

Documentação do Trabalho Prático

ListaDeTarefas – Gerenciamento Inteligente de Áudio (Wear OS)

Detalhe Informação

Disciplina Desenvolvimento Full-Stack

Aluno Pedro Martins

Data 29 de Novembro de 2025

1 Introdução e Objetivo

Este trabalho prático consiste no desenvolvimento de uma aplicação para Wear OS utilizando **Text-to-Speech (TTS)** para fornecer comandos e retorno de áudio ao usuário.

O foco principal foi **priorizar o uso de dispositivos Bluetooth** conectados ao relógio e apenas utilizar o alto-falante integrado caso nenhum fone esteja disponível.

O projeto evidencia o uso de recursos nativos do hardware, integração com APIs do sistema e boas práticas de organização em Kotlin.

2 Arquitetura e Implementação Técnica

A aplicação foi dividida em duas principais responsabilidades:

- **MainActivity.kt** — Implementação do TTS e lógica principal da interface
 - **AudioHelper.kt** — Verificação e gerenciamento das saídas de áudio disponíveis
-

2.1. Text-to-Speech (TTS)

- Inicialização realizada em `onInit()`
- Idioma configurado para **Português do Brasil** (`Locale("pt", "BR")`)
- Utilização de **QUEUE_FLUSH** para sobrepor mensagens anteriores

O método `speakMessage(text: String)` centraliza o envio das falas para o mecanismo de voz.

2.2. Gerenciamento de Saída de Áudio

O gerenciamento faz uso do AudioManager para detectar dinamicamente:

- 1 **Bluetooth A2DP** (prioridade máxima)
- 2 **Alto-falante embutido** como alternativa
- 3 Caso não haja saída → registro em log

A verificação é feita por métodos dedicados que consultam dispositivos ativos no sistema.

2.3. Detecção Automática de Conexões

Por meio do AudioDeviceCallback, o aplicativo reconhece quando:

- Fones Bluetooth são conectados — com feedback imediato por voz
- Dispositivos são desconectados — registrando evento no log

Permitindo assim uma experiência reativa e mais eficiente para o usuário.

3 Dificuldades Encontradas e Soluções

Desafio	Solução Aplicada
Travamentos e lentidão ao abrir o emulador	Testes priorizados somente com o Wear OS ativo
Erros de referência (Unresolved reference)	Reorganização das classes e imports corretos
Erros de sintaxe (Missing '})'	Revisão completa do código e fechamento correto de escopos
Conflito ao enviar para o GitHub (unrelated histories)	Reinicialização do repositório local e novo push via terminal

Essas resoluções contribuíram para a evolução contínua do funcionamento do aplicativo.

4 Conclusão

Os objetivos do trabalho foram plenamente atingidos:

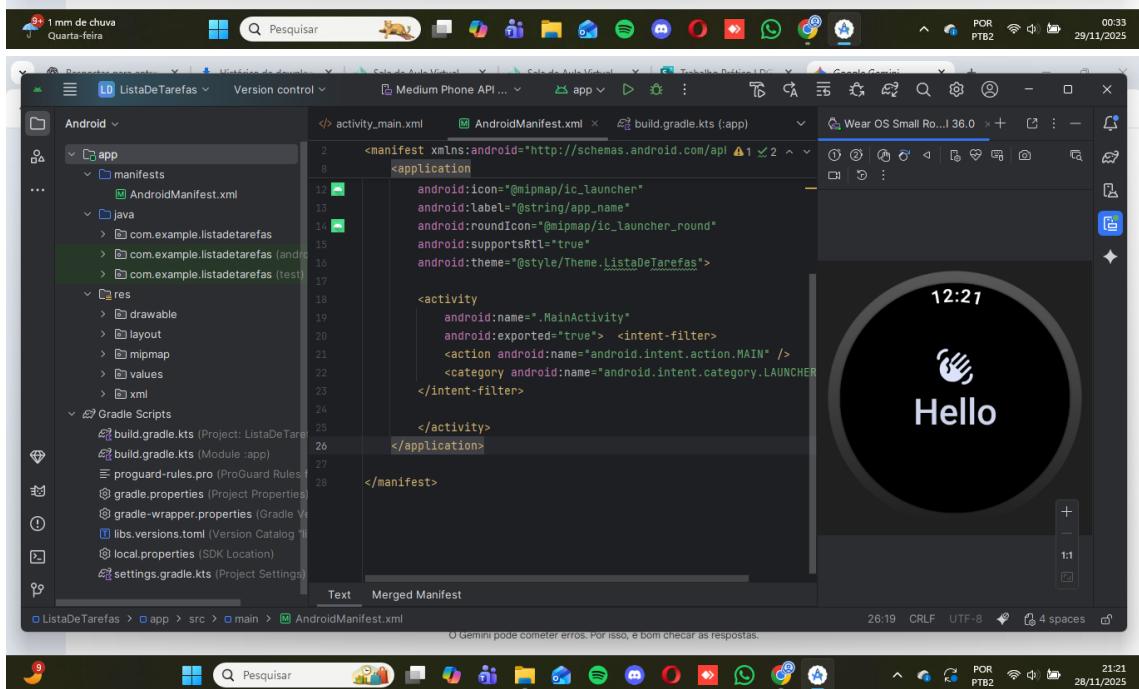
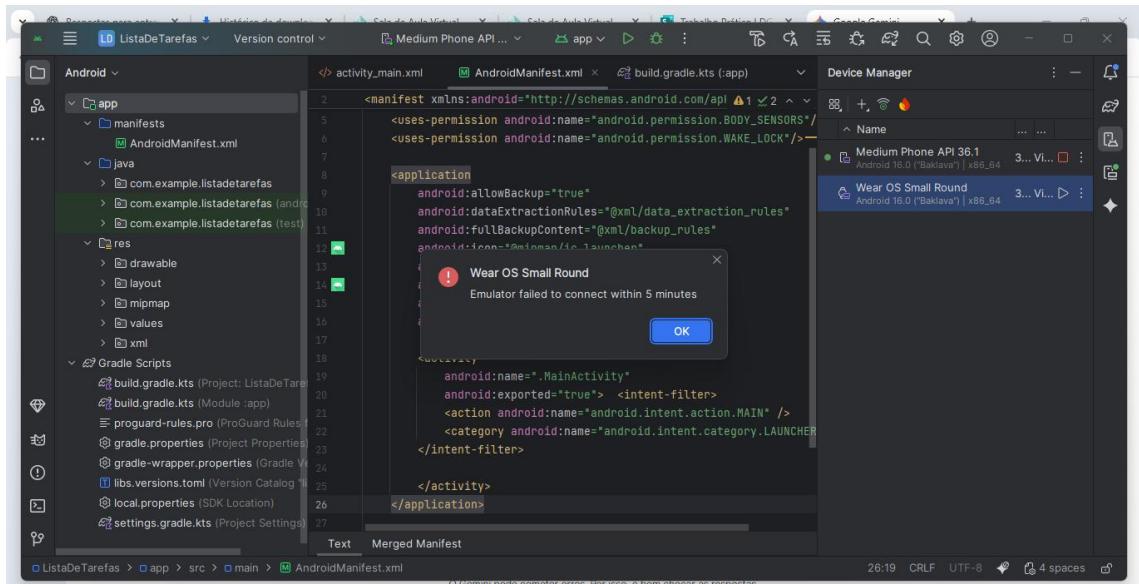
- ✓ Comunicação por voz funcional com Text-to-Speech
- ✓ Controle de hardware para priorização de áudio via Bluetooth
- ✓ Modularidade do código e monitoramento de dispositivos em tempo real
- ✓ Resolução de falhas práticas durante o desenvolvimento

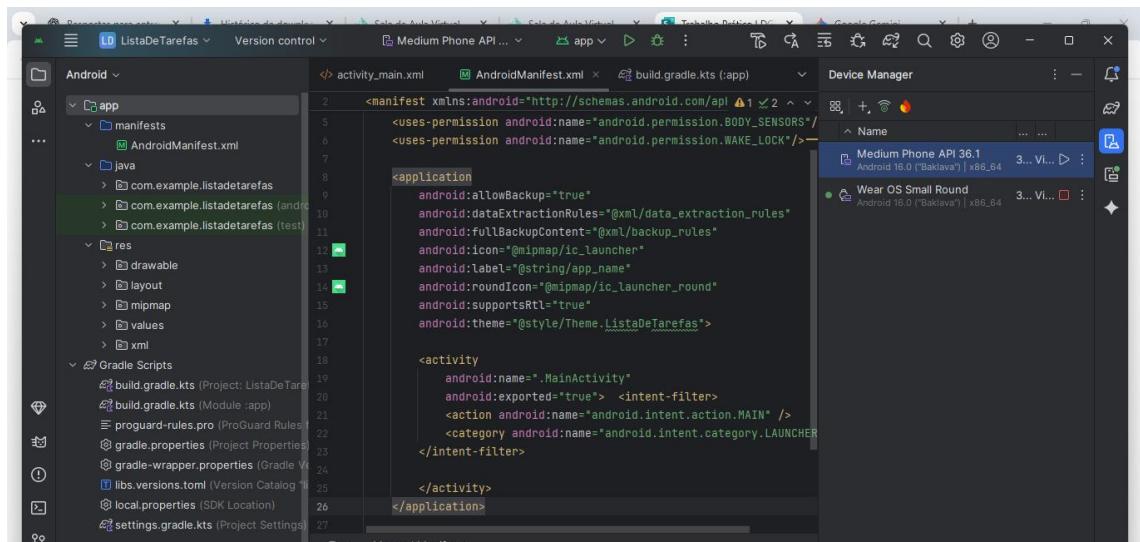
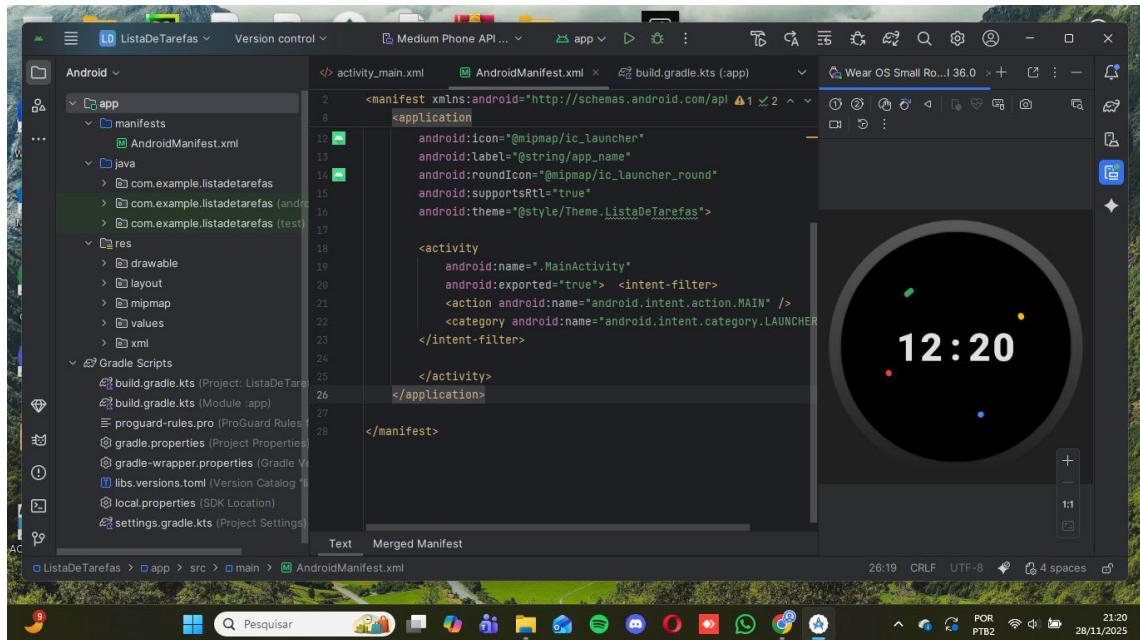
O projeto demonstra conhecimento de APIs Wear OS e boas práticas de programação para dispositivos vestíveis.

5 Evidências de Progresso

A seguir serão inseridas as **capturas de tela referentes aos erros enfrentados e progressos realizados ao longo das microatividades**, comprovando:

- Evolução do código
- Execução no emulador
- Ajustes e correções aplicadas





The screenshot shows the Android Studio interface with the Device Manager open. A 'Medium Phone API 36.1' device is selected. The code editor displays the `AndroidManifest.xml` file, which includes the following manifest code:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.BODY_SENSORS"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ListaDeTarefas">

        <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>
```

