



Actividad 1 - Instalación Android Studio y Diseño de Aplicación

Desarrollo de Aplicaciones Móviles II

Ingeniería en Desarrollo de Software

Tutor: Eduardo Israel Castillo García

Alumno: Jusi Ismael Linares Gutiérrez

Fecha: 25/03/2024

Índice

Introducción	3
Descripción	3
Justificación	3
Desarrollo:	4
Interfaz	4
Codificación.....	4
Prueba de la aplicación	8
Conclusión	8
GitHub	8

Introducción

En la actualidad, el desarrollo de aplicaciones móviles se ha convertido en una parte fundamental del mundo digital, brindando a los usuarios una amplia gama de herramientas y entretenimiento directamente en la palma de sus manos. Con la llegada del Buen Fin, un evento de gran relevancia en el ámbito comercial, surge la oportunidad de ofrecer a los usuarios una experiencia única a través de una aplicación móvil que les permita acceder a una galería de ringtones.

Este proyecto tiene como objetivo principal desarrollar una aplicación móvil para dispositivos Android utilizando Android Studio, una de las plataformas más populares y ampliamente utilizadas para el desarrollo de aplicaciones en este sistema operativo. La aplicación resultante proporcionará a los usuarios una selección de 10 ringtones en formato .mp3, ofreciendo así una variedad de opciones para personalizar el sonido de sus dispositivos móviles.

Descripción

El contexto presenta la necesidad de desarrollar una aplicación móvil para Android en anticipación al evento conocido como El Buen Fin. La aplicación debe permitir a los usuarios acceder y manejar una galería de tonos de llamada, ofreciendo una experiencia práctica y atractiva. Esta solicitud refleja la importancia de adaptarse a las tendencias del mercado y aprovechar eventos específicos para ofrecer soluciones tecnológicas pertinentes.

La actividad requiere que un ingeniero en desarrollo de software instale y configure Android Studio, la plataforma de desarrollo de aplicaciones móviles para Android. Una vez configurado, el ingeniero debe crear una aplicación que incluya una selección de 10 tonos de llamada en formato .mp3. Esto implica conocimientos técnicos en programación y desarrollo de aplicaciones móviles, así como habilidades para diseñar una interfaz de usuario intuitiva y funcional.

En resumen, la actividad representa una oportunidad para aplicar habilidades técnicas en el desarrollo de software y crear una solución práctica para las necesidades de los usuarios durante El Buen Fin.

Justificación

La elección de desarrollar una aplicación móvil para manejar una galería de tonos de llamada en anticipación al evento de El Buen Fin presenta numerosas ventajas y justificaciones. En primer lugar, la creación de una aplicación móvil permite aprovechar el creciente número de usuarios de dispositivos móviles, quienes buscan soluciones prácticas y convenientes en sus teléfonos inteligentes. Al ofrecer una aplicación dedicada para descargar y gestionar tonos de llamada, se brinda una experiencia optimizada y centrada en el usuario.

Además, una aplicación móvil ofrece una interfaz intuitiva y fácil de usar, lo que facilita la navegación y la selección de tonos de llamada para los usuarios. Esto contribuye a una experiencia de usuario satisfactoria, lo que a su vez puede aumentar la retención de usuarios y la popularidad de la aplicación.

Desarrollo:

Interfaz



Codificación

Linear layout que nos sirve para contener los elementos que crearemos a continuación (botón de play, nombre del ringtone y botón de descarga)

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="39dp"
    android:layout_marginBottom="16dp"
    android:orientation="horizontal">
```

Código del botón de reproducción (en forma de imagen utilizando un icono) donde le asignamos id, tamaño y la dirección del icono a utilizar para el botón

```
<!-- Botón de reproducción -->
<ImageButton
    android:id="@+id/btnPlay1"
    android:layout_width="10dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:scaleType="fitCenter"
    android:src="@drawable/icon_play" />
```

Código para el nombre del archivo

```

<!-- Nombre del archivo MP3 -->
<TextView
    android:id="@+id/fileName1"
    android:layout_width="196dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_weight="2"
    android:text="Arashi No Saxophone 2" />

```

Código del botón de descarga (utilizando un icono) donde le asignamos id, tamaño y la dirección de nuestro icono de descarga

```

<!-- Botón de descarga -->
<ImageButton
    android:id="@+id/btnDownload1"
    android:layout_width="10dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:scaleType="fitCenter"
    android:src="@drawable/icon_download" />

```

Código que solicita permiso para escribir en el almacenamiento externo del dispositivo en el Android manifest

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Con este código se crea un reproductor de medios y carga un archivo de audio tphone_ringtone.mp3 para su reproducción. Luego, busca y asigna un botón de reproducción con el id btnPlay1 (que ira cambiando de numero para que sean los 10 ringtones) para permitir su control

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    mediaPlayer = MediaPlayer.create(context: this, R.raw.tphone_ringtone)

    val btnPlay1: ImageButton = findViewById(R.id.btnPlay1)
    val btnPlay2: ImageButton = findViewById(R.id.btnPlay2)
    val btnPlay3: ImageButton = findViewById(R.id.btnPlay3)
    val btnPlay4: ImageButton = findViewById(R.id.btnPlay4)
    val btnPlay5: ImageButton = findViewById(R.id.btnPlay5)
    val btnPlay6: ImageButton = findViewById(R.id.btnPlay6)
    val btnPlay7: ImageButton = findViewById(R.id.btnPlay7)
    val btnPlay8: ImageButton = findViewById(R.id.btnPlay8)
    val btnPlay9: ImageButton = findViewById(R.id.btnPlay9)
    val btnPlay10: ImageButton = findViewById(R.id.btnPlay10)
}

```

Con esta función nos encargaremos de asignar un listener al botón de reproducción para controlar la reproducción del archivo de audio

```
assignListener(btnPlay1, fileName: "arashi_no_saxophone2")
assignListener(btnPlay2, fileName: "beyond_journeys_end")
assignListener(btnPlay3, fileName: "chala_head_chala")
assignListener(btnPlay4, fileName: "chavo_del_8")
assignListener(btnPlay5, fileName: "fairy_fountain")
assignListener(btnPlay6, fileName: "gusty_garden_galaxy")
assignListener(btnPlay7, fileName: "jurassic_park_3_satellite_phone")
assignListener(btnPlay8, fileName: "pokemon_battle_gold_and_silver")
assignListener(btnPlay9, fileName: "tphone_ringtone")
assignListener(btnPlay10, fileName: "zelda_theme")
```

Este código define una función llamada assignListener que asigna un listener a un botón de imagen (ImageButton). Cuando se hace clic en el botón, verifica si el reproductor de medios está reproduciendo sonido. Si lo está, pausa la reproducción y cambia la imagen del botón a un ícono de reproducción. Si el reproductor no está reproduciendo, carga y reproduce un archivo de audio correspondiente al nombre de archivo proporcionado, y cambia la imagen del botón a un ícono de pausa.

```
private fun assignListener(button: ImageButton, fileName: String) {
    button.setOnClickListener { it: View!
        if (mediaPlayer.isPlaying) {
            mediaPlayer.pause()
            button.setImageResource(R.drawable.icon_play)
        } else {
            val resID = resources.getIdentifier(fileName, defType: "raw", packageName)
            mediaPlayer = MediaPlayer.create(context: this, resID)
            mediaPlayer.start()
            button.setImageResource(R.drawable.icon_pause)
        }
    }
}
```

Este código define una función llamada assignDownloadListener que asigna un listener a un botón de imagen para descargar un archivo de audio. Cuando se hace clic en el botón, el código busca el audio correspondiente al nombre proporcionado en los recursos de la aplicación. Luego, crea un directorio de salida en el almacenamiento externo público y guarda el archivo de audio descargado en este directorio con el nombre proporcionado.

```

private fun assignDownloadListener(button: ImageButton, fileName: String) {
    button.setOnClickListener { it: View?
        val resID = resources.getIdentifier(fileName, "raw", packageName)
        val audioResource = resources.openRawResource(resID)

        val outputDir = File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC), child: "RingtonesJusi")
        if (!outputDir.exists()) {
            outputDir.mkdirs()
        }
        val outputFile = File(outputDir, child: "$fileName.mp3")

        try {
            val outputStream = FileOutputStream(outputFile)
            val buffer = ByteArray( size: 1024)
            var length: Int
            while (audioResource.read(buffer).also { length = it } != -1) {
                outputStream.write(buffer, off: 0, length)
            }
            outputStream.flush()
            outputStream.close()
            audioResource.close()

            // Notificar al usuario que la descarga fue exitosa
            Toast.makeText( context: this, text: "Descarga completada: ${outputFile.absolutePath}", Toast.LENGTH_SHORT).show()
        } catch (e: IOException) {
            e.printStackTrace()
            // Notificar al usuario si ocurre un error durante la descarga
            Toast.makeText( context: this, text: "Error al descargar el archivo", Toast.LENGTH_SHORT).show()
        }
    }
}

```

Este código obtiene referencias a diez botones de imagen (ImageButton) en la interfaz de usuario mediante sus ID y luego asigna un listener de descarga a cada uno de ellos utilizando la función assignDownloadListener. Cada listener está configurado para descargar un archivo de audio específico cuando se hace clic en su botón correspondiente.

```

val btnDownload1: ImageButton = findViewById(R.id.btnDownload1)
val btnDownload2: ImageButton = findViewById(R.id.btnDownload2)
val btnDownload3: ImageButton = findViewById(R.id.btnDownload3)
val btnDownload4: ImageButton = findViewById(R.id.btnDownload4)
val btnDownload5: ImageButton = findViewById(R.id.btnDownload5)
val btnDownload6: ImageButton = findViewById(R.id.btnDownload6)
val btnDownload7: ImageButton = findViewById(R.id.btnDownload7)
val btnDownload8: ImageButton = findViewById(R.id.btnDownload8)
val btnDownload9: ImageButton = findViewById(R.id.btnDownload9)
val btnDownload10: ImageButton = findViewById(R.id.btnDownload10)

assignDownloadListener(btnDownload1, fileName: "arashi_no_saxophone2")
assignDownloadListener(btnDownload2, fileName: "beyond_journeys_end")
assignDownloadListener(btnDownload3, fileName: "chala_head_chala")
assignDownloadListener(btnDownload4, fileName: "chavo_del_8")
assignDownloadListener(btnDownload5, fileName: "fairy_fountain")
assignDownloadListener(btnDownload6, fileName: "gusty_garden_galaxy")
assignDownloadListener(btnDownload7, fileName: "jurassic_park_3_satellite_phone")
assignDownloadListener(btnDownload8, fileName: "pokemon_battle_gold_and_silver")
assignDownloadListener(btnDownload9, fileName: "tphone_ringtone")
assignDownloadListener(btnDownload10, fileName: "zelda_theme")

```

Prueba de la aplicación

Link del video donde se prueba la funcionalidad de la aplicación

<https://drive.google.com/file/d/12uN3n4g8brk7MAvI3cOpVA7cubN2valh/view?usp=sharing>

Conclusión

La creación de una aplicación móvil para gestionar una galería de ringtones en Android, como la que hemos desarrollado durante estos días, responde a la necesidad de proporcionar a los usuarios una plataforma fácil de usar para acceder y descargar una variedad de tonos de llamada durante eventos promocionales como El Buen Fin. A través de Android Studio, hemos podido diseñar una interfaz de usuario intuitiva que permite a los usuarios reproducir y descargar sus tonos de llamada favoritos con solo unos pocos clics.

Esta aplicación demuestra la versatilidad y la utilidad de las herramientas de desarrollo de software modernas para crear soluciones prácticas y atractivas que satisfagan las necesidades del usuario. Además, el proceso de configuración de Android Studio y la implementación de las funcionalidades requeridas subrayan la importancia de la familiaridad con las herramientas de desarrollo y la comprensión de los principios de diseño de interfaces de usuario móviles.

GitHub

https://github.com/JusiLinGu/Practicas_UMI