

Actividad [#3] – [Diseño de Aplicación (Parte final) ]

[“Desarrollo de Aplicaciones Móviles II”]

Ingeniería en Desarrollo de Software

**Tutor: Bárbara Verónica Buendía Hernández**

**Alumno: Manuel Enrique Ramirez Lopez**

**Fecha: 26/07/2022**

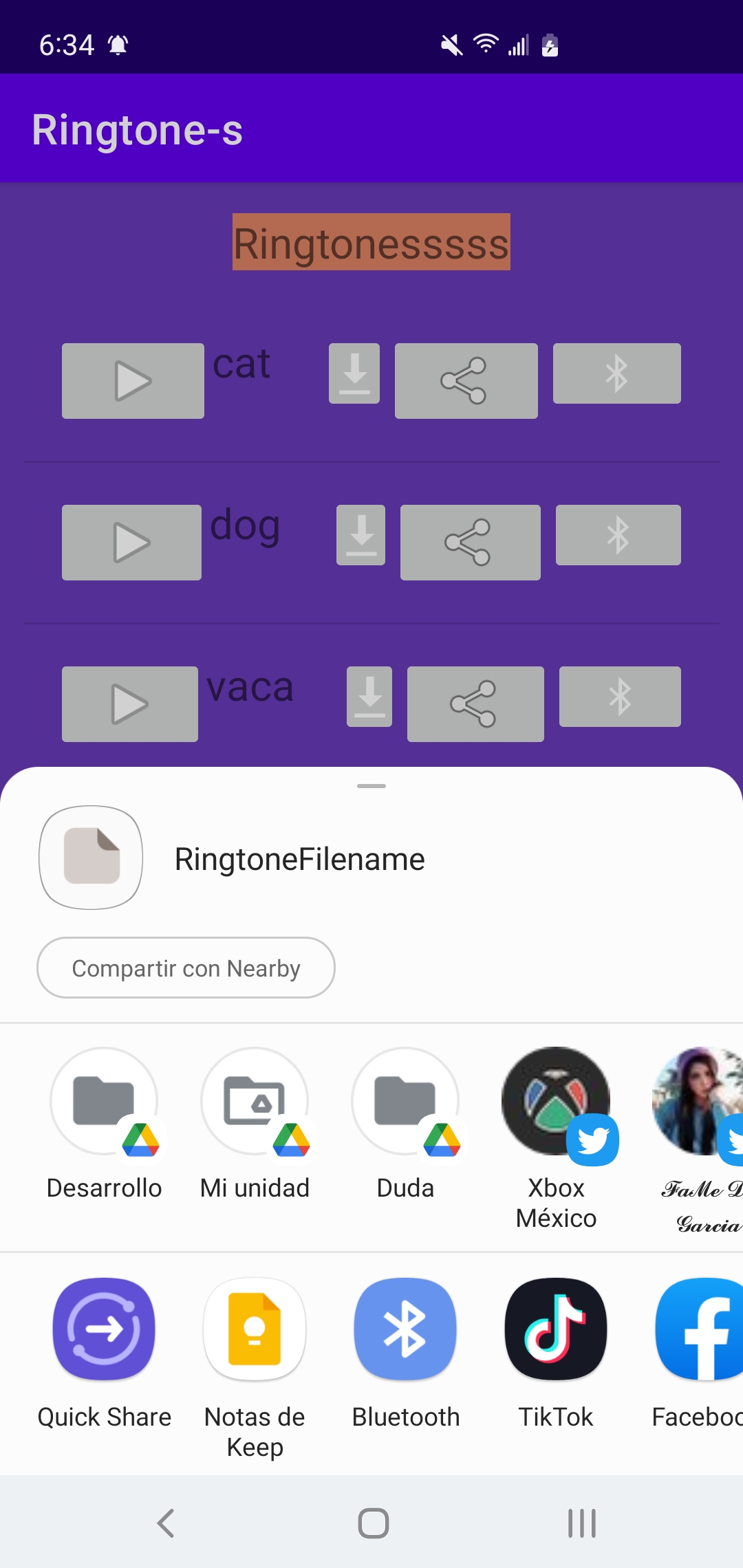
***Indice***

***[Codificación 4](#_Toc22951)***

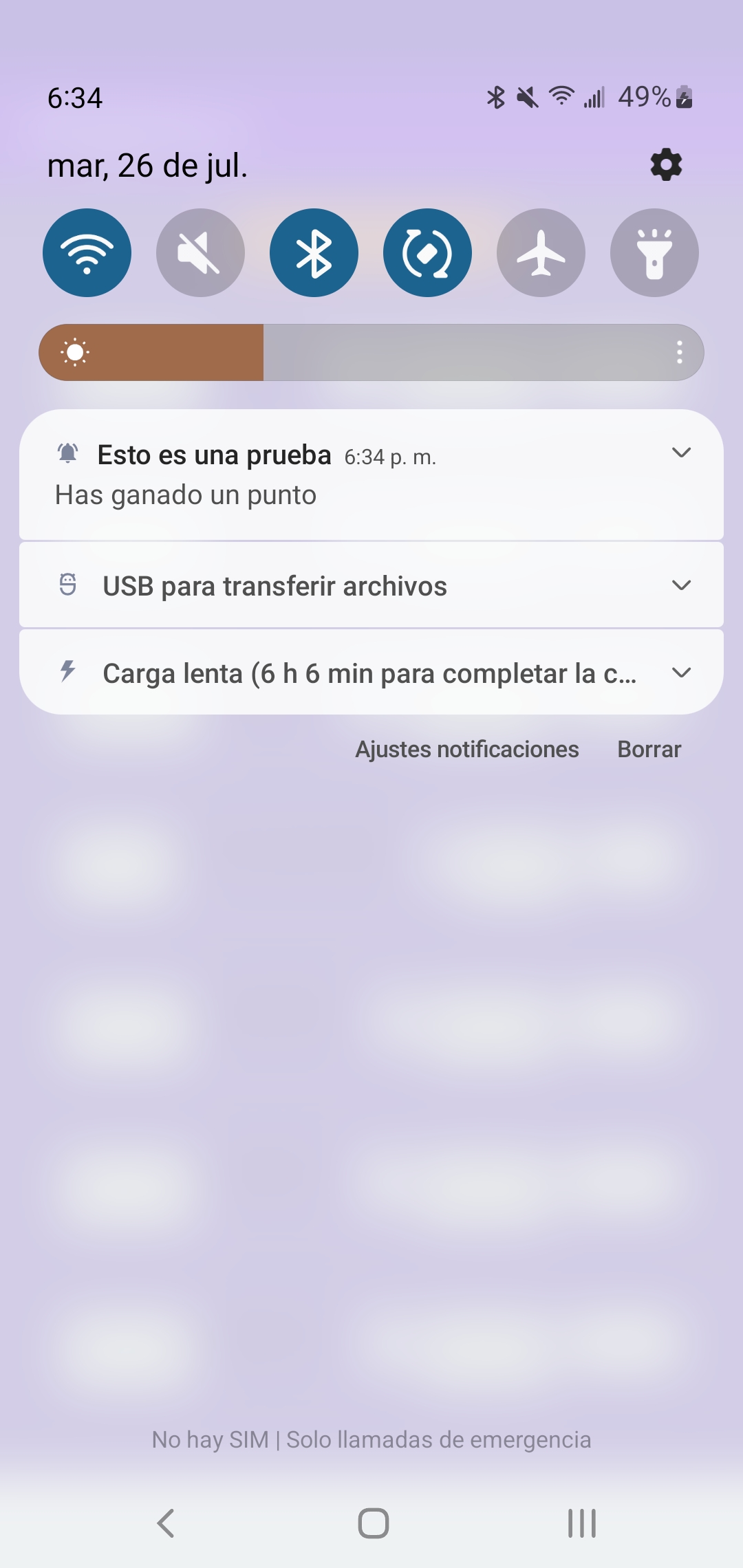
***[Prueba de Aplicación 20](#_Toc22030)***

***[Conclusion 23](#_Toc22905)***

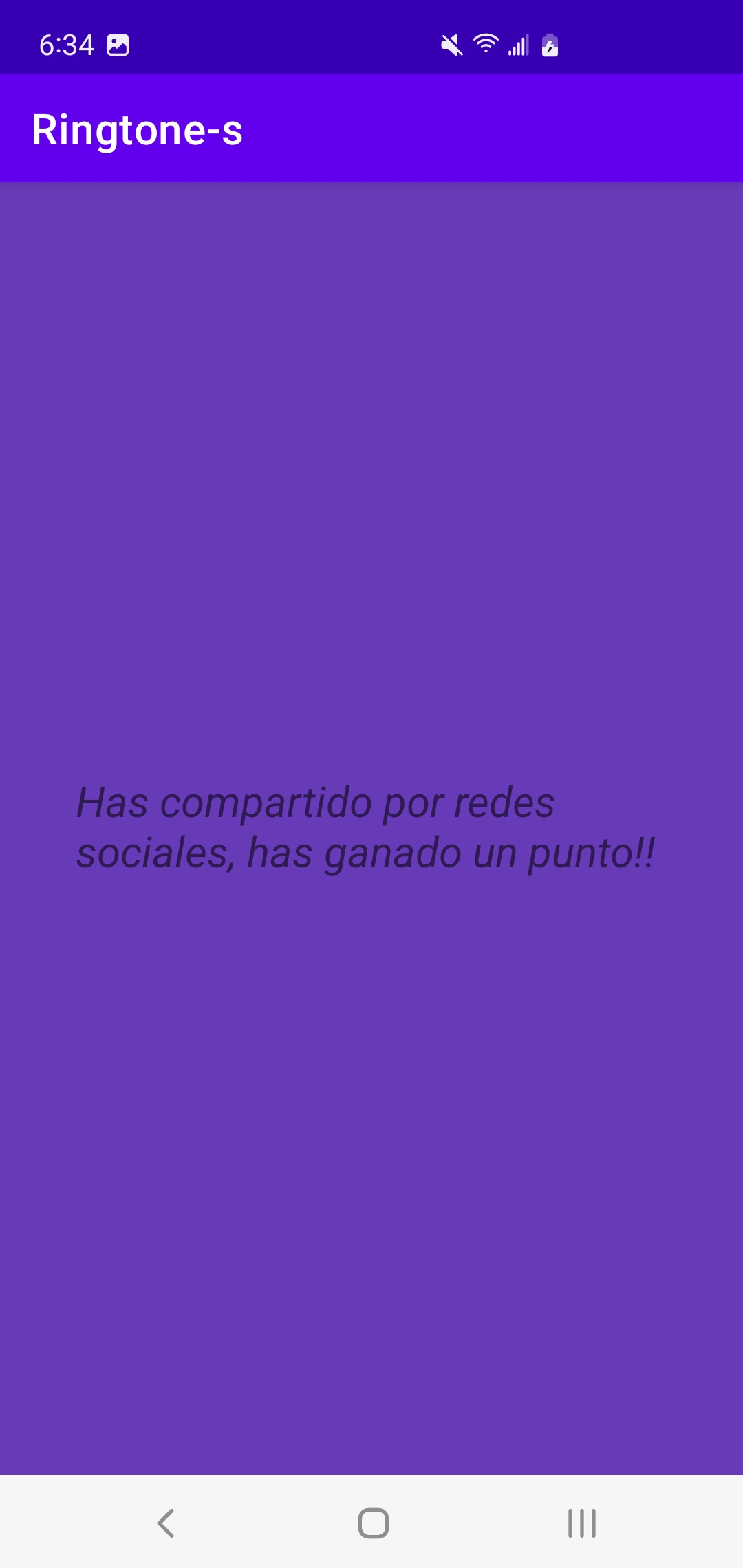
***Interfaz***



Al hacer clic en la parte de compartir se desplegara de forma inmediata una notificación la cual se vera reflejada en la parte superior con un pequeño icono de una campana.



Inmediatamente abrimos la barra de notificaciones y se mostrara el siguiente mensaje ”Esto es una prueba, has ganado un punto”



Al hacer clic en la notificación antes expuesta nos despejara una “segunda pantalla” la cual nos mostrara el siguiente mensaje.

# *Codificación*

*Android manifest*

<?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=".MessageActivity"  
 android:exported="false" />  
 <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>

*Main activity*

package mx.umi  
  
import android.app.NotificationManager  
import android.content.Context  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.ListView  
  
class MainActivity : AppCompatActivity() {  
  
 lateinit var notificationManager: NotificationManager  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 // it is a class to notify the user of events that happen.  
 // This is how you tell the user that something has happened in the  
 // background.  
 notificationManager = getSystemService(Context.*NOTIFICATION\_SERVICE*) as NotificationManager  
  
 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, notificationManager)  
  
 ringtonesList.*adapter* = adapter  
 }  
  
}

*Message Activity*

package mx.umi  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.widget.TextView  
  
class MessageActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_message*)  
  
 val message\_tv = findViewById<TextView>(R.id.*message\_tv*)  
 message\_tv.*text* = "¡Has compartido un audio! Ganaste un punto!"  
 }  
}

*Ringtone*

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

*Ringtone Adapter*

package mx.umi  
  
import android.app.Notification  
import android.app.NotificationChannel  
import android.app.NotificationManager  
import android.app.PendingIntent  
import android.content.Context  
import android.content.Intent  
import android.graphics.BitmapFactory  
import android.graphics.Color  
import android.media.MediaPlayer  
import android.net.Uri  
import android.os.Build  
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>, private val notificationManager: NotificationManager) : 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)  
 sendNotification(notificationManager)  
 **}** 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)  
 sendNotification(notificationManager)  
 **}** 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  
 }  
  
 fun sendNotification(notificationManager: NotificationManager){  
 lateinit var notificationChannel: NotificationChannel  
 lateinit var builder: Notification.Builder  
 val channelId = "i.apps.notifications"  
 val description = "Test notification"  
 // pendingIntent is an intent for future use i.e after  
 // the notification is clicked, this intent will come into action  
 val intent = Intent(mContext, MessageActivity::class.*java*)  
  
  
 // FLAG\_UPDATE\_CURRENT specifies that if a previous  
 // PendingIntent already exists, then the current one  
 // will update it with the latest intent  
 // 0 is the request code, using it later with the  
 // same method again will get back the same pending  
 // intent for future reference  
 // intent passed here is to our afterNotification class  
 val pendingIntent = PendingIntent.getActivity(mContext, 0, intent, PendingIntent.*FLAG\_UPDATE\_CURRENT*)  
  
 // checking if android version is greater than oreo(API 26) or not  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 notificationChannel = NotificationChannel(channelId, description, NotificationManager.*IMPORTANCE\_HIGH*)  
 notificationChannel.enableLights(true)  
 notificationChannel.*lightColor* = Color.*GREEN* notificationChannel.enableVibration(true)  
 notificationManager.createNotificationChannel(notificationChannel)  
  
 builder = Notification.Builder(mContext, channelId)  
 .setSmallIcon(R.drawable.*ic\_notification*)  
 .setLargeIcon(BitmapFactory.decodeResource(mContext.*resources*, R.drawable.*ic\_launcher\_background*))  
 .setContentTitle("Esta es una prueba")  
 .setContentText("Has ganado un punto")  
 .setContentIntent(pendingIntent)  
 .setAutoCancel(true)  
 } else {  
  
 builder = Notification.Builder(mContext)  
 .setSmallIcon(R.drawable.*ic\_notification*)  
 .setLargeIcon(BitmapFactory.decodeResource(mContext.*resources*, R.drawable.*ic\_launcher\_background*))  
 .setContentTitle("Esta es una prueba")  
 .setContentText("Has ganado un punto")  
 .setContentIntent(pendingIntent)  
 .setAutoCancel(true)  
 }  
 notificationManager.notify(1234, builder.build())  
 }  
}

*Ic notification*

<vector xmlns:android="http://schemas.android.com/apk/res/android"  
 android:width="24dp"  
 android:height="24dp"  
 android:viewportWidth="24"  
 android:viewportHeight="24"  
 android:tint="?attr/colorControlNormal">  
 <path  
 android:fillColor="@android:color/white"  
 android:pathData="M7.58,4.08L6.15,2.65C3.75,4.48 2.17,7.3 2.03,10.5h2c0.15,-2.65 1.51,-4.97 3.55,-6.42zM19.97,10.5h2c-0.15,-3.2 -1.73,-6.02 -4.12,-7.85l-1.42,1.43c2.02,1.45 3.39,3.77 3.54,6.42zM18,11c0,-3.07 -1.64,-5.64 -4.5,-6.32L13.5,4c0,-0.83 -0.67,-1.5 -1.5,-1.5s-1.5,0.67 -1.5,1.5v0.68C7.63,5.36 6,7.92 6,11v5l-2,2v1h16v-1l-2,-2v-5zM12,22c0.14,0 0.27,-0.01 0.4,-0.04 0.65,-0.14 1.18,-0.58 1.44,-1.18 0.1,-0.24 0.15,-0.5 0.15,-0.78h-4c0.01,1.1 0.9,2 2.01,2z"/>  
</vector>

*Activity Main*

<?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="@string/title"  
 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>

*Activity message*

<?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="306dp"  
 android:layout\_height="59dp"  
 android:foregroundTint="#FFFFFF"  
 android:text="Has compartido por redes sociales, has ganado un punto!!"  
 android:textSize="20sp"  
 android:textStyle="italic"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

*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>

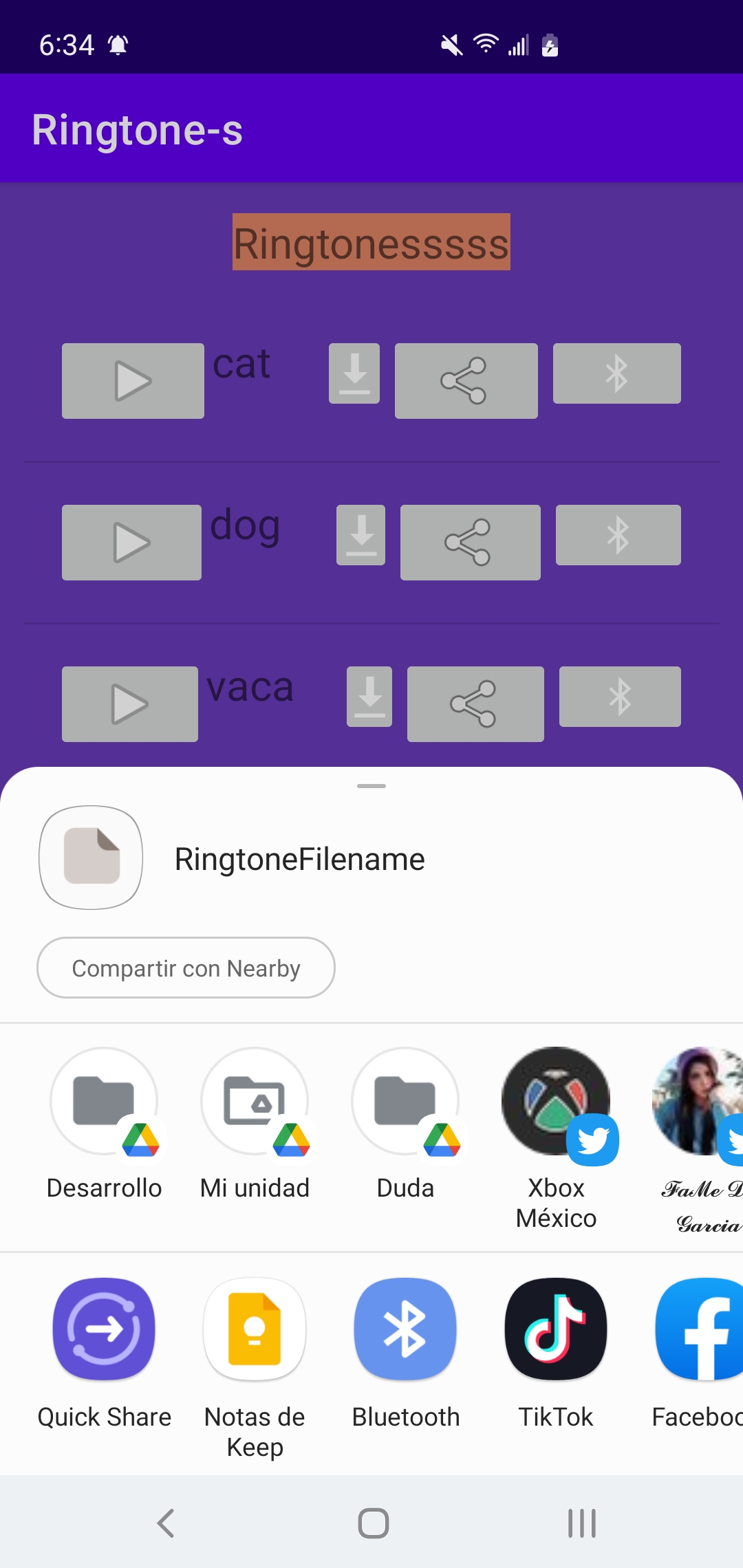
*Strings*

<resources>  
 <string name="app\_name">Ringtone-s</string>  
 <string name="title">Ringtone-s</string>  
 <string name="message\_text">Compartiste\_un\_audio..</string>  
</resources>

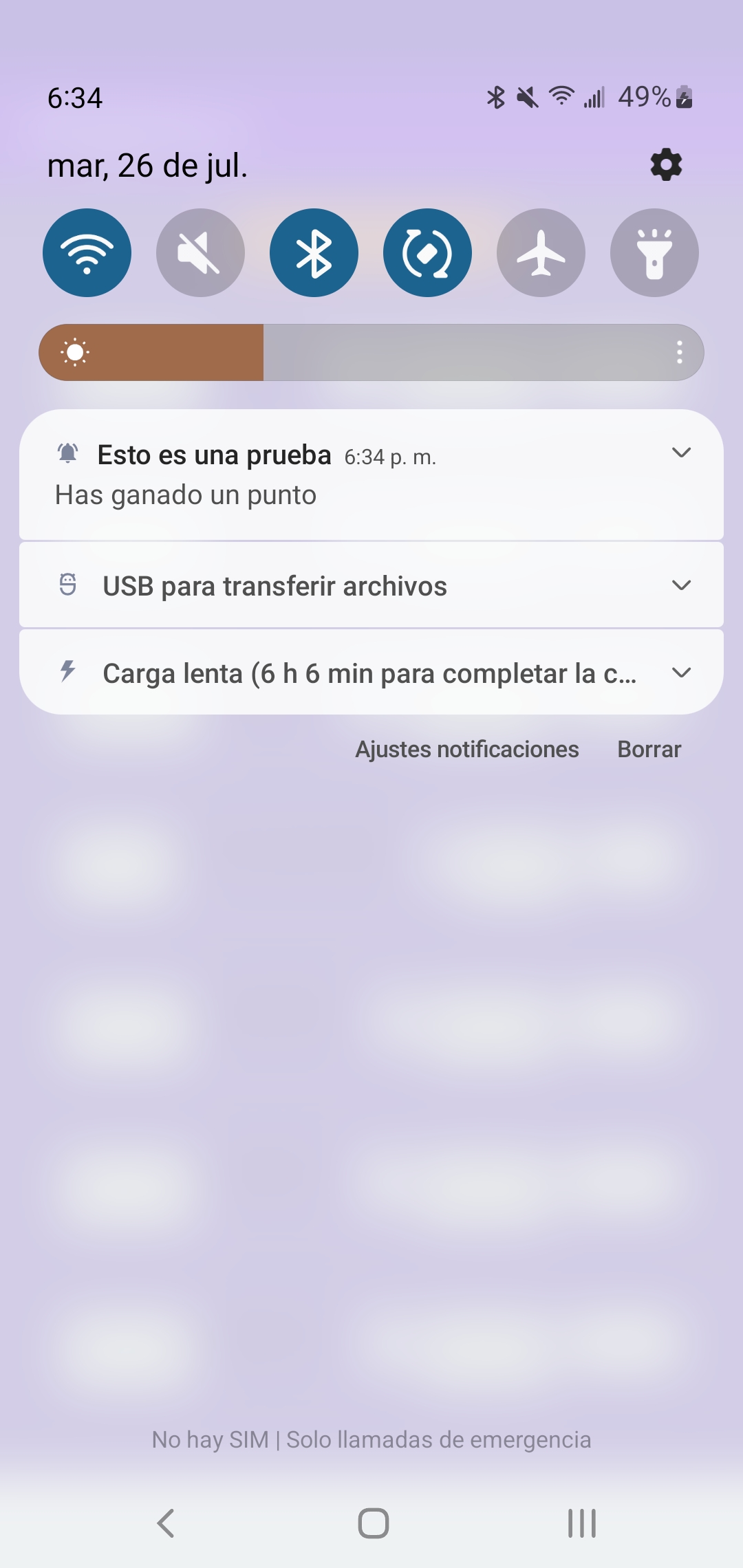
*Paths*

<?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>

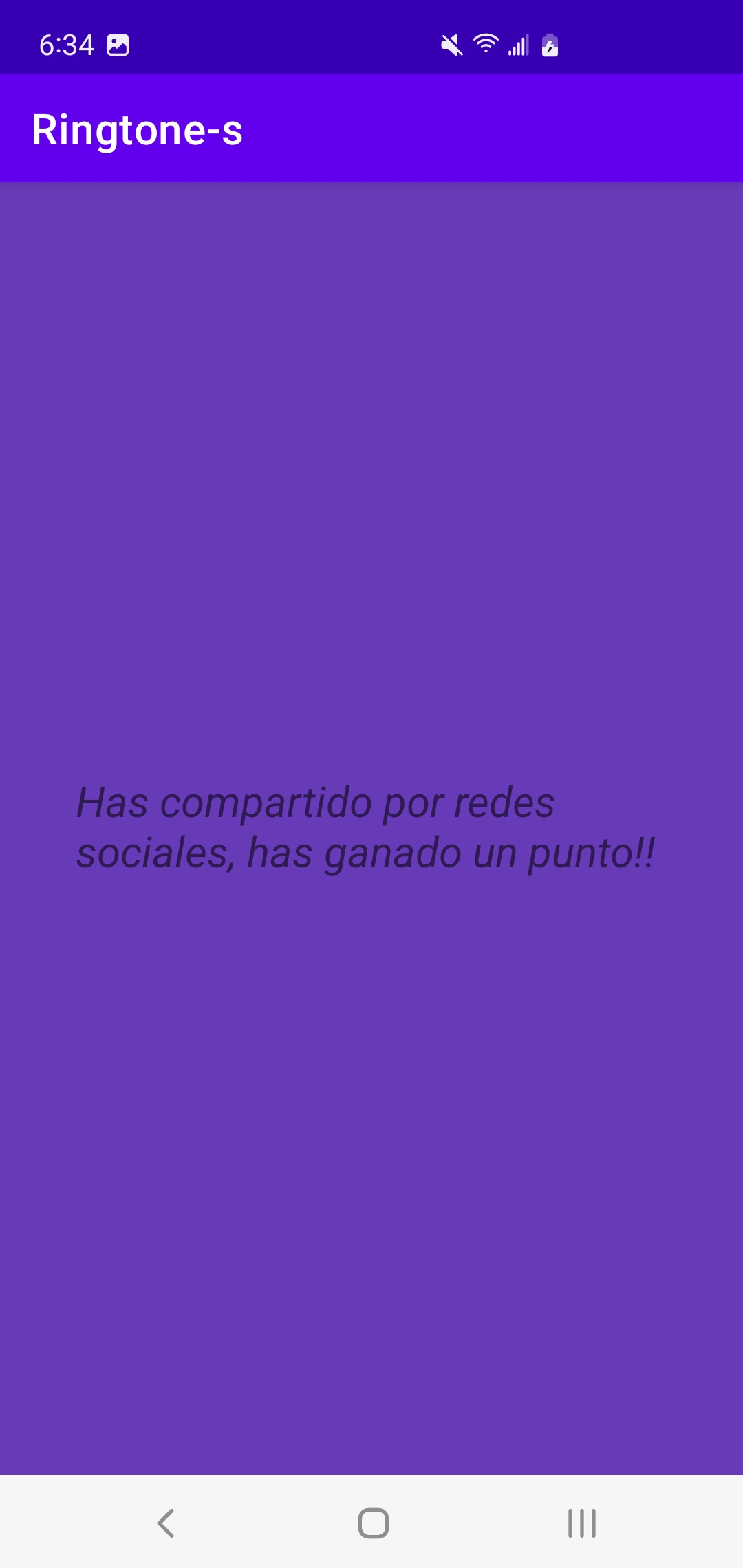
# *Prueba de Aplicación*



Al hacer clic en la parte de compartir se desplegara de forma inmediata una notificación la cual se vera reflejada en la parte superior con un pequeño icono de una campana.



Inmediatamente abrimos la barra de notificaciones y se mostrara el siguiente mensaje ”Esto es una prueba, has ganado un punto”



Al hacer clic en la notificación antes expuesta nos despejara una “segunda pantalla” la cual nos mostrara el siguiente mensaje.

# *Conclusion*

Para la creacion de cada App que se puede usar para distintos fines pero con un mismo objetivo es de suma importancia el saber el alcance que pueda tener, es decir saber que tan rentable resultara a corto, mediano y largo plazo para así colocarlo o crearlo de forma modularizada para que agregues, modifiques o elimines funciones que te puedan o no servir, adjunto a esto saber para que dispositivos y versiones quiero que esta funcione para así llegar a un numero máximo de usuarios activos (siempre y cuando se nos permita) con esto me refiero a la capacidad que pueda tener el servidor si es que este lo usa... En pocas palabras tener planificado de tal forma que sea rentable económicamente su mantenimiento y evolución.

Link Google drive:

<https://drive.google.com/file/d/1QkDbN2jtAe0W7A7WjqauSShN6fVPB3mp/view?usp=sharing>