



Estácio

Campus: Polo Centro II - Guarulhos - SP

Curso: Desenvolvimento full stack

Disciplina: RPG0025 - Lidando com sensores em dispositivos móveis

Turma: 2025.1

Aluno: Pedro Wilson Araújo Avilar

Matrícula: 2023 0916 8251

- **Título**

Missão Prática | Mundo 4 | Nível 3

- **Material de apoio**

<https://sway.cloud.microsoft/s/EmYWGfbAKD99npjX/embed>

- **Objetivo**

Esta missão prática tem como objetivo desenvolver um aplicativo Wear OS para a empresa, fictícia, chamada Doma, voltado a assistência a funcionários com necessidades especiais por meio de interações auditivas e comandos de voz. O projeto demonstra a aplicação prática de tecnologias wearables para melhorar a eficiência e a inclusão no ambiente de trabalho, oferecendo suporte em situações de risco, como a leitura de notificações recebidas pelo dispositivo e a detecção de acidentes (quedas do usuário ou do dispositivo).

- **Repositório git**

<https://github.com/PedroAvilar/DomaAssistente>

- **Tecnologias utilizadas**

- **Android Studio:** IDE oficial para desenvolvimento Android, usada para codificação, depuração e execução no emulador Wear OS.
- **Kotlin:** Linguagem principal do projeto, utilizada para toda a lógica de negócio e UI com Compose.
- **Jetpack Compose for Wear OS:** Biblioteca de UI declarativa para criar layouts escaláveis e responsivos em dispositivos wearables.
- **Android SDK:** Usado para compilação e compatibilidade.
- **Text-to-Speech:** API nativa para converter texto em fala.
- **SpeechRecognizer (Android Speech API):** Serviço de reconhecimento de voz para captura e interpretação de comandos falados.
- **SensorManager e Acelerômetro:** API de sensores do Android para detectar movimentos bruscos e quedas.
- **AudioManager & AudioDeviceInfo:** Classes para listar e monitorar dispositivos de saída de áudio conectados dinamicamente.
- **MediaPlayer (emergency_alarm.mp3):** Usado para reproduzir um alarme sonoro em situações de emergência.
- **NotificationListenerService:** Serviço para interceptar notificações do sistema e encaminhar ao TTS.
- **Emulador Wear OS (API 34):** Ambiente de simulação de smartwatch para teste de execução da aplicação e layout da interface.
- **Smartphone Android (14):** Execução do apk da aplicação para testes e validações das funcionalidades, permissões e uso do sensor.

- **Funcionalidades implementadas**

- **Listagem de dispositivos de áudio:** Identifica e exibe dinamicamente alto-falante embutido, fones Bluetooth e outras saídas de áudio.
- **Leitura de notificações:** Intercepta notificações do sistema e as converte em voz via TTS para acessibilidade.
- **Comandos de voz:** Reconhece comandos falados como “mensagens”, “dispositivo”, “status” e “ajuda”, respondendo adequadamente.
- **Deteção de quedas:** Monitora o acelerômetro para detectar movimentos bruscos e confirma se o usuário está bem.
- **Resposta a quedas:** Aguarda resposta do usuário por voz, caso o usuário diga que não está bem ou não forneça resposta, dispara alarme de emergência.
- **Alarme de emergência:** Reproduz som de alarme contínuo, sendo interrompido por comando de voz.
- **Feedback de permissões:** Orienta o usuário por voz para conceder permissões de notificação e microfone, com reavaliação imediata ao voltar.
- **Interface Wear OS:** Para listagens e interação por toque.

- **Estrutura dos arquivos e pastas**

Abaixo está a estrutura de diretórios e arquivos principais:

DomaAssistente/

```
— app/
  — src/main
    — java/com/doma/assistente/
      — audio/
        — AudioOutoutHelper.kt
      — emergency/
        — EmergencyHelper.kt
      — notification/
        — DomaNotificationListener.kt
      — sensors/
        — MotionDetector.kt
      — tts/
        — TextToSpeechHelper.kt
      — ui/
        — AppRoot.kt
        — MainScreen.kt
      — voice/
        — SpeechRecognitionHelper.kt
        — VoiceCommandProcessor.kt
      — MainActivity.kt
    — res/raw/
      — emergency_alarm.mp3
    — AndroidManifest.xml
  — build.gradle.kts
— build.gradle.kts
```

- **Descrição dos arquivos**

- **AudioOutoutHelper.kt**

app\src\main\java\com\doma\assistente\audio\AudioOutputHelper.kt

Helper para aquisição e mapeamento de dispositivos de saída de áudio conectados, incluindo alto-falante e fones.

- **EmergencyHelper.kt**

app\src\main\java\com\doma\assistente\emergency\EmergencyHelper.kt

Controla reprodução de alarme sonoro em loop.

- **DomaNotificationListener.kt**

app\src\main\java\com\doma\assistente\notification\DomaNotificationListener.kt

Extensão de NotificationListenerService, do sistema android, que intercepta notificações, armazena histórico e encaminha mensagens para leitura por TTS.

- **MotionDetector.kt**

app\src\main\java\com\doma\assistente\sensors\MotionDetector.kt

Implementa SensorEventListener, do sistema, para monitorar o acelerômetro, detectar movimentos bruscos (quedas), acionar confirmação por voz e alarme do EmergencyHelper.

- **TextToSpeechHelper.kt**

app\src\main\java\com\doma\assistente\tts\TextToSpeechHelper.kt

Wrapper para inicializar e controlar o mecanismo de Text-to-Speech do Android, oferecendo métodos de fala e callback de conclusão.

- **AppRoot.kt**

app\src\main\java\com\doma\assistente\ui\AppRoot.kt

Composable que inicializa o TTS e gerencia a apresentação da tela de carregamento ou erro antes de exibir a MainScreen.

- **MainScreen.kt**

app\src\main\java\com\doma\assistente\ui\MainScreen.kt

Composable contendo a interface do usuário principal, gerenciando permissões do app, incluindo listagem de dispositivos e lógica de toque para comandos de voz.

- **SpeechRecognitionHelper.kt**

app\src\main\java\com\doma\assistente\voice\SpeechRecognitionHelper.kt

Helper que encapsula a API SpeechRecognizer, definindo RecognitionListener para capturar resultados de voz e erros.

- **VoiceCommandProcessor.kt**

app\src\main\java\com\doma\assistente\voice\VoiceCommandProcessor.kt

Lógica de processamento de comandos de voz, mapeando frases reconhecidas a ações do sistema, como leitura de notificações e controle do alarme.

- **MainActivity.kt**

app\src\main\java\com\doma\assistente\MainActivity.kt

Activity principal que inicia a composição do UI e orquestra os componentes principais.

- **emergency_alarm.mp3**

app\src\main\res\raw\emergency_alarm.mp3

Arquivo mp3 usado para o alarme sonoro em caso de movimentos bruscos (queda) pelo arquivo EmergencyHelper.

- **AndroidManifest.xml**

app\src\main\AndroidManifest.xml

Declara permissões (notificação, áudio, Bluetooth) e registra o DomaNotificationListener como serviço de escuta de notificações.

- **build.gradle.kts**

app\build.gradle.kts

Configuração do módulo app, incluindo plugins, dependências e opções de compilação para Wear OS e Compose.

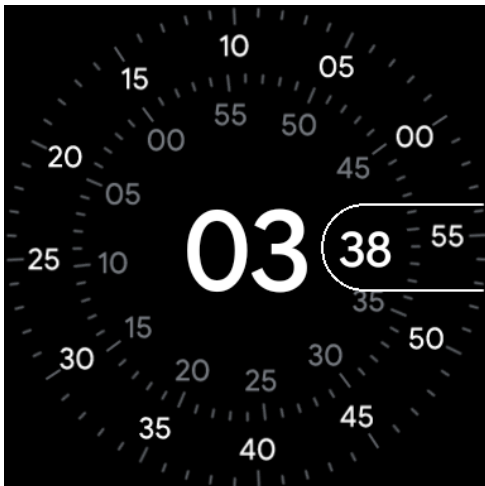
- **build.gradle.kts**

(root)

Configuração de versões de plugin.

- **Capturas de tela no emulador Wear OS Small Round no Android Studio**

- **Tela inicial do emulador**



- **Iniciando o TTS e falha ao iniciar**

Aguarde. Tentando
iniciar o Text To
Speech.

Falha ao iniciar o TTS.
Verifique permissões
ou reinicie o app.

- **Solicitando permissão ao microfone**

Allow Doma
Assistente to
record audio?

While using app

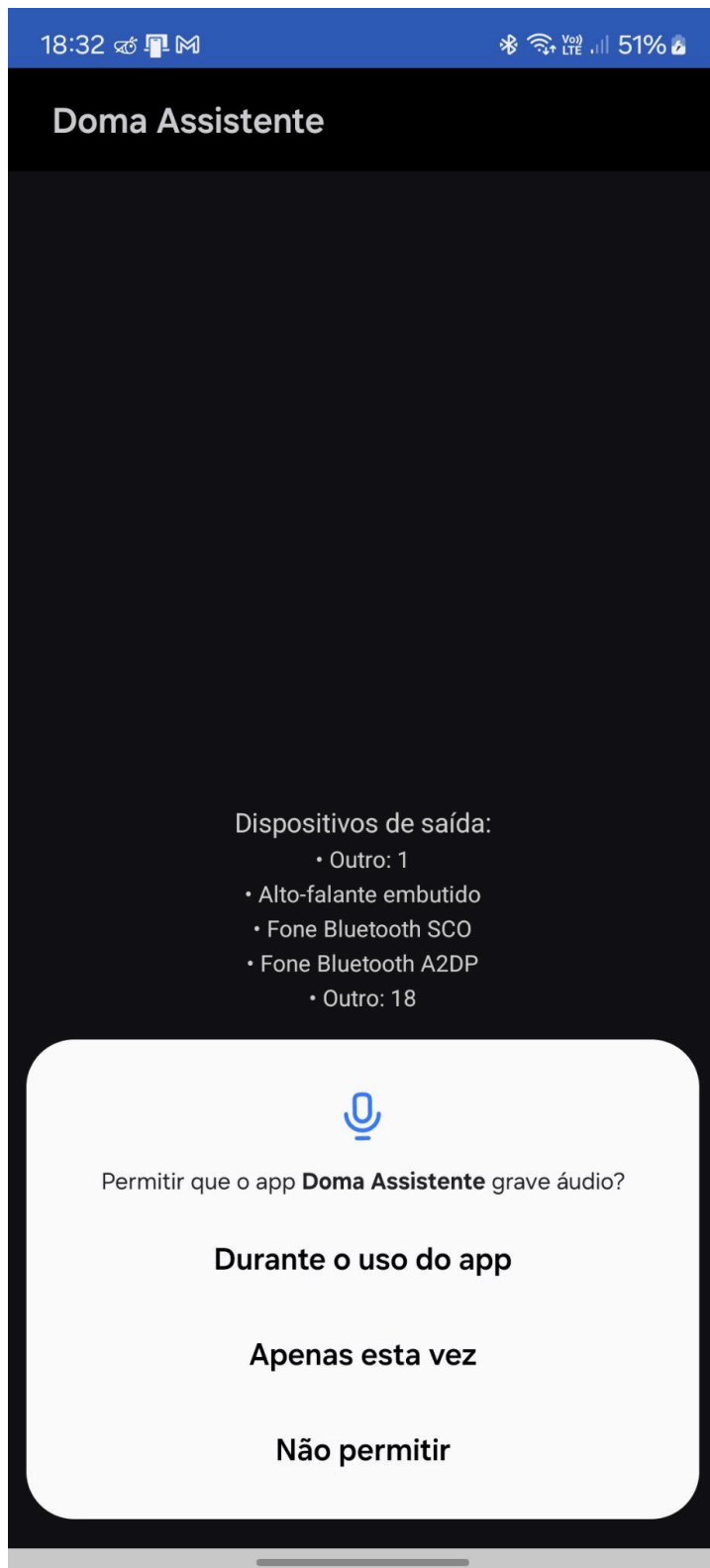
Only this time

- **Aplicativo em funcionamento**

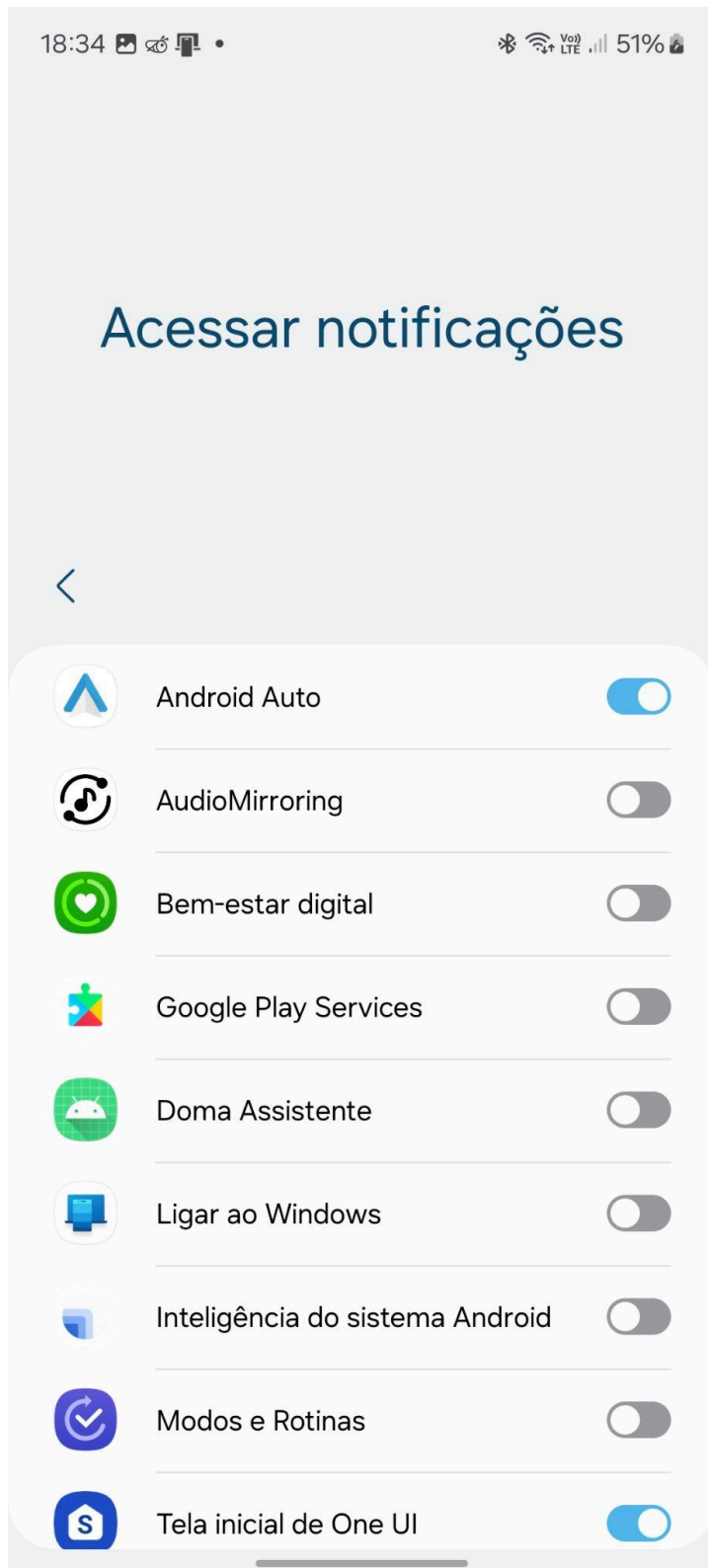
Dispositivos de saída:

- Alto-falante embutido
- Outro: 18

- Capturas de tela no smartphone Android
 - Solicitando permissão ao microfone



- **Solicitando permissão às notificações**



○ **Aplicativo em funcionamento**

