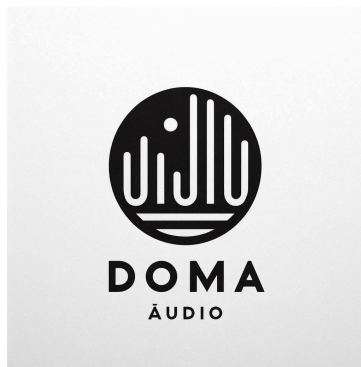


Desenvolvimento Full Stack

Aluno: Victor de A. Costa

Missão Prática I Nível 3 | Mundo 4 - 2025.1

RPG0025 - Lidando com sensores em dispositivos móveis



Para uma melhoria na eficiência e na comunicação interna, a empresa “Doma” quer desenvolver um aplicativo Wear OS para assistência aos funcionários que têm necessidades especiais, uma forma de solidificar a interação entre os mesmos. Assim, com os aplicativos wearables podem usar áudio para fornecer informações em tempo real, como leitura de mensagens de texto, notificações, lembretes e respostas a comandos de voz. Isso pode ser especialmente útil para pessoas com deficiência visual.

---> CÓDIGOS UTILIZADOS

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher"/>
```

```

        android:supportsRtl="true"
        android:theme="@style/Theme.DOMAAudio">

        <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>

    </application>

</manifest>

```

Audiohelper.xml

```

package com.example.domaaudio;

import android.content.Context;
import android.content.pm.PackageManager;
import android.media.AudioDeviceInfo;
import android.media.AudioManager;
import android.speech.tts.TextToSpeech;
import android.util.Log;

import java.util.Locale;

public class AudioHelper {

    public boolean isReady() {
        return isReady;
    }

    private static final String TAG = "AudioHelper";

    private final AudioManager audioManager;
    private final Context context;
    private TextToSpeech textToSpeech;
    private boolean isReady = false;

    public AudioHelper(Context context) {
        this.context = context;
    }

```

```

        this.audioManager = (AudioManager)
context.getSystemService(Context.AUDIO_SERVICE);

        this.textToSpeech = new TextToSpeech(context, status -> {
            if (status == TextToSpeech.SUCCESS) {
                Locale language = Locale.US;
                int result = textToSpeech.setLanguage(language);

                if (result == TextToSpeech.LANG_MISSING_DATA || result
== TextToSpeech.LANG_NOT_SUPPORTED) {
                    Log.e(TAG, "Idioma não suportado: " + language);
                } else {
                    isReady = true;
                    Log.i(TAG, "TextToSpeech inicializado com
sucesso");
                }
            } else {
                Log.e(TAG, "Falha ao inicializar o TextToSpeech");
            }
        });
    }

    public boolean isAudioDeviceConnected() {
        if
(!context.getPackageManager().hasSystemFeature(PackageManager.FEATURE_
AUDIO_OUTPUT)) {
            return false;
        }

        AudioDeviceInfo[] devices =
audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS);
        for (AudioDeviceInfo device : devices) {
            int type = device.getType();
            if (type == AudioDeviceInfo.TYPE_BLUETOOTH_A2DP ||
                type == AudioDeviceInfo.TYPE_BLUETOOTH_SCO ||
                type == AudioDeviceInfo.TYPE_WIRED_HEADPHONES ||
                type == AudioDeviceInfo.TYPE_WIRED_HEADSET ||
                type == AudioDeviceInfo.TYPE_USB_HEADSET ||
                type == AudioDeviceInfo.TYPE_USB_DEVICE ||
                type == AudioDeviceInfo.TYPE_LINE_ANALOG ||
                type == AudioDeviceInfo.TYPE_LINE_DIGITAL ||
                type == AudioDeviceInfo.TYPE_HDMI) {
                Log.i(TAG, "Dispositivo de áudio conectado: " +
device.getProductName());
            }
        }
    }

```

```

        return true;
    }
}
Log.i(TAG, "Nenhum dispositivo de áudio conectado");
return false;
}

public void speak(String text) {
    if (isReady) {
        textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null,
"UTTERANCE_ID");
    } else {
        Log.w(TAG, "TextToSpeech não está pronto ainda");
    }
}

public void shutdown() {
    if (textToSpeech != null) {
        textToSpeech.stop();
        textToSpeech.shutdown();
    }
}
}

```

MainActivity.xml

```

package com.example.domaaudio;

import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private AudioHelper audioHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

```

        audioHelper = new AudioHelper(this);

        Button speakButton = findViewById(R.id.speakButton);

        speakButton.setOnClickListener(v -> {
            if (!audioHelper.isAudioDeviceConnected()) {
                Toast.makeText(this, "Nenhum dispositivo de áudio
conectado. Conecte um fone ou caixa de som.",
Toast.LENGTH_SHORT).show();
            } else if (!audioHelper.isReady()) {
                Toast.makeText(this, "Inicializando o sistema de voz.
Por favor, aguarde...", Toast.LENGTH_SHORT).show();
            } else {
                audioHelper.speak("Seja bem-vindo ao DomaAudio!");
            }
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        if (audioHelper != null) {
            audioHelper.shutdown();
        }
    }
}

```

activity_main.xml

```

<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="8dp"
    android:background="@color/colorBackground">

    <TextView
        android:id="@+id/textAppName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```
    android:text="@string/app_name"
    android:textSize="16sp"
    android:textColor="@color/colorOnBackground"
    android:fontFamily="sans-serif-bold"
    android:textAlignment="center"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="16dp"/>
```

<ImageView

```
    android:id="@+id/imageLogo"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:src="@drawable/ic_volume_up"
    android:contentDescription="@string/app_name"
    app:layout_constraintTop_toBottomOf="@id/textAppName"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="8dp"/>
```

<TextView

```
    android:id="@+id/textDescription"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="@string/app_description"
    android:textSize="10sp"
    android:textColor="?attr/colorOnBackground"
    android:fontFamily="sans-serif-medium"
    android:textAlignment="center"
    android:layout_marginTop="8dp"
    app:layout_constraintTop_toBottomOf="@id/imageLogo"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
```

<Button

```
    android:id="@+id/speakButton"
    android:layout_width="wrap_content"
    android:layout_height="36dp"
    android:text="@string/button_speak"
    android:textSize="11sp"
    android:textAllCaps="false"
```

```

        android:drawableStart="@drawable/ic_volume_up"
        android:drawablePadding="4dp"
        android:backgroundTint="?attr/colorPrimary"
        android:textColor="?attr/colorOnPrimary"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"
        app:layout_constraintTop_toBottomOf="@id/textDescription"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:elevation="2dp"
        android:layout_marginTop="12dp"/>

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

strings.xml

```

<resources>
    <string name="app_name">D O M A Áudio</string>
    <string name="app_description">Sua Nova Experiência Sonora</string>
    <string name="button_speak">Falar</string>
    <string name="no_device">Nenhum dispositivo de áudio
conectado</string>
    <string name="tts_initializing">Inicializando voz...</string>
    <string name="welcome_message">Seja bem-vindo ao D O M A
Áudio!</string>
</resources>

```

styles.xml

```

<resources>

    <color name="colorBackground">#2196F3</color> <!-- branco -->
    <color name="colorOnBackground">#3F51B5</color> <!-- vermelho
vibrante -->
    <color name="colorPrimary">#E91E63</color> <!-- roxo -->
    <color name="colorOnPrimary">#3F51B5</color> <!-- branco -->

    <style name="Theme.DOMAAudio"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <item name="android:background">#2196F3</item>
        <item name="android:textColor">#0000FF</item>
        <item name="colorPrimary">#3F51B5</item>
        <item name="colorOnPrimary">@color/colorBackground</item>
    </style>

```

```

        <item name="android:textColorPrimary">#3F51B5</item>
        <item
name="android:buttonStyle">@style/ButtonStyle.DOMAAudio</item>
    </style>

    <style name="ButtonStyle.DOMAAudio"
parent="Widget.MaterialComponents.Button">
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">40dp</item>
        <item name="android:textSize">12sp</item>
        <item name="android:textAllCaps">false</item>
        <item name="android:paddingStart">12dp</item>
        <item name="android:paddingEnd">12dp</item>
        <item name="android:backgroundTint">#3F51B5</item>
        <item name="android:textColor">#3F51B5</item>
        <item name="android:elevation">2dp</item>
    </style>
</resources>

```

build.gradle.kts

```

plugins {
    id("com.android.application")
}

android {
    namespace = "com.example.domaaudio" // Usar "=" no Kotlin DSL
    (build.gradle.kts)
    compileSdk = 33

    defaultConfig {
        applicationId = "com.example.domaaudio"
        minSdk = 30
        targetSdk = 33
        versionCode = 1
        versionName = "1.0"
    }

    buildTypes {
        release {
            isMinifyEnabled = false

```



```

        proguardFiles (
            getDefaultProguardFile("proguard-android-optimize.txt"),
            "proguard-rules.pro"
        )
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

dependencies {
    implementation("com.google.android.material:material:1.2.0")
    implementation("androidx.appcompat:appcompat:1.3.0")
    implementation("androidx.constraintlayout:constraintlayout:2.0.4")
}

```

Imagens



