

Actividad [1] - [Instalación Android Studio y Diseño de Aplicación]

[Desarrollo de Aplicaciones Móviles 2]

**Ingeniería en Desarrollo de
Software**

Tutor: Eduardo Israel Castillo García

Alumno: Carlos Francisco Estrada Salazar

Fecha: 04/Dic/2024

INDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
Interfaz	6
Codificación	7
Prueba de la aplicación	13
CONCLUSIÓN	14
REFERENCIAS	15

GitHub Link:

INTRODUCCIÓN

En el contexto actual, donde las herramientas digitales son esenciales para responder a las necesidades del mercado, el desarrollo de aplicaciones móviles se convierte en una habilidad indispensable para los ingenieros en software. En esta actividad, se trabajará en la instalación y configuración de Android Studio, un entorno de desarrollo integrado (IDE) ampliamente utilizado para la creación de aplicaciones Android. Este proceso inicial sentará las bases para diseñar una aplicación móvil funcional que permita manejar una galería de ringtones.

El escenario propuesto se enmarca en la temporada del Buen Fin, un evento comercial destacado que impulsa el desarrollo de soluciones tecnológicas innovadoras. La aplicación requerida debe incluir, al momento de su descarga, una colección predefinida de 10 audios en formato .mp3. Este proyecto no solo abarca aspectos técnicos como la instalación de software y la configuración del entorno de desarrollo, sino que también fomenta la creatividad y el diseño centrado en el usuario, al ofrecer un producto atractivo y funcional para el mercado.

A través de esta actividad, se busca fortalecer las competencias en desarrollo móvil, desde la preparación inicial del entorno hasta la implementación de funciones básicas de una aplicación. El objetivo es combinar habilidades técnicas y creativas para crear una solución acorde a las demandas actuales del sector tecnológico.

DESCRIPCIÓN

El contexto presentado sitúa al desarrollador de software en un escenario realista y oportuno, relacionado con el Buen Fin, un evento comercial de gran relevancia en el mercado mexicano. Este marco no solo incrementa la exigencia de crear soluciones rápidas y eficientes, sino que también pone de manifiesto la importancia de ofrecer productos tecnológicos atractivos que mejoren la experiencia del usuario final. La actividad asignada consiste en diseñar una aplicación móvil para el sistema operativo Android, con la funcionalidad específica de manejar una galería de ringtones, lo cual apunta a cubrir una necesidad puntual de personalización y entretenimiento para los usuarios.

La primera etapa de esta tarea implica instalar y configurar Android Studio, que es la herramienta principal para el desarrollo de aplicaciones Android. Este proceso es esencial, ya que asegura un entorno adecuado para programar, probar y depurar aplicaciones. Una vez completada la configuración, se requiere crear una aplicación que contenga 10 audios en formato .mp3 listos para su uso, lo que añade una dimensión práctica al ejercicio al introducir elementos de diseño y gestión de recursos multimedia.

Esta actividad combina habilidades técnicas y creativas, desafiando al desarrollador a dominar tanto el manejo de Android Studio como la implementación de funciones básicas que respondan a las expectativas del usuario. Además, subraya la relevancia de crear aplicaciones atractivas y funcionales en un tiempo limitado, un aspecto clave en la dinámica actual del mercado tecnológico.

JUSTIFICACIÓN

La implementación de una solución basada en una aplicación móvil para la actividad presentada es altamente relevante debido a varias razones técnicas, comerciales y sociales. En primer lugar, el uso de Android Studio como entorno de desarrollo es ideal, ya que es una herramienta ampliamente reconocida por su capacidad para crear aplicaciones nativas con un alto rendimiento y compatibilidad en dispositivos Android, que dominan el mercado global de teléfonos inteligentes.

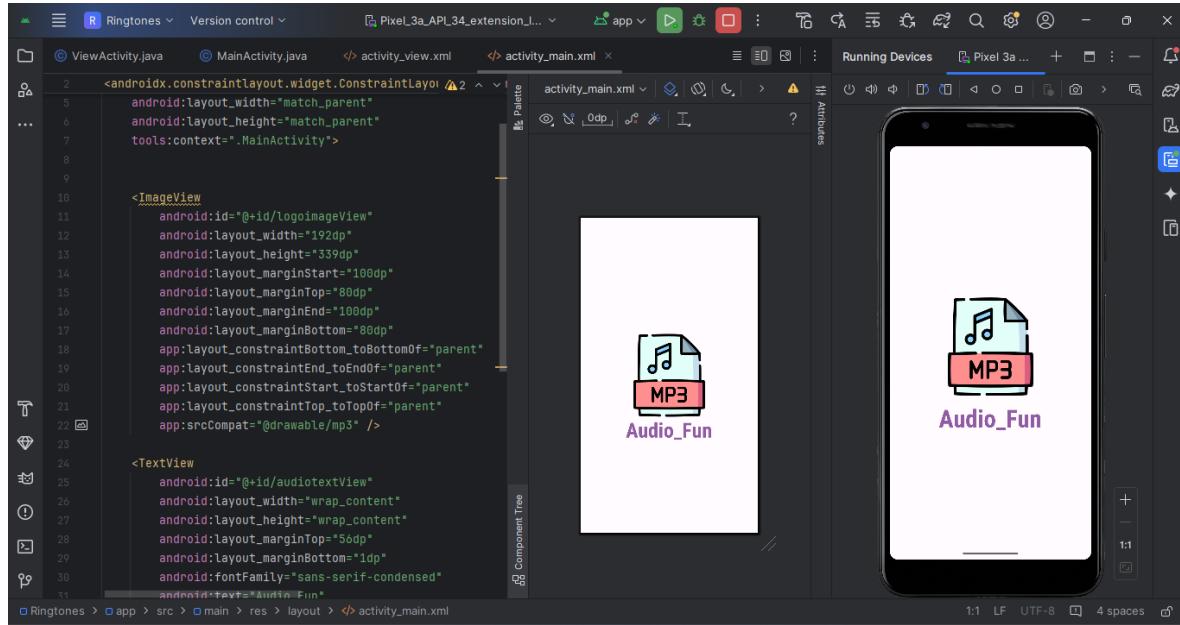
El enfoque de esta actividad se alinea con la necesidad actual de personalización y conveniencia que buscan los usuarios en sus dispositivos. Una galería de ringtones preinstalada con audios en formato .mp3 no solo facilita el acceso a contenido listo para usar, sino que también ofrece una experiencia de usuario atractiva y sencilla. Además, esta funcionalidad se presenta en un contexto comercial como el Buen Fin, lo que aumenta su potencial de aceptación y éxito al estar diseñada para un evento que promueve el consumo y la tecnología.

Por otro lado, esta solución fomenta el aprendizaje y la práctica de habilidades clave para los desarrolladores de software, como la configuración de entornos de desarrollo, la integración de recursos multimedia y el diseño de interfaces amigables. Estas competencias son esenciales en un entorno profesional competitivo, y este tipo de actividad permite adquirirlas de forma práctica y aplicada.

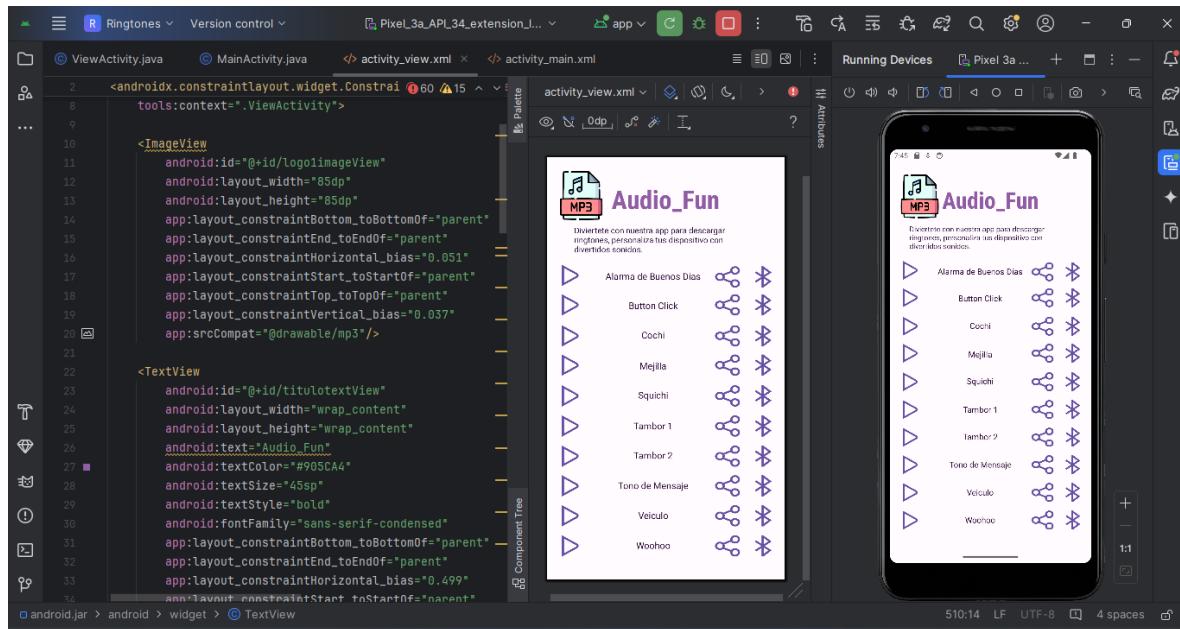
Finalmente, la elección de Android Studio y la creación de una aplicación funcional refuerzan la capacidad del ingeniero en desarrollo de software para responder de manera efectiva a las necesidades del mercado, al tiempo que mejora su experiencia técnica en un área de gran demanda.

DESARROLLO

Interfaz

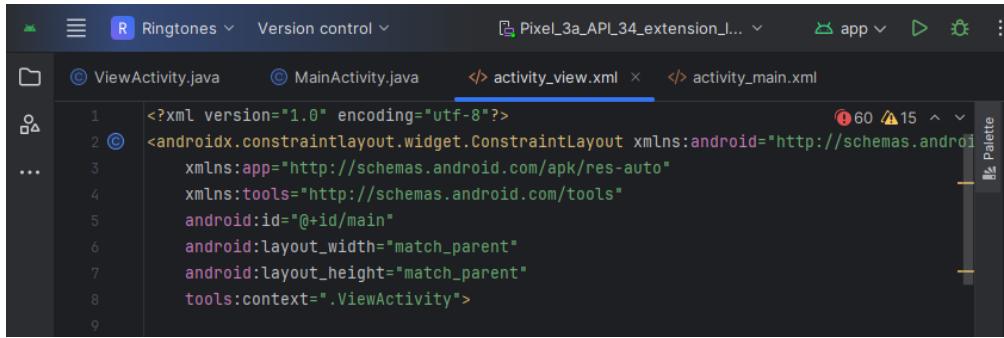


Creando la animación de inicio.



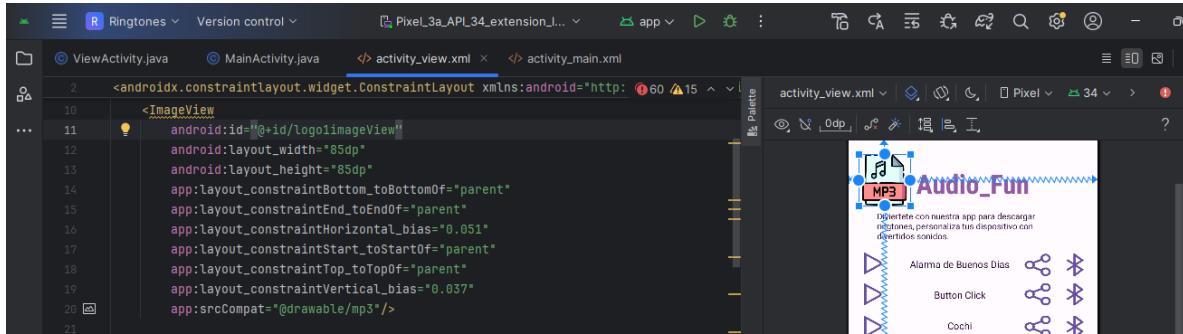
Creando la pantalla de selección de tono.

Codificación



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ViewActivity">
```

En la parte principal del archivo XML encontramos el contenedor principal de los elementos de la interfaz. Permite crear diseños responsivos estableciendo restricciones entre sus elementos hijos.



Un **ImageView** se utiliza para mostrar una imagen en la interfaz de usuario. Este código contiene varios atributos que configuran su apariencia y comportamiento.

- Este ImageView muestra una imagen de 85x85 dp.
- Está centrado en la parte superior izquierda del contenedor con un ligero desplazamiento horizontal y vertical, definido por los valores bias.
- La imagen que muestra proviene del archivo mp3 dentro de los recursos drawable.

```

1 package com.example.ringtones;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.os.Handler;
7 import android.view.WindowManager;
8 import android.view.animation.Animation;
9 import android.view.animation.AnimationUtils;
10 import android.widget.ImageView;
11 import android.widget.TextView;
12

```

En el **MainActivity.java**, en el primer bloque de código, se colocan todas las librerías que se utilizaran.

```

14 <> public class MainActivity extends AppCompatActivity {
15

```

MainActivity: Es la actividad principal de la aplicación. Define el comportamiento de la primera pantalla que se muestra al abrir la aplicación.

extends AppCompatActivity: Indica que la clase hereda de AppCompatActivity, que proporciona compatibilidad con versiones anteriores de Android y funcionalidad adicional para actividades.

```

15
16
17 @Override
18     protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

onCreate: Método que se ejecuta cuando la actividad se crea por primera vez. Aquí se inicializan elementos y se configura el diseño.

savedInstanceState: Permite restaurar el estado de la actividad si fue destruida previamente.

```

getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
setContentView(R.layout.activity_main);

```

Establece la actividad en pantalla completa, ocultando la barra de estado.

```

setContentView(R.layout.activity_main);

```

Vincula el archivo de diseño XML activity_main.xml como la interfaz de usuario para esta actividad.

```
//Añadir animación
Animation animation1 = AnimationUtils.loadAnimation( context: this,R.anim.aparecer_desplasar);
Animation animation2 = AnimationUtils.loadAnimation( context: this,R.anim.aparecer_desplasar);
```

AnimationUtils.loadAnimation: Carga una animación desde el archivo XML en la carpeta res/anim. Aquí se carga una animación llamada aparecer_desplasar.

```
TextView audiotextView = findViewById(R.id.audiotextView);
ImageView logoimageVideo = findViewById(R.id.logoimageView);
```

findViewById: Busca las vistas definidas en el diseño por su identificador.

audiotextView: Un TextView que muestra texto.

logoimageVideo: Un ImageView que muestra una imagen.

```
audiotextView.setAnimation(animation1);
logoimageVideo.setAnimation(animation2);
```

Asigna las animaciones cargadas a las vistas. Esto hará que las vistas se animen según la definición en aparecer_desplasar.

```
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {
        Intent intent = new Intent( packageContext: MainActivity.this, ViewActivity.class);
        startActivity(intent);
        finish();
    }
}, delayMillis: 6000);
```

Handler().postDelayed: Ejecuta un bloque de código después de un retraso especificado (en milisegundos).

Runnable: Contiene el código que se ejecutará después del retraso.

6000: Especifica un retraso de 6 segundos.

Acciones dentro de Runnable:

Crea un Intent para iniciar una nueva actividad (ViewActivity).

startActivity: Inicia la nueva actividad.

finish: Finaliza la actividad actual (MainActivity).

```

14 <> public class MainActivity extends AppCompatActivity {
15
16     ...
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
21         setContentView(R.layout.activity_main);
22
23         //Añadir animación
24         Animation animation1 = AnimationUtils.loadAnimation(this,R.anim.aparecer_desplazar);
25         Animation animation2 = AnimationUtils.loadAnimation(this,R.anim.aparecer_desplazar);
26
27         audiotextView = findViewById(R.id.audiotextView);
28         logoimageVideo = findViewById(R.id.logoimageView);
29
30         audiotextView.setAnimation(animation1);
31         logoimageVideo.setAnimation(animation2);
32
33         new Handler().postDelayed(new Runnable() {
34             ...
35             @Override
36             public void run() {
37                 Intent intent = new Intent(getApplicationContext(), ViewActivity.class);
38                 startActivity(intent);
39                 finish();
39             }
38         }, 6000);
39     }
40 }

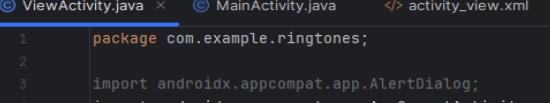
```

Ringtones > app > src > main > java > com > example > ringtones > MainActivity > onCreate > anonymous Runnable > run

FLUJO GENERAL:

1. La actividad se muestra en pantalla completa.
2. Asigna un diseño con elementos animados (TextView y ImageView).
3. Las animaciones se reproducen.
4. Despues de 6 segundos, se inicia la actividad ViewActivity y se cierra MainActivity.

Este código muestra animaciones breves antes de redirigir al usuario a la siguiente pantalla de la aplicación.

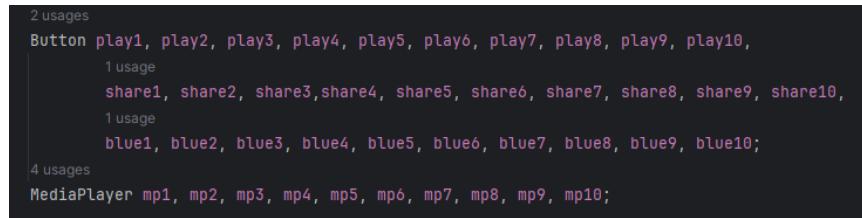


The screenshot shows the Android Studio interface with the following details:

- Project Bar:** Shows "Ringtones" as the selected project.
- Toolbars:** Version control and Pixel_3a_API_34_extensions.
- File List:** Shows "ViewActivity.java", "MainActivity.java", "activity_view.xml", and "activity_view.xml" again.
- Code Editor:** Displays the Java code for ViewActivity.java. The code includes imports for AlertDialog, AppCompatActivity, DialogInterface, Intent, MediaPlayer, Bundle, View, Button, and Toast from androidx.appcompat.app and android.* packages.

```
1 package com.example.ringtones;
2
3 import androidx.appcompat.app.AlertDialog;
4 import androidx.appcompat.app.AppCompatActivity;
5 import android.content.DialogInterface;
6 import android.content.Intent;
7 import android.media.MediaPlayer;
8 import android.os.Bundle;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.Toast;
```

En el **ViewActivity.java**, en el bloque principal del código se agregan las librerías necesarias para el funcionamiento.



Se crean los objetos necesarios para el funcionamiento.

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Ringtones > app > src > main > java > com > example > ringtones > ViewActivity.java
- Code Editor:** The code for the `onCreate()` method of `ViewActivity` is displayed. The code initializes various buttons and share buttons by finding them by ID.

```
public class ViewActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        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);
        share9=(Button)findViewById(R.id.share9);
        share10=(Button)findViewById(R.id.share10);

        blue1=(Button)findViewById(R.id.blue1);
        blue2=(Button)findViewById(R.id.blue2);
        blue3=(Button)findViewById(R.id.blue3);
    }
}
```

- Toolbars and Status Bar:** The top bar shows tabs for ViewActivity.java, MainActivity.java, activity_view.xml, and activity_main.xml. The status bar at the bottom shows the time as 57:40, battery level as LF, and system status as UTF-8.

findViewById: Busca y vincula cada botón definido en el archivo de diseño XML mediante sus IDs.

```

    mp1=MediaPlayer.create( context: this,R.raw.alarma_good_morning);
    mp2=MediaPlayer.create( context: this,R.raw.button_click);
    mp3=MediaPlayer.create( context: this,R.raw.cochi);
    mp4=MediaPlayer.create( context: this,R.raw.mejilla_2);
    mp5=MediaPlayer.create( context: this,R.raw.squichi);
    mp6=MediaPlayer.create( context: this,R.raw.tambor_1);
    mp7=MediaPlayer.create( context: this,R.raw.tambor_2);
    mp8=MediaPlayer.create( context: this,R.raw.tono_mensaje);
    mp9=MediaPlayer.create( context: this,R.raw.vehicle006);
    mp10=MediaPlayer.create( context: this,R.raw.woohoo);

```

MediaPlayer.create: Inicializa un objeto MediaPlayer para reproducir archivos de audio.

Cada **mpX** (por ejemplo, mp1, mp2, etc.) está asociado con un archivo de audio ubicado en la carpeta **res/raw** del proyecto. Los nombres de los archivos indican su propósito, como tonos de alarma, clics de botones, efectos de tambor, etc002E

```

play1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mp1.isPlaying()) {
            mp1.pause();
            Toast.makeText( context: ViewActivity.this, text: "Pause",Toast.LENGTH_SHORT).show();
        }else{
            mp1.start();
            Toast.makeText( context: ViewActivity.this, text: "Play", Toast.LENGTH_SHORT).show();
        }
    }
});

```

setOnClickListener: Vincula una acción al evento de clic para cada botón.

Comportamiento:

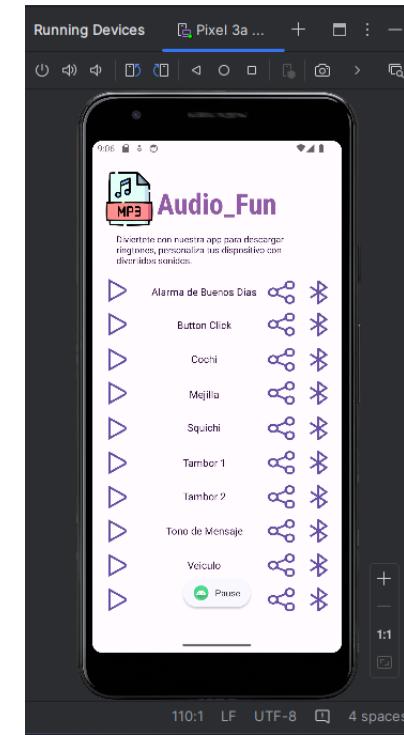
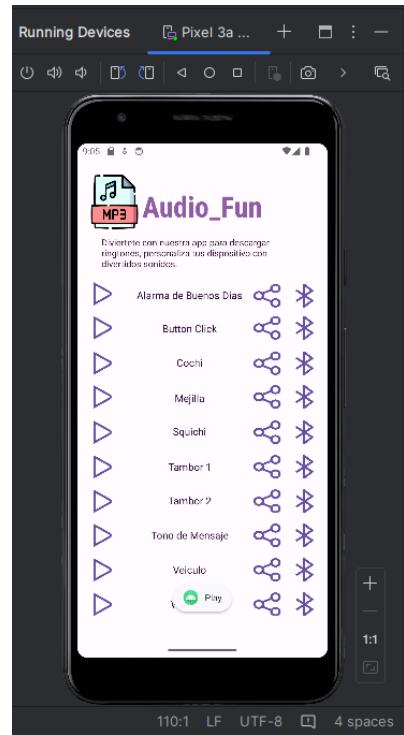
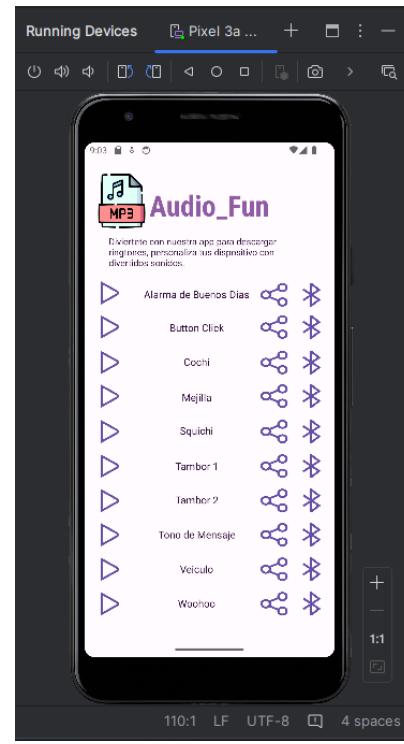
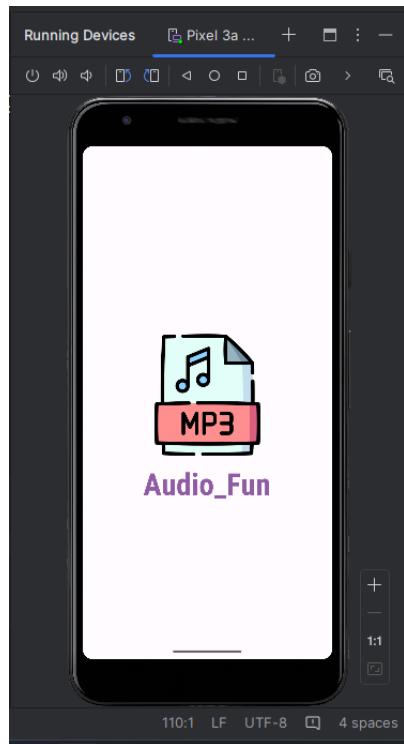
- Si el audio correspondiente está reproduciéndose (isPlaying()), se pausa con pause().
- Si no se está reproduciendo, se inicia con start().
- Se muestra un mensaje emergente (Toast) indicando el estado actual: "Play" o "Pause".

Este patrón se repite para todos los botones play (play1 a play10), cada uno controlando su respectivo archivo de audio.

El objetivo principal del código es permitir a los usuarios interactuar con diferentes clips de audio:

1. **Reproducir/Pausar audios:** Cada botón play controla un archivo de audio específico.
2. **Notificación visual:** Se utiliza un Toast para notificar al usuario sobre el estado de reproducción.

Prueba de Aplicación



CONCLUSIÓN

La actividad de instalación y configuración de Android Studio, junto con el diseño de una aplicación para manejar una galería de ringtones, representa un ejercicio fundamental para fortalecer habilidades clave en el desarrollo de software. En el ámbito laboral, este tipo de práctica permite a los ingenieros de software enfrentar desafíos reales como el diseño de aplicaciones funcionales que cumplan con las necesidades específicas de los usuarios, en este caso, proporcionar una herramienta útil para personalizar sus dispositivos móviles durante una temporada comercial como el Buen Fin.

Desde una perspectiva técnica, la instalación de Android Studio no solo facilita el desarrollo de aplicaciones móviles, sino que también proporciona un entorno robusto que integra herramientas esenciales para la programación, depuración y diseño visual de interfaces de usuario. Asimismo, la creación de una aplicación con funcionalidades específicas, como incluir archivos .mp3 preinstalados, ayuda a comprender cómo manejar recursos dentro del sistema Android, lo cual es una competencia crucial en el desarrollo de software móvil.

En la vida cotidiana, actividades como esta fomentan la capacidad de resolver problemas y de adaptarse a nuevas herramientas tecnológicas. Además, ayudan a entender el impacto práctico de las aplicaciones móviles en el día a día de las personas, promoviendo el desarrollo de productos útiles y atractivos.

REFERENCIAS

MoureDev by Brais Moure. (2020, 17 enero). *ANDROID STUDIO: COMO Crear una APP (para Principiantes)*  [Tutorial] [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=BQaxPwZWboA>