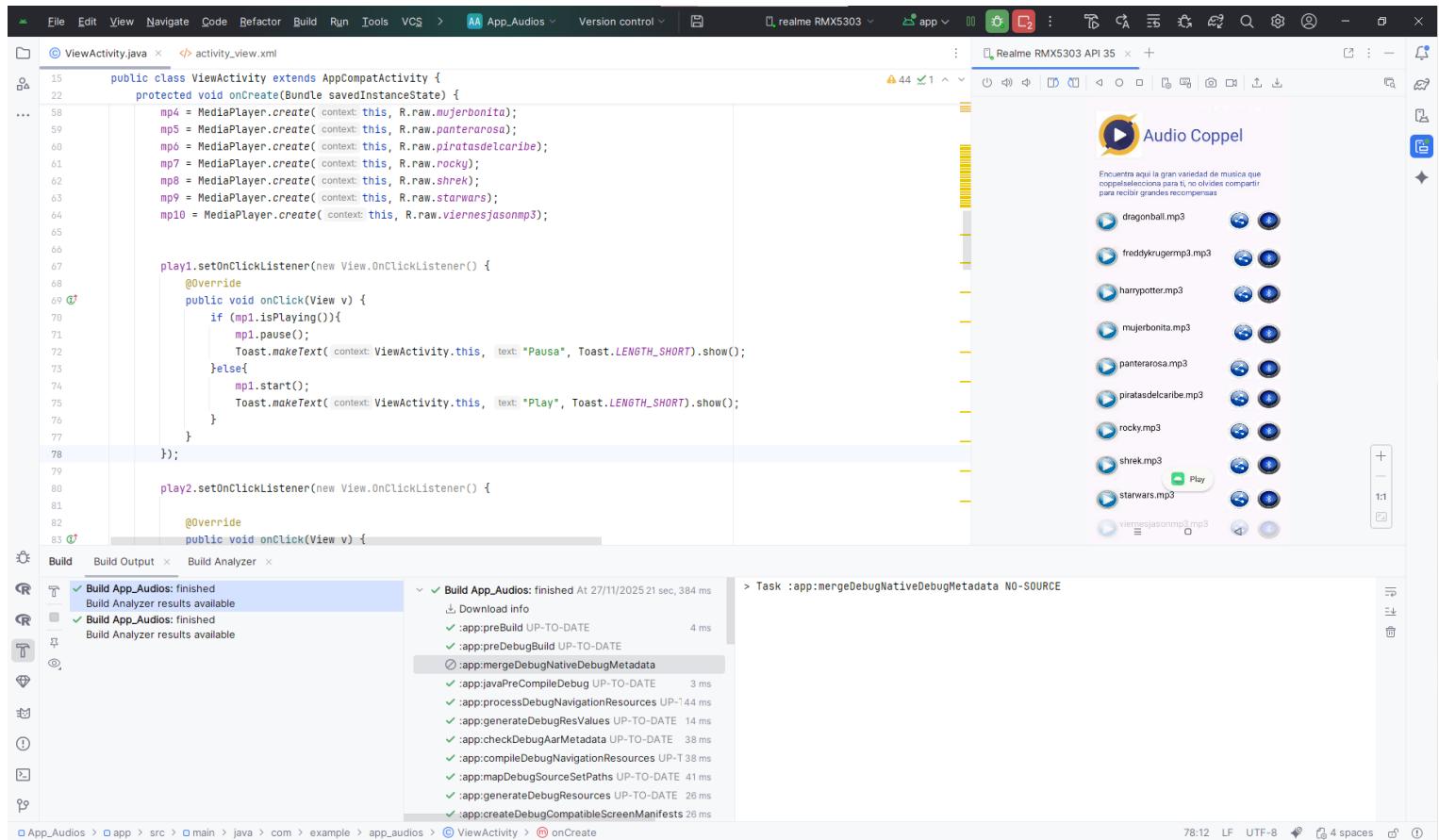
Tiene 10 botones para reproducir audios (DragonBall, Harry Potter, Star Wars, etc.).

The screenshot shows the Android Studio interface with the code editor open to ViewActivity.java. The code implements a MediaPlayer for each of three buttons (play1, play2, play3) to handle play/pause functionality and show a Toast message indicating the state.

```
15     public class ViewActivity extends AppCompatActivity {
16         protected void onCreate(Bundle savedInstanceState) {
17             ...
18             mp7 = MediaPlayer.create(context: this, R.raw.rocky);
19             mp8 = MediaPlayer.create(context: this, R.raw.shrek);
20             mp9 = MediaPlayer.create(context: this, R.raw.starwars);
21             mp10 = MediaPlayer.create(context: this, R.raw.viernesjasonmp3);
22 
23             ...
24 
25             play1.setOnClickListener(new View.OnClickListener() {
26 
27                 @Override
28                 public void onClick(View v) {
29                     if (mp1.isLooping()) {
30                         mp1.pause();
31                         Toast.makeText(context: ViewActivity.this, text: "Pausa", Toast.LENGTH_SHORT).show();
32                     } else {
33                         mp1.start();
34                         Toast.makeText(context: ViewActivity.this, text: "Play", Toast.LENGTH_SHORT).show();
35                     }
36                 }
37             });
38 
39             play2.setOnClickListener(new View.OnClickListener() {
40 
41                 @Override
42                 public void onClick(View v) {
43                     if (mp2.isLooping()) {
44                         mp2.pause();
45                         Toast.makeText(context: ViewActivity.this, text: "Pausa", Toast.LENGTH_SHORT).show();
46                     } else {
47                         mp2.start();
48                         Toast.makeText(context: ViewActivity.this, text: "Play", Toast.LENGTH_SHORT).show();
49                     }
50                 }
51             });
52 
53             play3.setOnClickListener(new View.OnClickListener() {
54 
55                 @Override
56                 public void onClick(View v) {
57                     if (mp3.isLooping()) {
58                         mp3.pause();
59                         Toast.makeText(context: ViewActivity.this, text: "Pausa", Toast.LENGTH_SHORT).show();
60                     } else {
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            ...
103        }
104    }
105 }
```

Cada botón hace play/pause con MediaPlayer y muestra un Toast que dice "Play" o "Pausa".

○ Prueba de la aplicación



Se da el comienzo de entra al Run con la conexión de nuestro celular al android studio, para verificar su funcionalidad, esperando unos minutos que se conecten y al hacerlo se puede apreciar en la imagen los detalles de la reproducción de la música donde vemos el mensaje de “Play” reproduciendo con éxito sin mostrar ningún error.



Audio Coppel

Encuentra aqui la gran variedad de musica que coppelselecciona para ti, no olvides compartir para recibir grandes recompensas



dragonball.mp3



freddykrugermp3.mp3



harrypotter.mp3



mujerbonita.mp3



panterarosa.mp3



piratasdelcaribe.mp3



rocky.mp3



shrek.mp3



starwars.r



Pausa



viernesjasonmp3.mp3



Donde podemos apreciar que al darle al botón play se “pausa” la reproducción de la música.



Conclusion

Realiza renderizados de layouts en tiempo real, y cuenta con la posibilidad de utilizar parámetros tools. Ejecuta la aplicación en tiempo real desde el propio teléfono móvil. Su potente emulador ayuda a comprobar el estado de la aplicación en el momento, sin necesidad de un ordenador.

Permite simular diferentes dispositivos y tabletas, pudiendo visualizarlas en un mismo entorno. De esta forma podemos trabajar varias aplicaciones simultáneamente y ver las partes de código necesarias de cada una.

Podrás crear elementos gráficos para la interfaz de tu aplicación sin necesidad de utilizar el código. Esta función te ayudará a diseñar el aspecto visual de tu App de una forma muy sencilla. Es el IDE oficial de Android. Esto asegura el correcto funcionamiento del software, ya que es el mismo que utilizan los propios desarrolladores del sistema operativo oficial de Android para crear todas sus aplicaciones. Sin embargo, también podrás encontrar algunas desventajas.

REFERENCIA:

Nielfa, J. S. (2021, 14 diciembre). ▷ **Android Studio el entorno de desarrollo oficial de Android.** Scoreapps. <https://scoreapps.com/blog/android-studio/#:~:text=Ventajas%20de%20Android%20Studio,-Adem%C3%A9s%20de%20las&text=Realiza%20renderizados%20de%20layouts%20en,sin%20necesidad%20de%20un%20ordenador.>

Link Github Zip Audio:

https://github.com/cassandraortiz31/actividadesCasandranuevo/blob/7edd3185e1baddded0191380768134a1c97f49cf/Cassandra%20Ortiza_act2_DAM2.zip