USO DE AUDIO CON ANDROID

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

LUIS HUMBERTO RIVAS RODRÍGUEZ
INGENIERO EN SISTEMAS INFORMÁTICOS
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.

Agenda

1 **Generalidades**
Conceptos

MediaPlayer 2
Funcionalidades

3 **Desafío de Clases**
Ejemplo Práctico

Cierre 4
Valoraciones Finales

RESULTADOS DE APRENDIZAJE

Competencias a desarrollar al finalizar la sesión.

INTERPRETA

Los componentes para usar audio dentro de una interfaz gráfica en Android.

COMPRENDE

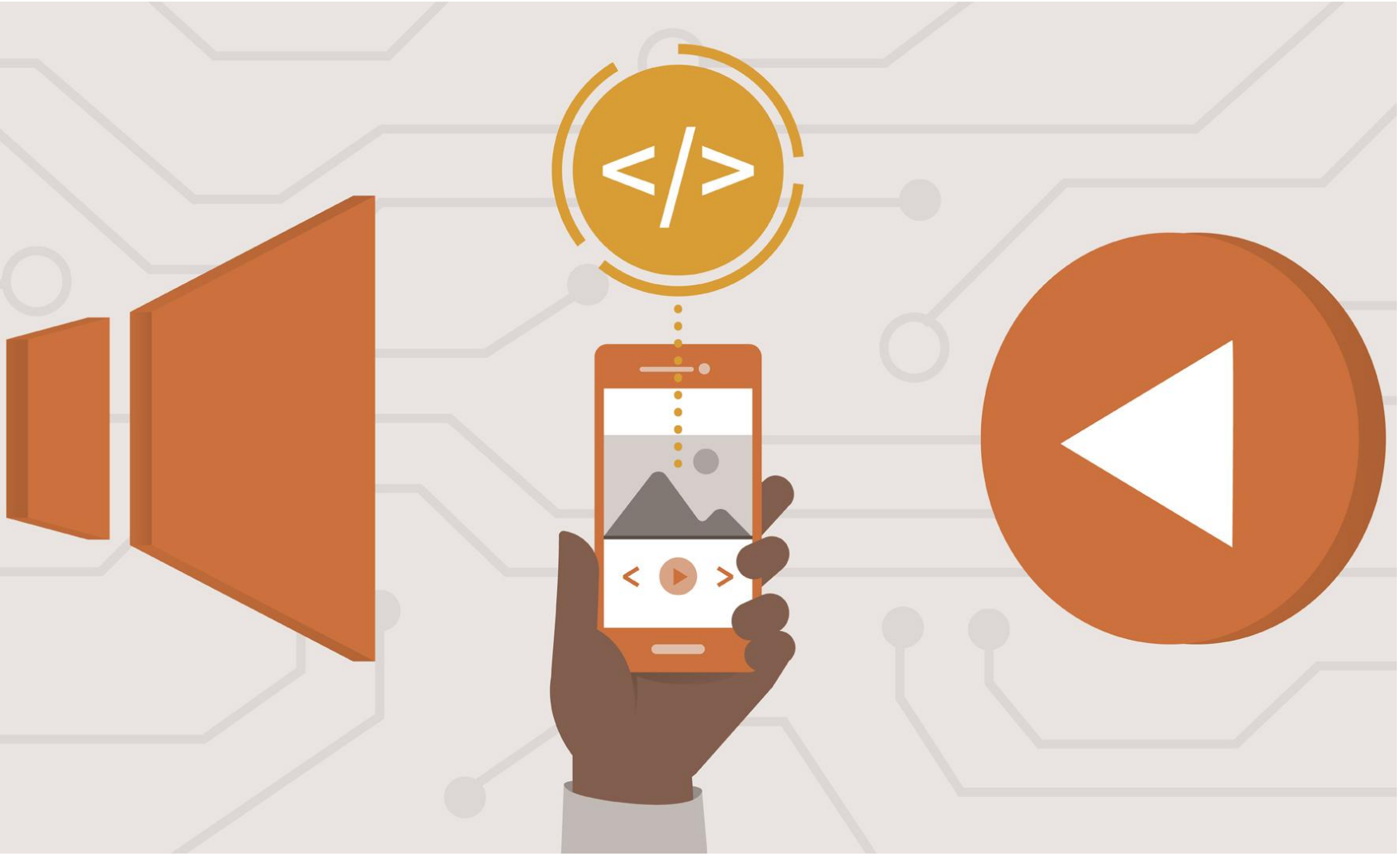
La relación que tienen las librerías y su importancia en el desarrollo de aplicaciones basadas en Android.

CONSTRUYE

Aplicaciones Android haciendo uso de los componentes avanzados como son las librerías en Android.

CONTEXTO **DE USO**

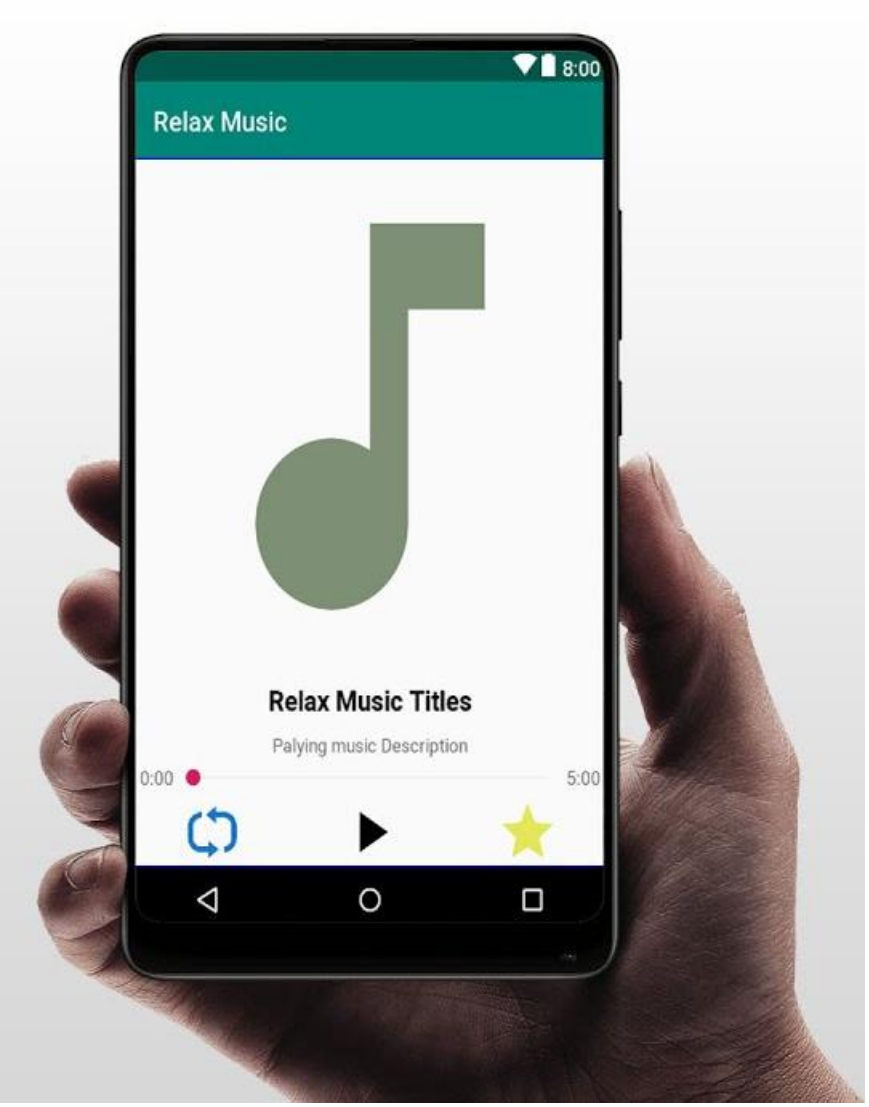
Requerimientos principales



- En algún momento se requiere hacer uso de contenido multimedia. (Audio, Video).
- Para esta sesión nos vamos a enfocar en el control MediaPlayer y como implementarlo a través de contenido local como remoto.

MEDIAPLAYER

COMPONENTE

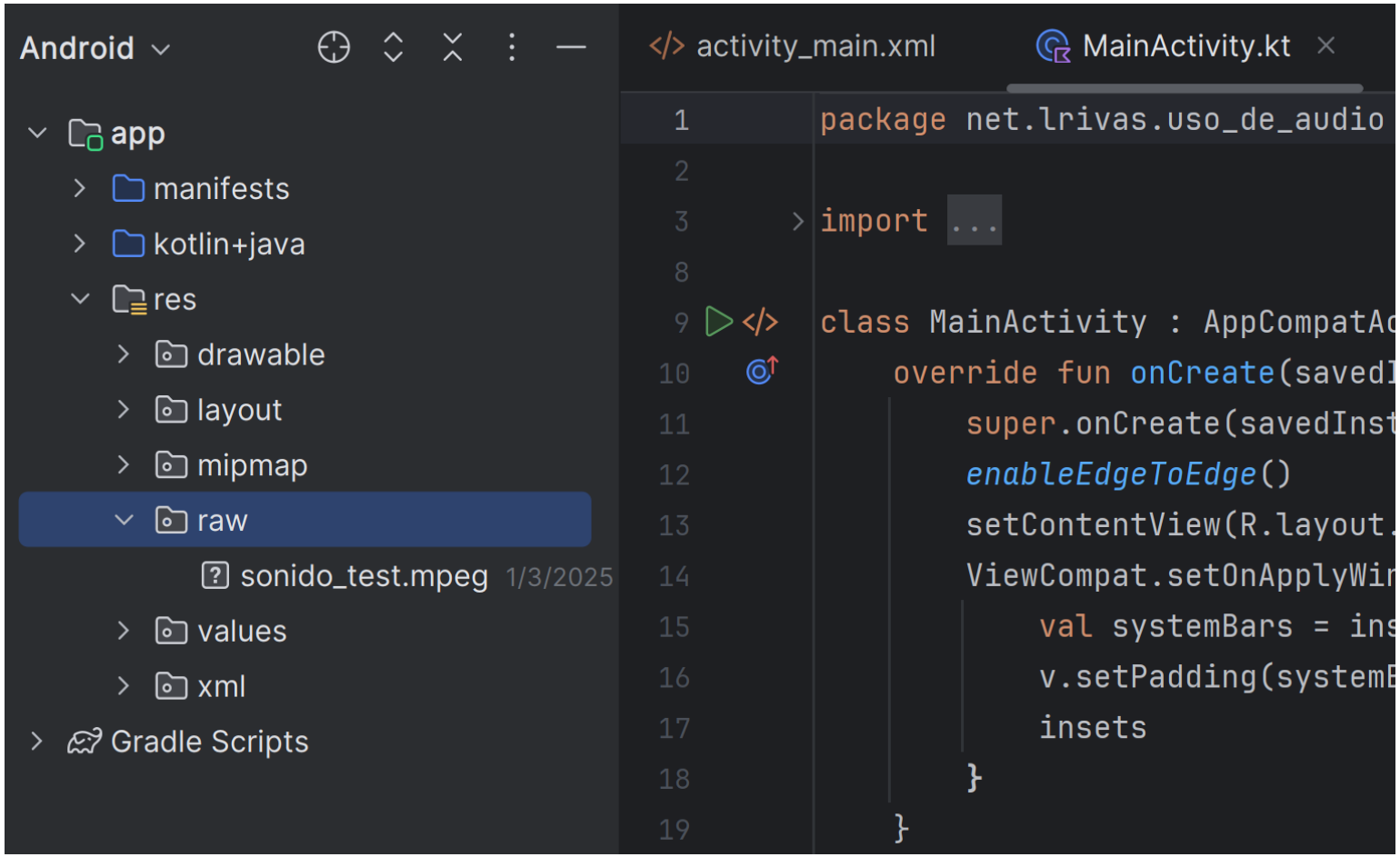


- El marco de trabajo de contenido multimedia de Android admite la reproducción de diversos tipos de contenido multimedia comunes para integrar audio, video e imágenes con facilidad en las apps.
- Puedes reproducir audio o video desde archivos multimedia almacenados en los recursos de tu app (recursos sin procesar), desde archivos independientes del sistema de archivos o desde un flujo de datos que llega a través de una conexión de red, todo mediante diferentes API de MediaPlayer.

Fuente: https://developer.android.com/guide/topics/media/mediaplayer?hl=es-419

DIRECTORIO RAW

CONTEXTUALIZACIÓN

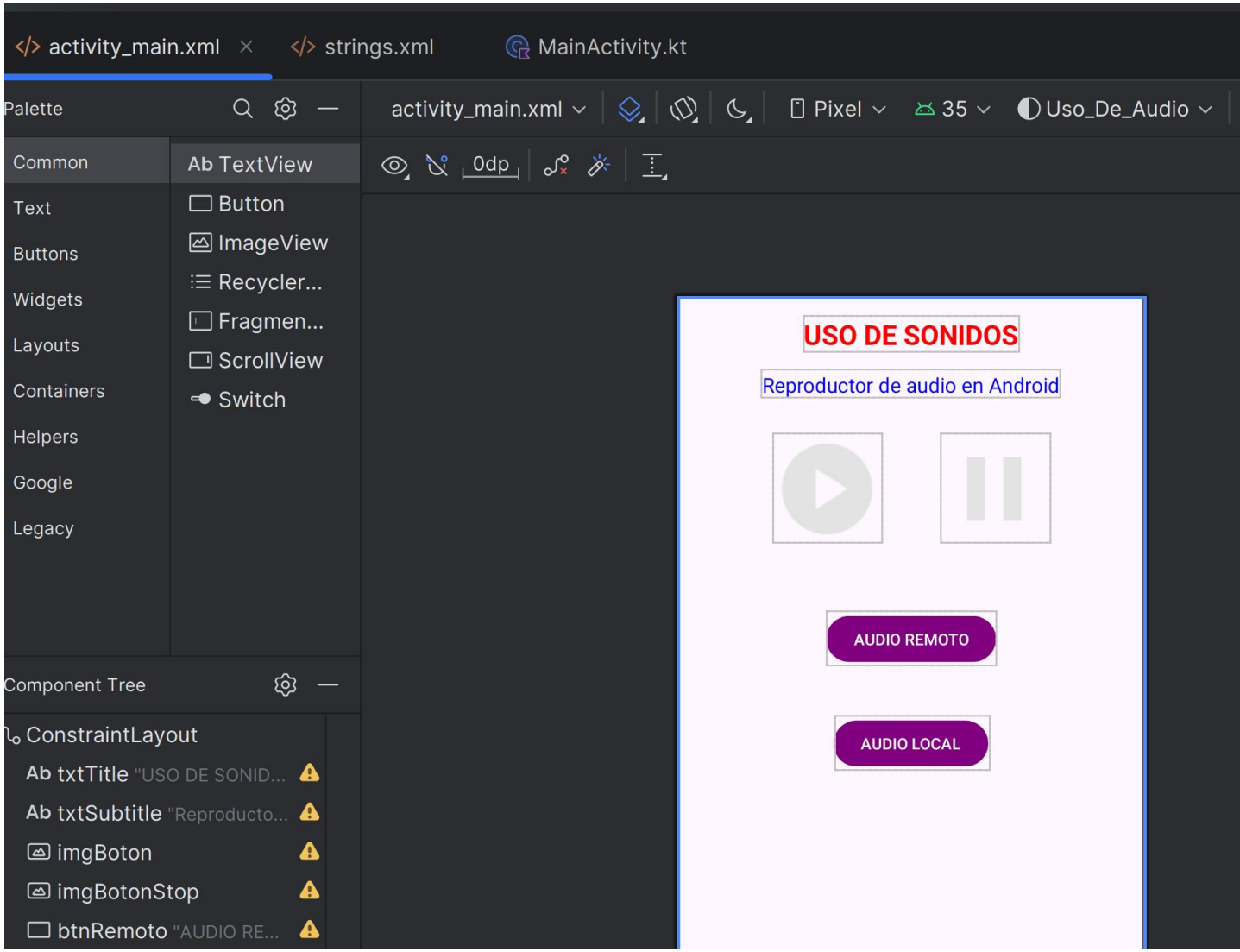


- Cuándo se trata de trabajar con contenido multimedia local, basta con agregar los recursos a la carpeta **raw**.
- Recuerde que se recomienda siempre utilizar nombres cortos, sin espacios ni mayúsculas.

USO DE MEDIA PLAYER

COMPONENTES

- El uso de la clase MediaPlayer permite ejecutar / procesar contenido local y/o remoto.
- Los métodos Create(), start(), pause(), stop() y release() son métodos esenciales en el uso de la clase.



REPRODUCCIÓN LOCAL

MEDIAPLAYER

```
btnLocal.setOnClickListener {
    try {
        if (reproductorMedia?.isPlaying == true) {
            reproductorMedia?.stop()
        }
        reproductorMedia?.reset()
        try {
            val idRecurso = R.raw.sonido_test
            reproductorMedia?.setDataSource( context: this, android.net.Uri.parse( uriString: "android.resource://$packageName/$idRecurso"))

            reproductorMedia?.setOnPreparedListener {
                fuenteAudioEstablecida = true
                mostrarMensaje( mensaje: "Audio local listo para reproducir")
            }
            reproductorMedia?.setOnErrorListener { _, _, _ ->
                fuenteAudioEstablecida = false
                mostrarMensaje( mensaje: "Error al cargar el audio local")
                false
            }
            mostrarMensaje( mensaje: "Cargando audio local...")
            reproductorMedia?.prepare()
        } catch (e: Exception) {
            mostrarMensaje( mensaje: "Error al acceder al archivo de audio local")
        }
    } catch (e: Exception) {
        mostrarMensaje( mensaje: "Error al cargar el audio local")
    }
}
```

- Los recursos deben estar en el directorio raw.

REPRODUCCIÓN REMOTA

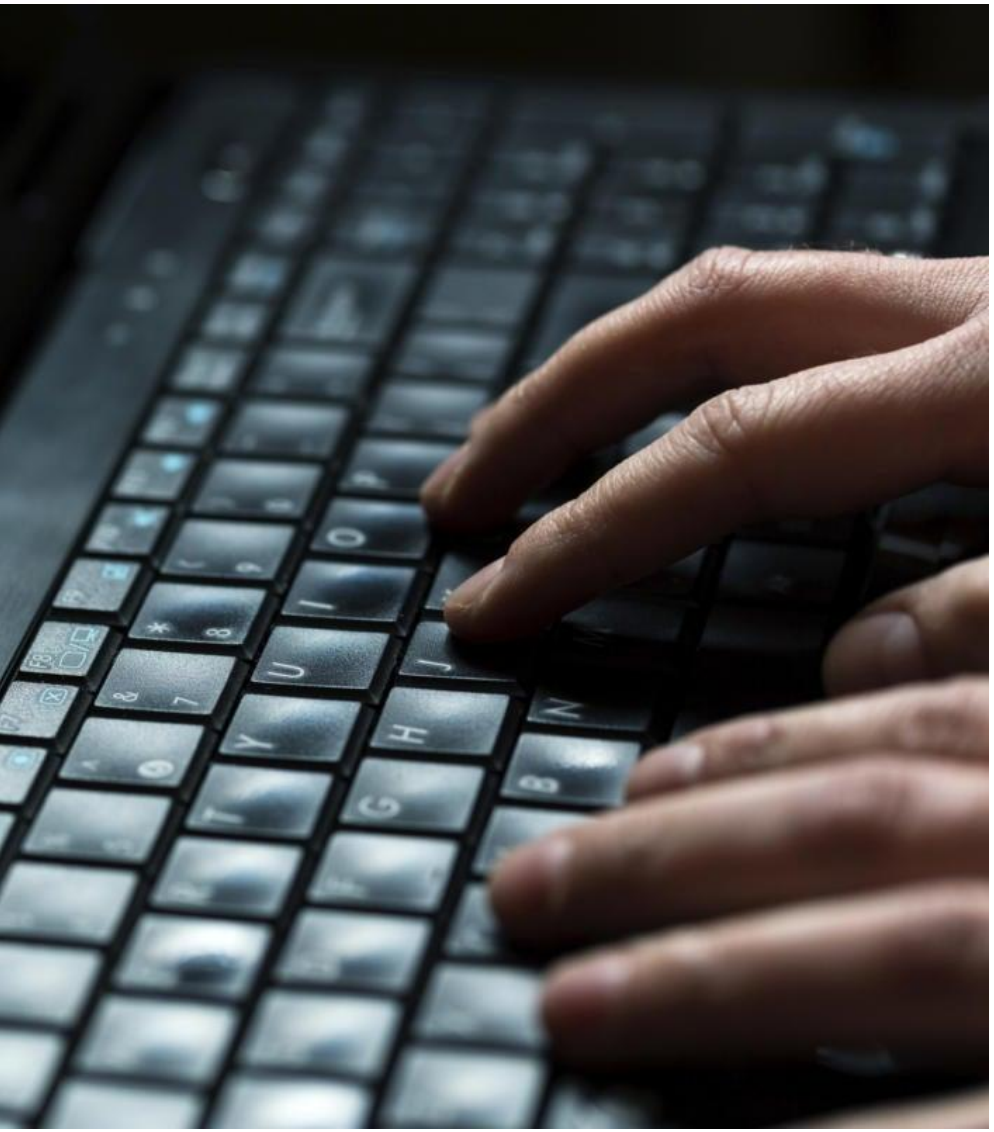
MEDIAPLAYER

- En este escenario los recursos están en un servidor remoto.
- Se recomienda url bajo protocolo https.
- Por el tipo de recurso, se recomienda audios cortos.

```
btnRemoto.setOnClickListener {
    try {
        if (!hayConexionInternet()) {
            mostrarMensaje( mensaje: "No hay conexión a internet")
            return@setOnClickListener
        }

        if (reproductorMedia?.isPlaying == true) {
            reproductorMedia?.stop()
        }
        reproductorMedia?.reset()

        val urlAudio = "https://tonosmovil.net/wp-content/uploads/tonosmovil.net_himno_champions_league.mp3"

        reproductorMedia?.setDataSource(urlAudio)
        reproductorMedia?.setOnPreparedListener {
            fuenteAudioEstablecida = true
            mostrarMensaje( mensaje: "Audio remoto listo para reproducir")
        }

        reproductorMedia?.setOnErrorListener { _, _, _ ->
            fuenteAudioEstablecida = false
            mostrarMensaje( mensaje: "Error al cargar el audio remoto")
            false
        }

        mostrarMensaje( mensaje: "Cargando audio remoto...")
        reproductorMedia?.prepareAsync()
```

LIVE CODING

TRABAJO A REALIZAR



- Tomando como referencia los contenidos visto en la sesión, elabore una aplicación que permita utilizar recursos locales como remotos en Android.

VALORACIONES **FINALES**

Comentarios sobre el tema.

1

MediaPlayer

Es la clase encargada de gestionar los recursos audio en una aplicación android.

2

ExoPlayer

Es un componente de terceros que permite generar reproductores de audio o video en Android.

3

Creativo

Saber distribuir y seleccionar los diferentes métodos y propiedades de una clase, permitirá al programador trabajar de forma rápida y eficiente

¿PREGUNTAS?

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

LUIS HUMBERTO RIVAS RODRÍGUEZ
INGENIERO EN SISTEMAS INFORMÁTICOS
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.