



## **Doma Wear OS**

**Matheus José Ribeiro de Moura 2023 0713 6158**

**Polo Rua Tereza - Petrópolis - RJ**

**Lidando Com Sensores em Dispositivos Móveis – Turma 9001 – 2025.1 FLEX**

### **Objetivo da Prática**

A prática tem como objetivo criar apps WEAR OS acessíveis para pessoas com necessidades especiais. O doma wear os foi pensado para a empresa DOMA aumentar o sentimento de pertencimento dos funcionários com a empresa.

O app é feito 100% em kotlin usando recursos nativos para gerar uma experiência completamente acessível aos usuários.

## Visão Geral

O DomaWear é um aplicativo projetado para dispositivos Wear OS, com foco em acessibilidade para funcionários com deficiência visual. Ele fornece instruções de navegação dentro de uma empresa e emite alertas de emergência com feedback sonoro e de voz.

---

## Estrutura do Projeto

### 1. Arquivo `activity_main.xml`

Esse é o layout principal da interface. Ele utiliza um `ConstraintLayout` contendo um `LinearLayout` centralizado, com os seguintes elementos:

- `TextView`: Exibe um texto descritivo inicial.
- `android:text`: Mensagem de instrução inicial.
- `android:contentDescription`: Descrição para leitores de tela.
- `Button`: Botão que aciona a leitura em voz alta das instruções da empresa.
- `android:contentDescription`: Indica função acessível do botão.

*Atributos importantes:*

- `android:importantForAccessibility="yes"`: Garante que os elementos sejam lidos por tecnologias assistivas.
- `android:background="#000000"`: Fundo preto para contraste.

### 2. Arquivo `MainActivity.kt`

Classe principal responsável pela lógica do aplicativo.

Imports e funções:

- `android.speech.tts.TextToSpeech`: Leitura em voz alta do conteúdo textual.
- `android.media.MediaPlayer`: Toca sons (áudio de alerta).
- `android.os.Handler`, `Looper`: Controla execução de alertas com delay.
- `AlertDialog`: Exibe pop-up de emergência.

*Variáveis principais:*

- `tts`: Instância do `TextToSpeech`.
- `mediaPlayer`: Toca o som de alerta.
- `handler`: Manipula a fila de alertas agendados.
- `ttsInitialized`: Indica se o TTS foi inicializado com sucesso.

- ``welcomeMessage``: Mensagem falada ao abrir o app.
- ``instructionText``: Texto com a localização dos setores

### **Métodos:**

- ``onCreate()``: Inicializa o TTS, carrega o áudio de alerta, configura o botão e aciona a primeira leitura da mensagem de boas-vindas.
  - ``startEmergencyAlerts()``: Cria um intervalo aleatório entre 45 a 60 segundos para acionar alertas.
  - ``triggerEmergency()``: Exibe um ``AlertDialog`` com som e voz, que pode ser fechado tocando na tela.
  - ``onInit()``: Inicializa o TTS com o idioma ``pt-BR``.
  - ``onDestroy()``: Libera os recursos do TTS e do mediaPlayer.
- 

### **Funcionalidades de Acessibilidade**

- Mensagem falada inicial explicando o propósito do aplicativo.
  - Instruções sobre localização de setores lidas por comando do botão central.
  - Alertas de emergência em áudio e texto falado, exibidos em intervalos aleatórios.
- 

### **Considerações Finais**

Este aplicativo oferece suporte acessível essencial para navegação em ambientes corporativos por usuários com deficiência visual. Seu funcionamento depende do TextToSpeech do Android e da configuração de idioma correta no dispositivo.