

DESARROLLO DE UNA MESA DE MEZCLAS VIRTUAL PARA DISPOSITIVOS ANDROID

Grado en Ingeniería de Sistemas Audiovisuales y Multimedia

Autor: María de la Osa Martínez

Tutor: David Gualda Gómez

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

1. Introducción

Diseño y desarrollo de una mesa de mezclas virtual minimalista para dispositivos móviles.



Iconos Android y Kotlin combinados [1]

¿Por qué una mesa de mezclas?

- Potenciar los conocimientos adquiridos
- Afrontar un reto de programación



Mesa de mezclas [2]

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

2. Objetivos y fases de desarrollo



Diagrama de objetivos

2. Objetivos y fases de desarrollo

Meses	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Investigación																
Diseño																
Desarrollo																
Resultados																
Documentación																

Diagrama de Gantt sobre las fases del proyecto

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
- 3. Definiciones y conceptos**
4. Mesa de mezclas
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

3. Definiciones y conceptos

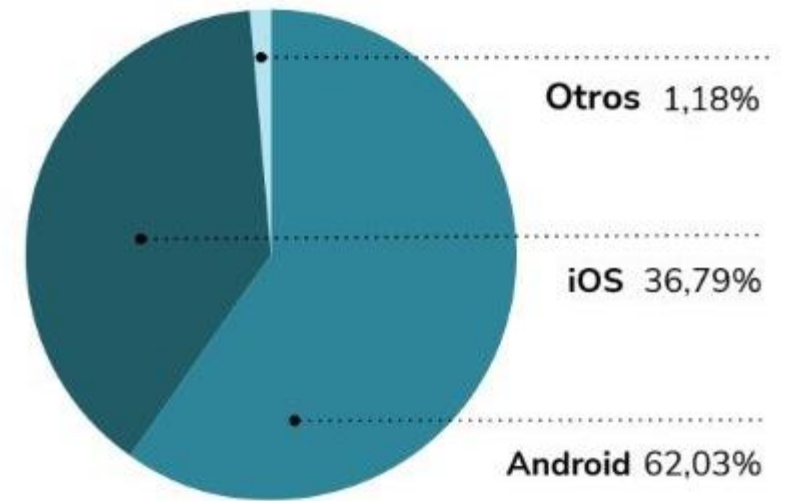
Android

- Sistema operativo de código abierto creado en 2005
- 13 versiones disponibles
- Diseño de interfaz intuitivo
- Se trata del sistema operativo más utilizado en el mundo



Logotipo Android [3]

% de Ingresos desde móvil
distribuido por sistema operativo

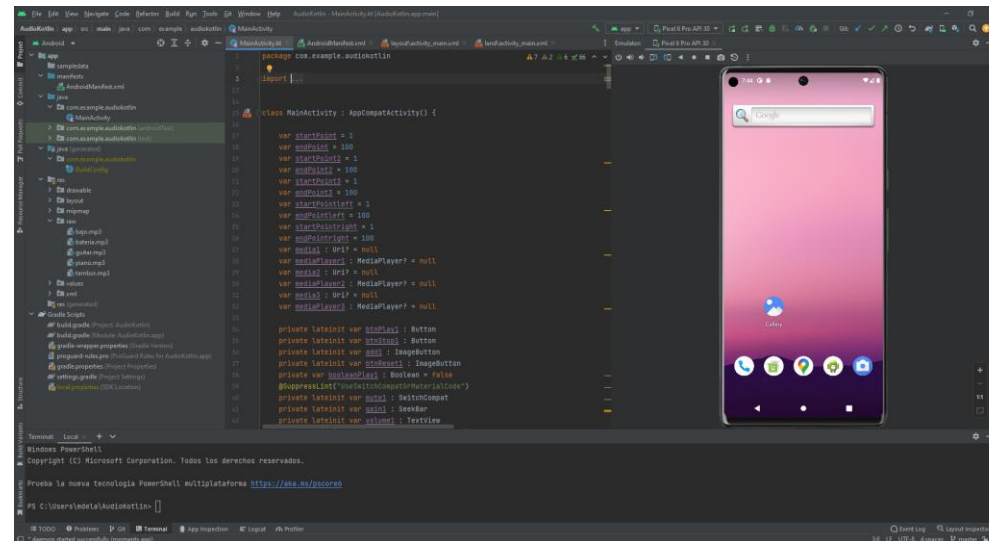


Comparativa usuarios de Android frente a iOS en 2022 [4]

3. Definiciones y conceptos

Android Studio

- Entorno de desarrollo integrado oficial para Android
- Diferentes lenguajes válidos
- Posee editor de código, sistema de compilación y funciones adicionales para añadir recursos.
- Emuladores para diferentes dispositivos.



Vista Android Studio

3. Definiciones y conceptos

Kotlin

- Google le da soporte desde 2017
- Versátil y moderno
- Curva de aprendizaje sencilla
- Elimina la redundancia que se puede encontrar en Java.
- Simplificación de las llamadas de red y acceso a base de datos
- Puede ser usado con programación lineal o con programación orientada a objetos
- Mayor seguridad que otros lenguajes



Logotipo Kotlin [5]

3. Definiciones y conceptos

Extensiones de archivos de audio

Audio comprimido con pérdidas	Audio comprimido sin pérdidas
OGG (Ogg Vorbis)	FLAC (Free Lossless Audio Codec)
MP3 (MPEG-1 Audio Layer 3)	WAV (Waveform Audio File Format)
AAC (Advanced Audio Coding)	

3. Definiciones y conceptos

Ganancia frente a volumen

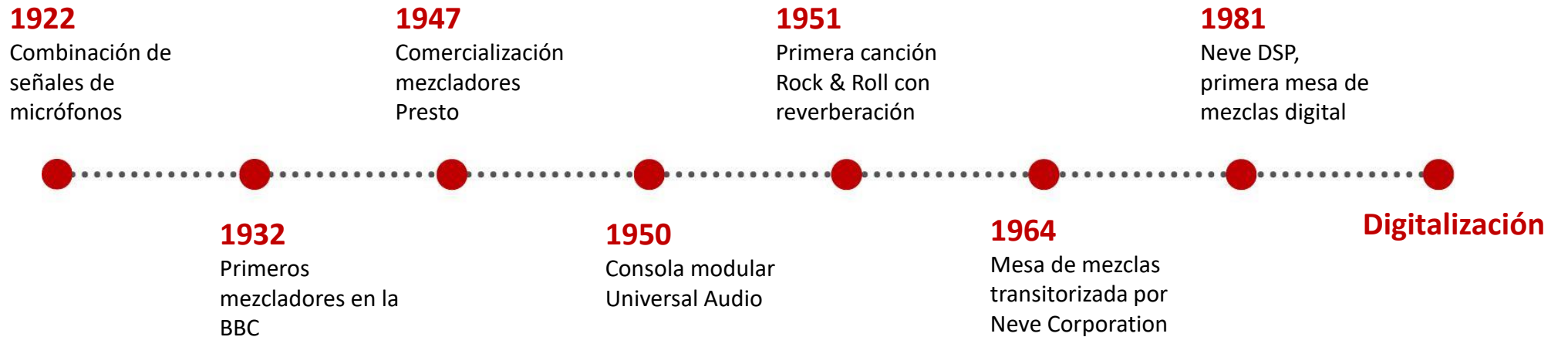
Ganancia	Volumen
RAE: acción y efecto de ganar	RAE: intensidad del sonido
Se mide en decibelos (dB)	Se trata de una percepción subjetiva de la potencia del sonido
A mayor ganancia, mayor distorsión	A mayor volumen, no hay distorsión

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
- 4. Mesa de mezclas**
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

4. Mesas de mezclas

Historia



4. Mesas de mezclas

Tipos de mesas

A pesar de que se pueden diferenciar por la tecnología utilizada o por la funcionalidad, todas trabajan de la misma manera.

Comparando tecnología utilizada
Analógicas
Digitales
Virtuales

Comparando la funcionalidad
Split o básicas
In line
Monitores
Mesas de DJ
Autoamplificadas
Portátiles

3. Definiciones y conceptos

Partes

CONEXIONES DE SALIDA

CONEXIONES DE ENTRADA

GANANCIA

CONEXIONES PARA
CANALES DE ENTRADA

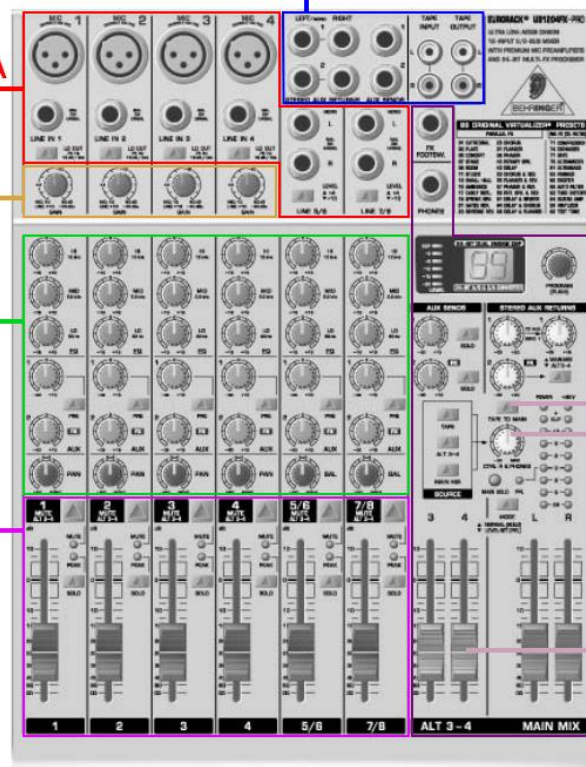
CONTROLES DE
ENRUTAMIENTO

CONTROLES DE SALIDA

PULSADOR
SWITCH

TIPO DE
BOTONES

FADER



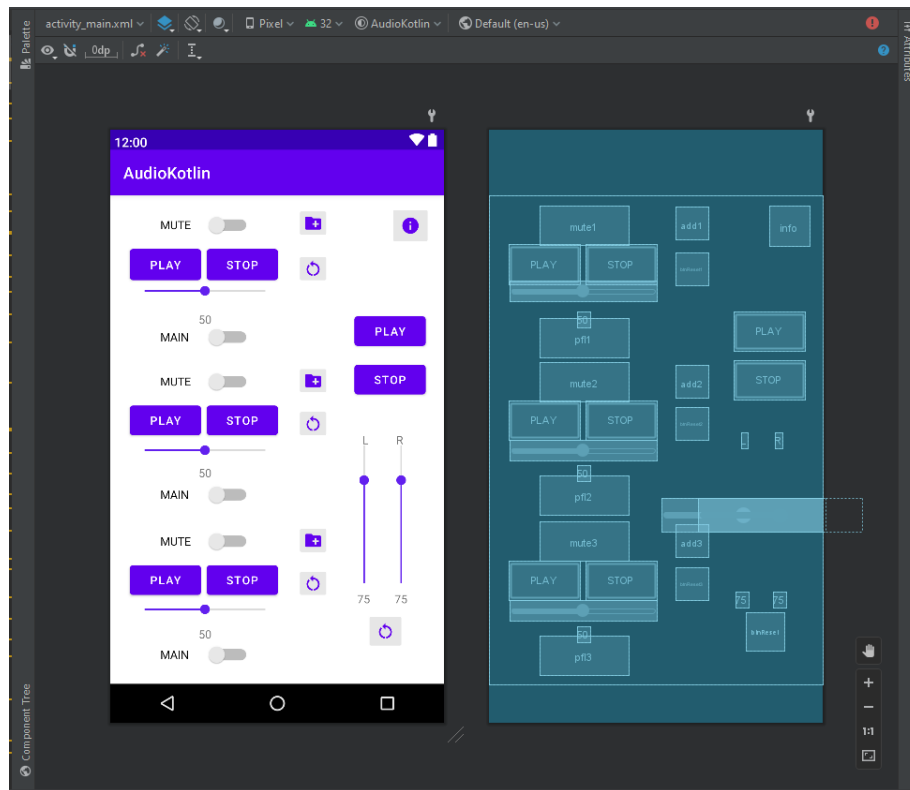
Explicación de las partes [6]

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
- 5. Desarrollo**
6. Resultados
7. Costes
8. Conclusiones

5. Desarrollo

Activity_main.xml



Diseño de Android Studio

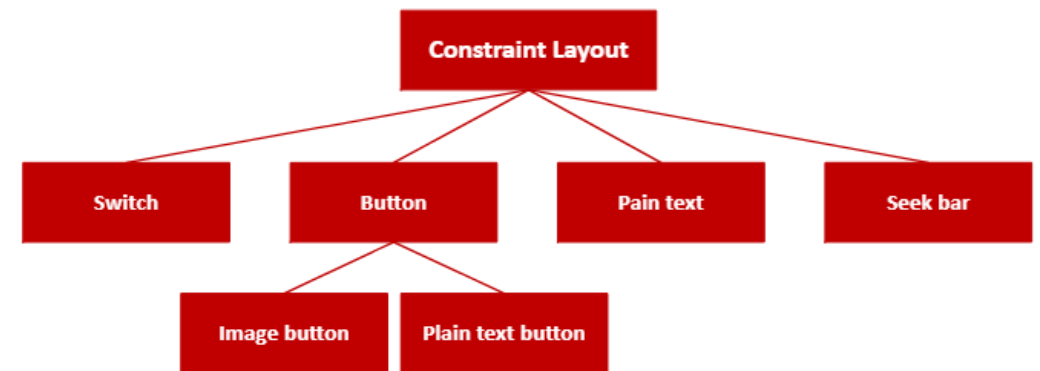


Diagrama de constantes

5. Desarrollo

MainActivity.kt

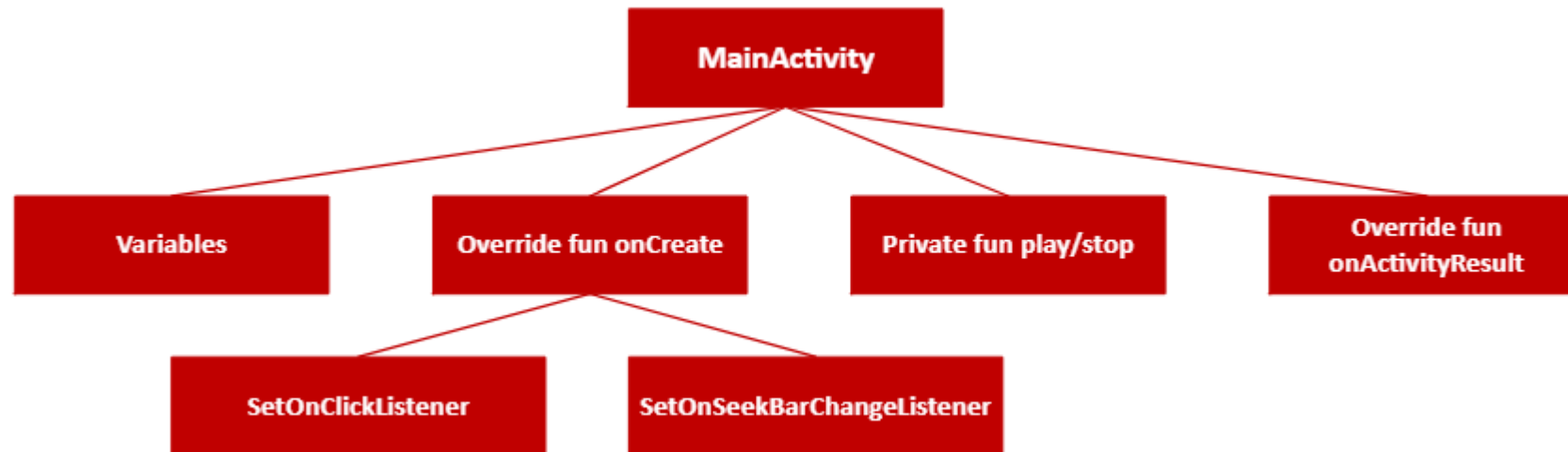


Diagrama de variables y funciones

5. Desarrollo

MainActivity.kt → override fun onCreate

```
gain1.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener{
    override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
        volume1.text = progress.toString()
        vol1 = (progress * 0.01).toFloat()
        if (booleanPlay) {
            mediaPlayer1?.setVolume((volleft*vol1),(volright*vol1))
        }else {
            mediaPlayer1?.setVolume(vol1, vol1)
        }
    }
})

override fun onStartTrackingTouch(seekBar: SeekBar?) {
    if (seekBar != null) {
        startPoint = seekBar.progress
    }
}

override fun onStopTrackingTouch(seekBar: SeekBar?) {
    if (seekBar != null) {
        endPoint = seekBar.progress
    }
    if (seekBar != null) {
        Toast.makeText(context: this@MainActivity, text: "Ganancia 1: " + seekBar.progress,
            Toast.LENGTH_SHORT).show()
    }
}
})
```

setOnSeekBarChangeListener pista 1

```
add1.setOnClickListener { it: View!
    val intent = Intent()
        .setType("audio/*")
        .setAction(Intent.ACTION_GET_CONTENT)
    startActivityForResult(Intent.createChooser(intent, title: "Select an audio file"), requestCode: 1)
}

btnPlay1.setOnClickListener{ it: View!
    booleanPlay1 = true
    if ((!booleanPlay) and (!booleanPlay2) and (!booleanPlay3)) {
        stopAudio1()
        playAudio1()
        mediaPlayer1?.setVolume(vol1, vol1)
    }
}

btnStop1.setOnClickListener{ it: View!
    stopAudio1()
    booleanPlay1 = false
}

btnReset1.setOnClickListener{ it: View!
    gain1.progress = 50
    media1 = null
    if (mute1.isChecked){
        mute1.isChecked = false
    }
}

mute1.setOnClickListener { it: View!
    if (mediaPlayer1?.isPlaying == true) {
        if (mute1.isChecked){
            mediaPlayer1?.setVolume( leftVolume: 0F, rightVolume: 0F)
        } else {
            mediaPlayer1?.setVolume(vol1,vol1)
        }
    }
}
}
```

setOnClickListener pista 1

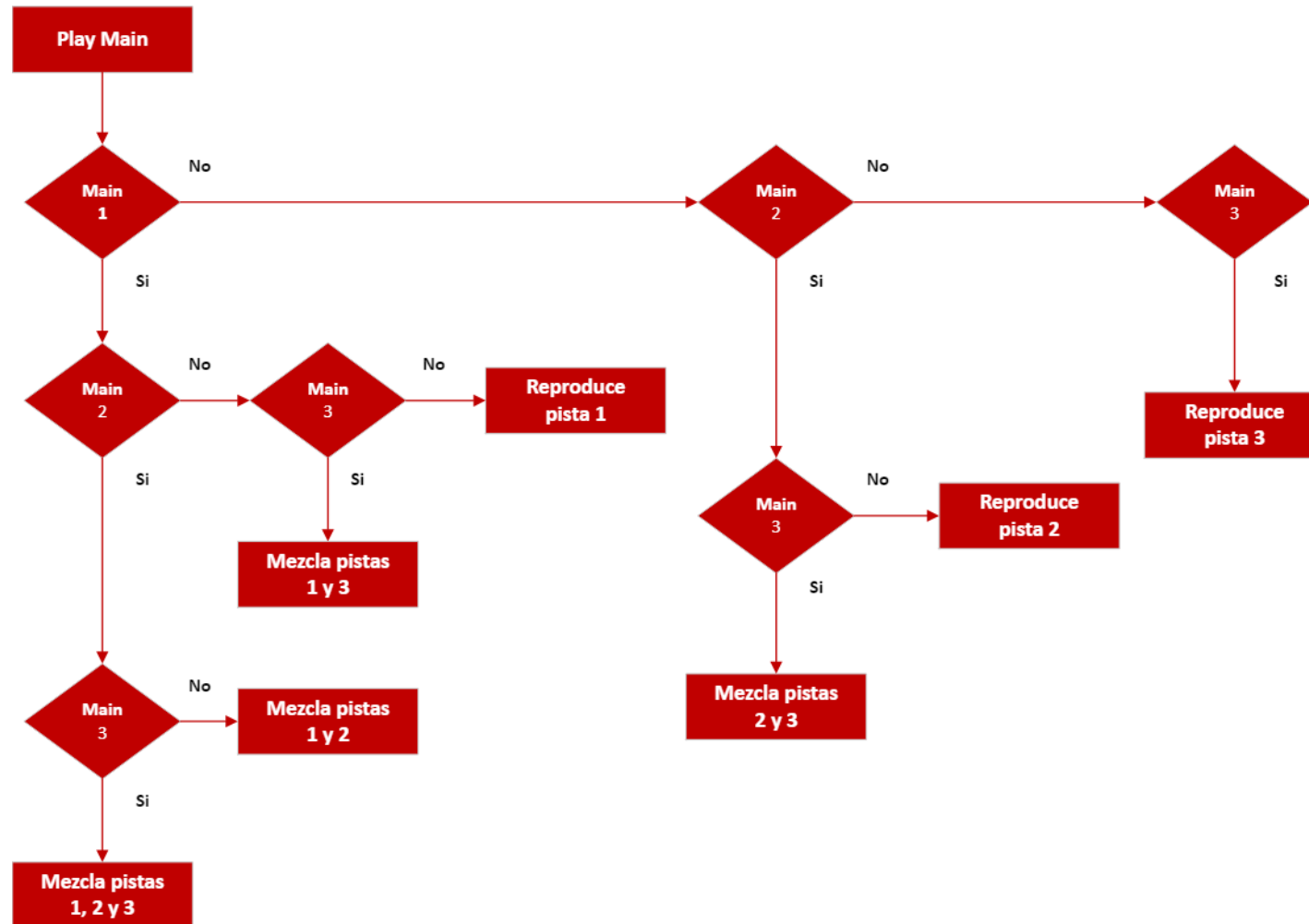
5. Desarrollo

MainActivity.kt → Private fun play y stop

```
private fun playAudio1(){
    if (media1 != null) {
        mediaPlayer1 = MediaPlayer.create(context: this, media1)
    }else{
        mediaPlayer1 = MediaPlayer.create(context: this, R.raw.piano)
    }
    mediaPlayer1!!.isLooping = true
    mediaPlayer1!!.start()
}
private fun stopAudio1(){
    if (mediaPlayer1?.isPlaying == true) {
        mediaPlayer1!!.stop()
        mediaPlayer1 = null
    }
}
}
```

Private fun play y stop pista 1

5. Desarrollo

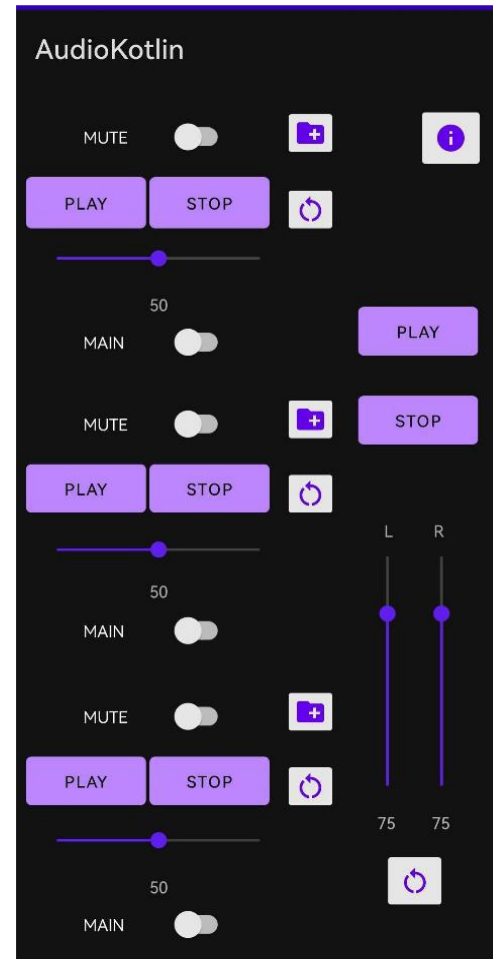


Flujograma de salida máster

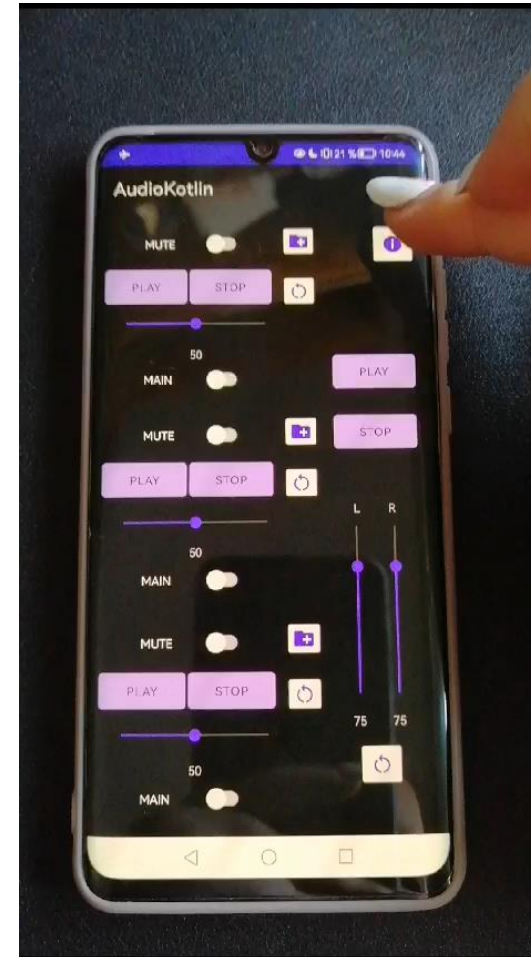
Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
- 6. Resultados**
7. Costes
8. Conclusiones

6. Resultado



Resultado del prototipo



Demo

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
6. Resultados
- 7. Costes**
8. Conclusiones

7. Costes

Costes materiales:

- Ordenador: 350€
- Dispositivo móvil Android: 200€
- Auriculares: 30€

Tiempo estimado: 360 horas

Costes humanos:

- Ingeniero/a con conocimientos de programación: salario medio de 30€/h neto, 10.800€ para el proyecto (360 x 30)

Tipo de coste	Coste total en euros (€)
Mano de obra	10.800€
Material	580€
Total	11.380€

Tabla de costes

Índice de contenidos

1. Introducción
2. Objetivos y fases de desarrollo
3. Definiciones y conceptos
4. Mesa de mezclas
5. Desarrollo
6. Resultados
7. Costes
8. Conclusiones

8. Conclusiones

Objetivo principal:

- Desarrollo de un prototipo

Competencias empleadas:

- Programación
- Conocimiento de audio
- Idioma Moderno

Competencias adquiridas:

- Kotlin y aplicaciones Android
- Ampliación del entendimiento del mundo del audio
- Recopilación y resumen de información

Trabajos futuros:

- Habilitar la vista para múltiples dispositivos
- Implementación de audios a partir de internet
- Rotación de la aplicación
- Introducción de filtros y herramientas de ecualización

- [1] https://2.bp.blogspot.com/-e5SH_qsqwfc/Wguw5FFJ99I/AAAAAAAAADZ4/TiktoHLI_9gdzR_INSsb-ulyewkkF3XDACLcBGAs/s640/jjhghjhj.jpg
- [2] <https://www.gladoop.com>
- [3] https://www.brandemia.org/wp-content/uploads/2012/10/logo_principal.jpg
- [4] <https://marketing4ecommerce.net/android-o-ios-que-sistema-operativo-consigue-mas-conversiones-flat101/>
- [5] https://cdn.icon-icons.com/icons2/2699/PNG/512/kotlinlang_logo_icon_170356.png

El proyecto se encuentra disponible en:

Código fuente Kotlin:

- <https://github.com/mdelaosa/AudioKotlin>

APK, demo, memoria y presentación:

- https://github.com/mdelaosa/Memoria_TFG

¡Gracias!