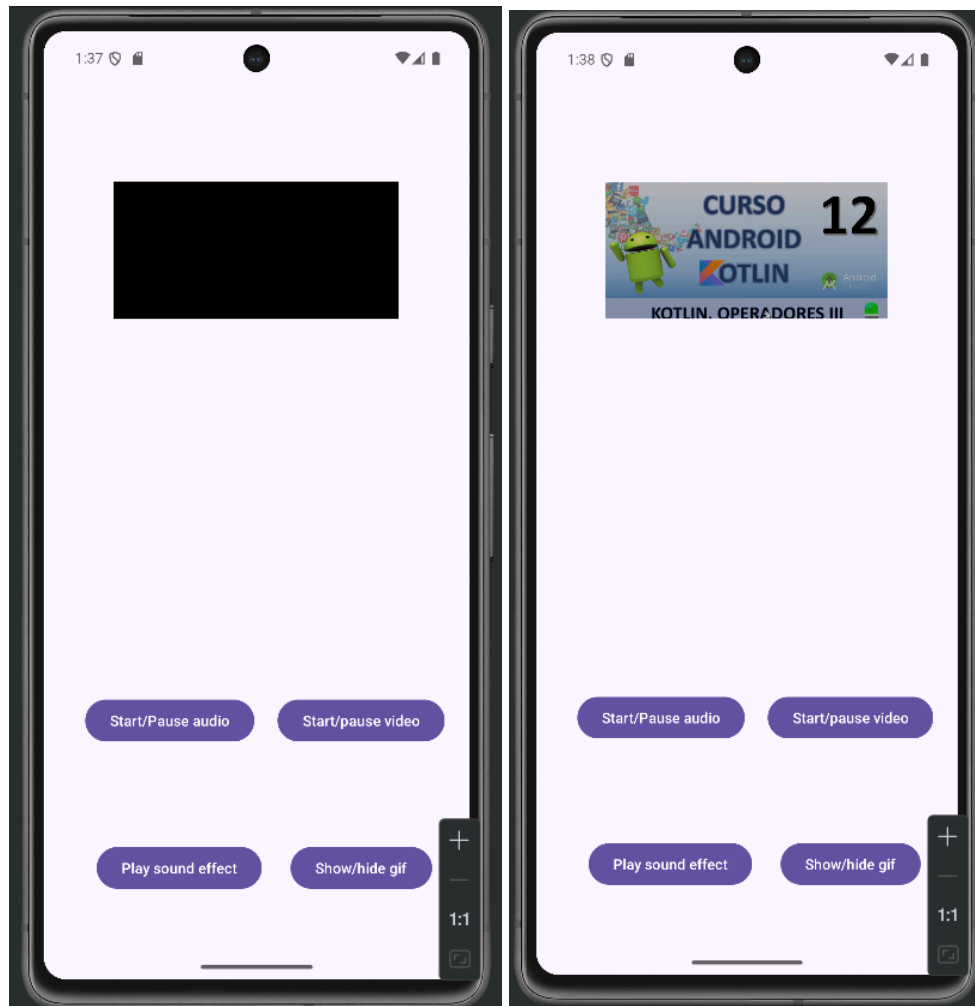


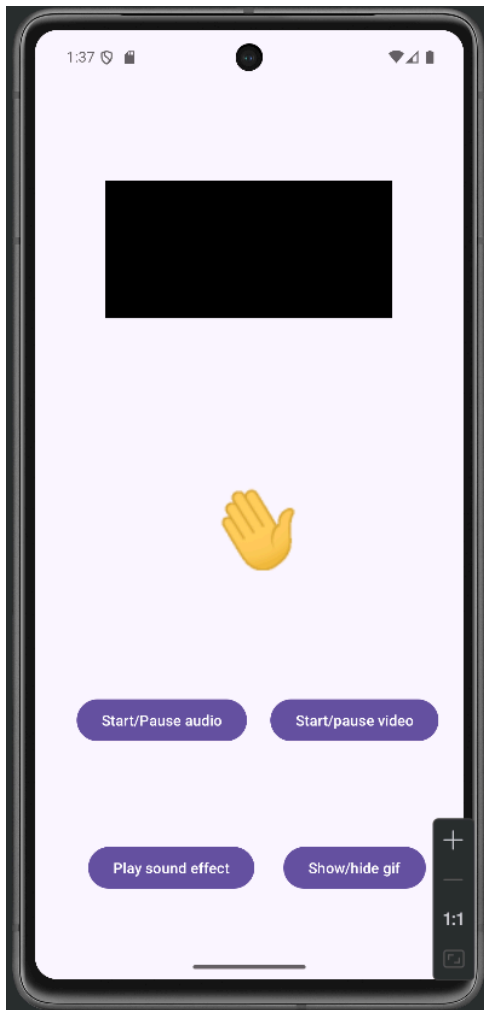
Tarea individual 9 - Uso de librerías de Audio, Video e Imagen

Actividad vacía con cuatro botones, VideoView e ImageView:



En sound effect reproduce un sonido de un segundo, estilo notificación, usando `'SoundPool'`

El Start/pause audio trabaja con MediaPlayer, y todos los archivos usados estan en res/raw, excepto el gif, que esta en res/drawable.



Para dibujar el gif he tenido que usar la libreria 'Glide', ya que el ImageView no soporta el formato Gif directamente, sino que hay que convertirlo antes.

Código en Kotlin:

```
package com.example.ejer9

import android.media.AudioAttributes
import android.media.MediaPlayer
import android.media.SoundPool
import android.net.Uri
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.VideoView
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.bumptech.glide.Glide
```

```

class MainActivity : AppCompatActivity() {

    private lateinit var soundPool: SoundPool
    private var soundId: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v,
insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }

        val videoView: VideoView = findViewById(R.id.videoView)
        val videoPath:String = "android.resource://" + getPackageName() + "/" +
R.raw.sample_video
        val videoUri: Uri = Uri.parse(videoPath)
        videoView.setVideoURI(videoUri)

        val audioPath:String = "android.resource://" + getPackageName() + "/" +
R.raw.queen_dontstopmenow
        val audioUri:Uri = Uri.parse(audioPath)
        val mediaPlayer:MediaPlayer = MediaPlayer.create(this, audioUri)

        val btnStateAudio:Button = findViewById(R.id.btnStateAudio)
        btnStateAudio.setOnClickListener{

            if(mediaPlayer.isPlaying){
                mediaPlayer.stop()
                mediaPlayer.prepare()
            }else{
                mediaPlayer.start()
            }

        }

        val btnStateVideo:Button = findViewById(R.id.btnStateVideo)
        btnStateVideo.setOnClickListener{

            if(videoView.isPlaying){
                videoView.pause()
            }else{

```

```

        videoView.start()
    }

}

val ivGif: ImageView = findViewById(R.id.ivGif)
var isGifShown:Boolean = false
Glide.with(this).asGif().load(R.drawable.hi).override(60,60).into(ivGif)

val btnStateGif:Button = findViewById(R.id.btnStateGif)
btnStateGif.setOnClickListener{

    isGifShown = !isGifShown

    if(isGifShown){
        ivGif.visibility = View.VISIBLE

    } else {
        ivGif.visibility = View.GONE
    }

}

val audioAttributes = AudioAttributes.Builder()
    .setUsage(AudioAttributes.USAGE_MEDIA)
    .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
    .build()

soundPool = SoundPool.Builder()
    .setMaxStreams(1) //Maximo de sonidos simultaneos
    .setAudioAttributes(audioAttributes)
    .build()

soundId = soundPool.load(this, R.raw.notification, 1)

val btnPlayEffect:Button = findViewById(R.id.btnPlayEffect)
btnPlayEffect.setOnClickListener{
    soundPool.play(soundId, 1.0f, 1.0f, 0, 0, 1.0f) // Volumen 1.0 (izq y
der), sin loop, velocidad normal
}
}
}

```