

Actividad [#2] – [Instalación Android Studio y Diseño de Aplicación] [“Desarrollo de Aplicaciones Móviles II”]

Ingeniería en Desarrollo de Software

**Tutor: Bárbara Verónica Buendía Hernández**

**Alumno: Manuel Enrique Ramirez Lopez**

**Fecha: 18/07/2022**

***Indice***

*[Investigación](#_Toc12472)* [1](#_Toc12472)

*[Interfaz](#_Toc11738)* [2](#_Toc11738)

*[Conclusion](#_Toc2737)* [10](#_Toc2737)

# *Investigación*

***¿Qué es Android Studio?***

Es una herramienta de trabajo enfocada al desarrollo de aplicaciones móviles, maneja distintos tipos de paquetes para una mejor visualización y distintos tipos de lenguajes de programación, gracias a su modularidad permite tener compatibilidad con las diversas versiones de sistemas operativos.

Android Studio permite una sólida integración de características y funciones con aplicaciones que mejoran con el tiempo. De esta manera tenemos lo siguiente:  
  
-El sistema de compilación es flexible, pero compatible con Gradle, lo que permite una automatización de compilación flexible y de alto rendimiento. Groovy y Kotlin DSL son dos lenguajes que se utilizan para construir scripts.

-El propósito de este entorno es permitir a los usuarios operar sin problemas con una gran cantidad de funciones prácticas y útiles.

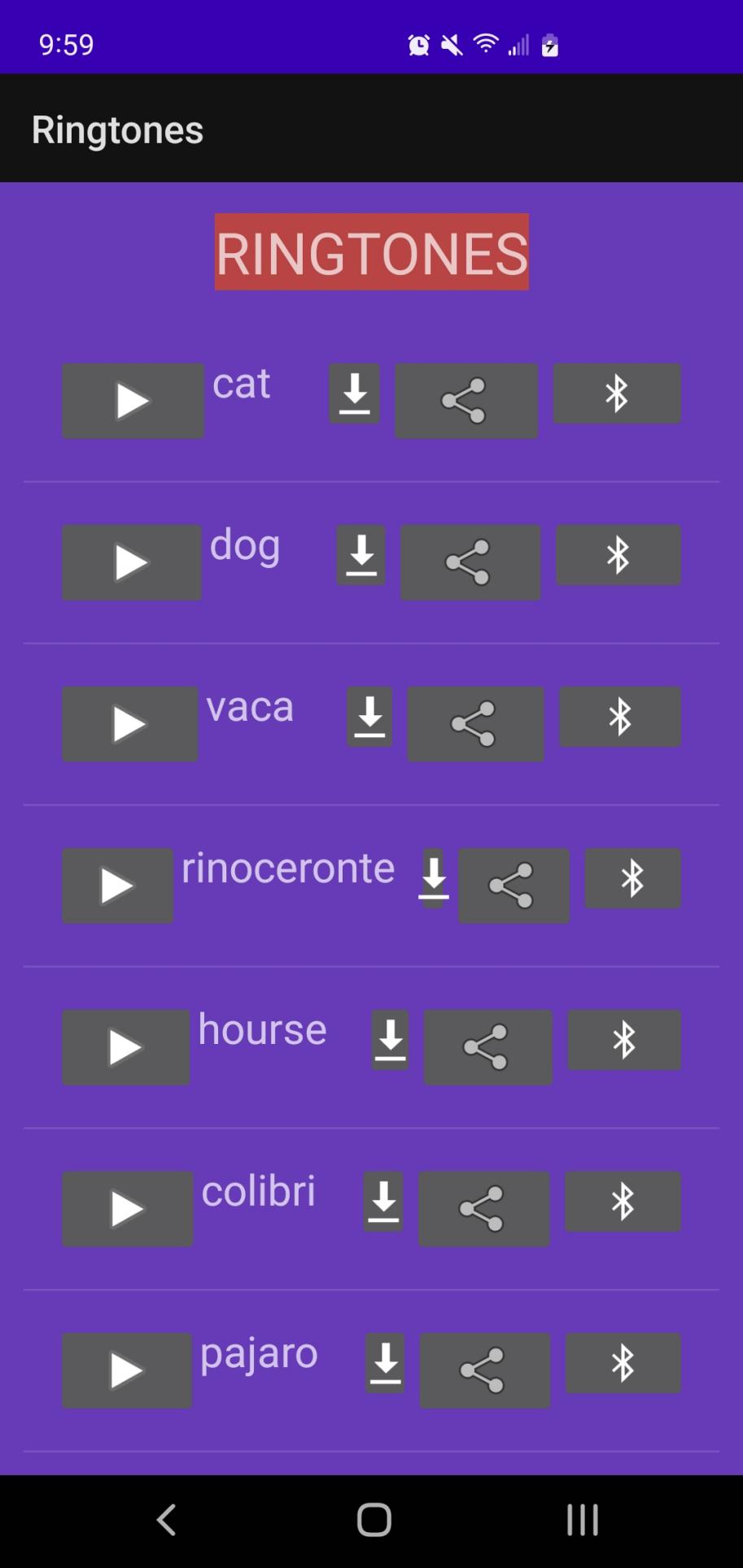
-Esta plataforma le permite desarrollar aplicaciones para cualquier dispositivo Android. Contiene patrones de compilación que lo ayudan a introducir la funcionalidad común de otras aplicaciones más rápido, así como a importar código de muestra. Aumentar el número de herramientas de prueba con marcos. Libere el código y los recursos de la aplicación sin reiniciar la aplicación. Brinda compatibilidad con servicios en la nube como Google Cloud Platform. Admite lenguajes como NDK y C.

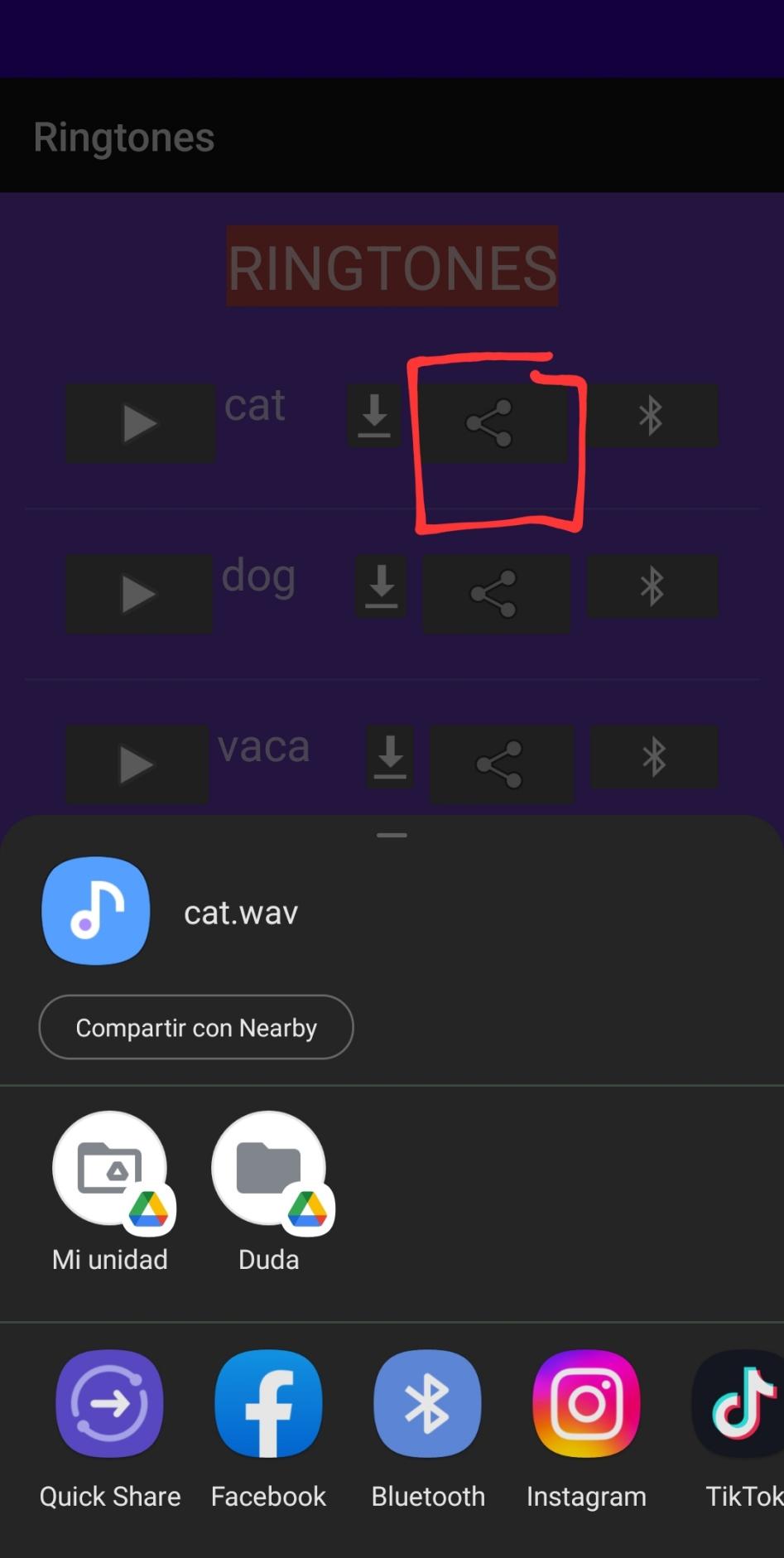
***¿Qué es un archivo .wav?***

Todo sobre archivos WAV

WAV es un formato de contenedor sin pérdida que almacena archivos de audio sin comprimir e incluye el formato de audio PCM. Como resultado, un documento de formato WAV proporciona una calidad de sonido óptima, a pesar de que consume el mayor espacio de almacenamiento en comparación con otros formatos de audio. Para enviar a través de Internet y para transmisión, dicho archivo no es adecuado. Fue desarrollado por IBM y Microsoft y se considera el formato de audio estándar para PC con Windows. Cuanto más fina sea la resolución al mostrar la señal, mejor será la calidad del sonido. Para un sonido de primera clase, se recomienda una velocidad de bits de 24 bits como configuración, 16 bits corresponden a la calidad del CD.

# *Interfaz*

******

**Botón share**

**Botón Bluetooth**

*Codificación*

***Main activity***

package mx.umi  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.ListView  
  
class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val ringtonesList = findViewById<ListView>(R.id.*ringtonesList*)  
  
 val listaRingtones = *listOf*(Ringtone("cat", "colibri"), Ringtone("dog", "gallo"), Ringtone("vaca", "jirafa"),Ringtone("rinoceronte", "pajaro"),Ringtone("hourse", "gallo"),Ringtone("colibri", "gallo"),Ringtone("pajaro", "gallo"),Ringtone("iguana", "gallo"),Ringtone("gallo", "gallo"),Ringtone("jirafa", "gallo"))  
  
 val adapter = RingtoneAdapter(this, listaRingtones)  
  
 ringtonesList.*adapter* = adapter  
 }  
}

***Ringtone***

package mx.umi  
  
class Ringtone (  
 var ringtoneName : String,  
 var ringtoneFileName: String  
)

***Activity main xml***

# 

<?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"  
 android:background="#673AB7"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginBottom="26dp"  
 android:background="#1B3E98"  
 android:backgroundTint="#B84444"  
 android:text="RINGTONES"  
 android:textSize="30dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/ringtonesList"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ListView  
 android:id="@+id/ringtonesList"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="12dp"  
 android:layout\_marginEnd="12dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Ringtone adapter**

package mx.umi  
  
import android.content.Context  
import android.content.Intent  
import android.media.MediaPlayer  
import android.net.Uri  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ArrayAdapter  
import android.widget.ImageButton  
import android.widget.TextView  
import android.widget.Toast  
import androidx.core.content.ContextCompat.startActivity  
import androidx.core.content.FileProvider  
import java.io.\*  
  
class RingtoneAdapter(private val mContext: Context, private val listaRingtones: List<Ringtone>) : ArrayAdapter<Ringtone>(mContext, 0, listaRingtones) {  
  
 override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {  
 val layout = LayoutInflater.from(mContext).inflate(R.layout.*item\_ringtone*, parent, false)  
 val ringtone = listaRingtones[position]  
 layout.findViewById<TextView>(R.id.*ringtoneName*).*text* = ringtone.ringtoneName  
  
 layout.findViewById<ImageButton>(R.id.*play\_btn*).setOnClickListener**{** //make a audio file in the res/raw directory  
 if(ringtone.ringtoneFileName == "dog"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*dog*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "cat"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "gallo"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "colibri"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "rinoceronte"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "vaca"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "pajaro"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "iguana"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "hourse"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }else if(ringtone.ringtoneFileName== "jirafa"){  
 val mediaPlayer = MediaPlayer.create(*context*, R.raw.*cat*)  
 mediaPlayer?.start()  
 }  
  
 **}** layout.findViewById<ImageButton>(R.id.*download\_btn*).setOnClickListener**{** Toast.makeText(*context*, "You clicked me download.", Toast.*LENGTH\_SHORT*).show()  
 **}** layout.findViewById<ImageButton>(R.id.*shared\_btn*).setOnClickListener**{** val uri = getAudioToShare(R.raw.*cat*)  
 val intent = Intent(Intent.*ACTION\_SEND*)  
 intent.*type* = "\*/\*"  
 intent.putExtra(Intent.*EXTRA\_STREAM*, uri)  
 intent.putExtra(Intent.*EXTRA\_TEXT*, "Te comparto el siguiente audio")  
 intent.putExtra(Intent.*EXTRA\_SUBJECT*, "Comparte un audio")  
 intent.*flags* = Intent.*FLAG\_GRANT\_READ\_URI\_PERMISSION* startActivity(*context*, Intent.createChooser(intent, "Compartir sonido"), null)  
 **}** layout.findViewById<ImageButton>(R.id.*bluetooth*).setOnClickListener**{** val uri = getAudioToShare(R.raw.*cat*)  
 val intent = Intent(Intent.*ACTION\_SEND*)  
 intent.*type* = "\*/\*"  
 intent.putExtra(Intent.*EXTRA\_STREAM*, uri)  
 intent.putExtra(Intent.*EXTRA\_TEXT*, "Te comparto el siguiente audio a traves de bluetooth")  
 intent.putExtra(Intent.*EXTRA\_SUBJECT*, "Comparte un audio")  
 intent.*flags* = Intent.*FLAG\_GRANT\_READ\_URI\_PERMISSION* startActivity(*context*, Intent.createChooser(intent, "Compartir sonido"), null)  
 **}** return layout  
 }  
  
  
 private fun getAudioToShare(resSoundId: Int): Uri? {  
 var uri: Uri? = null  
 var buffer: ByteArray? = null  
 val fIn: InputStream = *context*.getResources().openRawResource(resSoundId)  
 var size = 0  
 try {  
 size = fIn.available()  
 buffer = ByteArray(size)  
 fIn.read(buffer)  
 fIn.close()  
 } catch (e: IOException) {  
 // *TODO Auto-generated catch block* return null  
 }  
 val path = "/data/data/mx.umi/cache/audios/"  
 val filename = "cat.wav"  
 val f = File(path)  
 val exists = f.exists()  
 if (!exists) {  
 f.mkdirs()  
 }  
 val save: FileOutputStream  
 val fileSaved = File(path, filename)  
 try {  
 save = FileOutputStream(fileSaved)  
 save.write(buffer)  
 save.flush()  
 save.close()  
 } catch (e: FileNotFoundException) {  
 // *TODO Auto-generated catch block* Toast.makeText(  
 *context*,  
 e.message,  
 Toast.*LENGTH\_LONG* ).show()  
 return null  
 } catch (e: IOException) {  
 // *TODO Auto-generated catch block* Toast.makeText(  
 *context*,  
 e.message,  
 Toast.*LENGTH\_LONG* ).show()  
 return null  
 }  
 //uri = FileProvider.getUriForFile(this, "mx.edu.sendingfiles.provider", fileSaved);  
 uri = FileProvider.getUriForFile(*context*, "mx.umi.provider", fileSaved)  
 return uri  
 }  
  
  
  
}

**AndroidManifest**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="mx.umi">  
  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Ringtones">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <provider  
 android:name="androidx.core.content.FileProvider"  
 android:authorities="${applicationId}.provider"  
 android:exported="false"  
 android:grantUriPermissions="true">  
 <meta-data  
 android:name="android.support.FILE\_PROVIDER\_PATHS"  
 android:resource="@xml/paths" />  
 </provider>  
  
 </application>  
  
</manifest>

**Paths**

<?xml version="1.0" encoding="utf-8"?>  
<paths>  
 <cache-path name="shared\_images" path="images/" />  
 <external-path name="external\_files" path="."/>  
 <external-files-path name="external\_files\_path" path="." />  
 <root-path name="root" path="." />`  
</paths>

**Item ringtone**

<?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="wrap\_content"  
 android:layout\_margin="10dp">  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="16dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginEnd="16dp"  
 android:backgroundTint="@color/black"  
 android:orientation="horizontal"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <ImageButton  
 android:id="@+id/play\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/ic\_media\_play" />  
  
 <TextView  
 android:id="@+id/ringtoneName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Ringtone 1"  
 android:textSize="20sp" />  
  
 <ImageButton  
 android:id="@+id/download\_btn"  
 android:layout\_width="8dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/stat\_sys\_download" />  
  
 <ImageButton  
 android:id="@+id/shared\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/ic\_menu\_share" />  
  
 <ImageButton  
 android:id="@+id/bluetooth"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 app:srcCompat="@android:drawable/stat\_sys\_data\_bluetooth" />  
  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

# *Conclusion*

*Un aspecto fundamental para la elaboracion de “x” proyecto es la comprensión de que se quiere hacer ya que de ser mal interpretado puede dar doble trabajo para llegar a un fin en común o afectar el tiempo de entrega del proyecto, otro aspecto importante es dar “vista” a la app para hacerla visualmente mas llamativa ya que por mejor que funcione u ofrezca características fundamentales para lo que se planea que funcione si la interfaz maneja mal su paleta de colores o tipografía, para el usuario le resultara mas sencillo buscar otra alternativa.*