



**UNIVERSIDADE ESTÁCIO DE TAGUATINGA BRASÍLIA/DF**  
**Tecnologia em Desenvolvimento Full Stack RPG0025 - Lidando com**  
**sensores em dispositivos móveis**

**Aluno:** Yan Silva Sales

**Matrícula:** 202302573592

**GitHub:** <https://github.com/YanSales/My-App-3>

## **RELATÓRIO DA MISSÃO PRÁTICA**

### **INTRODUÇÃO**

Este documento detalha o desenvolvimento de um aplicativo Wear OS projetado para melhorar a comunicação e a eficiência entre funcionários com deficiência na empresa "Doma". O aplicativo utiliza recursos de áudio para oferecer suporte, como leitura de mensagens e notificações, resposta a comandos de voz e emissão de alertas de segurança.

### **OBJETIVOS**

O objetivo principal do projeto é criar um aplicativo Wear OS acessível que auxilie funcionários com deficiências visuais, proporcionando uma interface de comunicação eficaz e segura.

### **REQUISITOS Requisitos Funcionais**

1. **Leitura de Mensagens:** O aplicativo deve ser capaz de ler mensagens de texto em voz alta.
2. **Notificações por Áudio:** Notificar os usuários sobre eventos importantes através de alertas sonoros.
3. **Respostas a Comandos de Voz:** O aplicativo deve responder a comandos de voz para executar ações específicas.
4. **Alertas de Segurança:** Emitir notificações sonoras em caso de emergências, alertas de tempestades ou outras situações críticas.

### **Requisitos Não Funcionais**

1. Desempenho: O aplicativo deve operar de forma eficiente em dispositivos Wear OS sem causar lentidão.
2. Usabilidade: A interface deve ser simples e intuitiva, facilitando o uso por pessoas com deficiências visuais.
3. Conectividade: O aplicativo deve gerenciar corretamente as conexões de áudio via Bluetooth.

## **ARQUITETURA DO SISTEMA**

### **Componentes Principais**

1. Gerenciador de Áudio: Controla as saídas de áudio disponíveis e gerencia a reprodução de som no dispositivo.
2. Módulo de Conectividade Bluetooth: Facilita a conexão com fones de ouvido Bluetooth e outros dispositivos de áudio.
3. Interface de Usuário (UI): Projetada para ser acessível e facilitar a navegação com comandos de voz.

## **CONFIGURAÇÃO DO AMBIENTE Ferramentas Necessárias**

- Editor de Texto/IDE: Android Studio.
- Flutter SDK: Para desenvolvimento em Flutter.
- Android Studio: Para desenvolvimento e emulação em plataformas Android.
- Simulador Android: Para testar o aplicativo.
- Navegador Web: Google Chrome.

### **Passos de Configuração**

1. Instalação do Flutter SDK: Baixe e configure o Flutter SDK.
2. Configuração do Android Studio: Configure o Android Studio com as SDKs necessárias para Wear OS.
3. Configuração do Simulador: Prepare um simulador para Wear OS ou conecte um dispositivo wearable real.

## IMPLEMENTAÇÃO Saídas de Áudio

A implementação começa com a verificação das saídas de áudio disponíveis, como alto-falantes integrados ou fones de ouvido Bluetooth. Utilizamos o `AudioManager` para gerenciar as saídas de áudio.

```
import android.content.Context import
android.media.AudioDeviceInfo import
android.media.AudioManager import
android.content.pm.PackageManager

class AudioHelper(context: Context) { private val audioManager:
    AudioManager = context.getSystemService(Context.AUDIO_SERVICE) as
    AudioManager

    fun audioOutputAvailable(type: Int): Boolean {
        if
        (!context.packageManager.hasSystemFeature(PackageManager.FEATURE_AUDIO_OUTPUT))
        { return false
          } return
        audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS).any { it.type ==
        type } }
    }
```

## Detecção Dinâmica de Dispositivos de Áudio

O aplicativo pode detectar dinamicamente a adição ou remoção de dispositivos de áudio, como fones de ouvido Bluetooth, utilizando callbacks.

```
audioManager.registerAudioDeviceCallback(object : AudioDeviceCallback() {
    override fun onAudioDevicesAdded(addedDevices: Array<out AudioDeviceInfo>?)
    { super.onAudioDevicesAdded(addedDevices) if
        (audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2DP)) {
            // Um fone de ouvido Bluetooth acabou de ser conectado
        }
    }

    override fun onAudioDevicesRemoved(removedDevices: Array<out
    AudioDeviceInfo>?) { super.onAudioDevicesRemoved(removedDevices) if
        (!audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2DP)) {
            // Um fone de ouvido Bluetooth não está mais conectado
        }
    }
}, null)
```

## Facilitação da Conexão Bluetooth

Para facilitar a conexão com dispositivos Bluetooth, o aplicativo pode

redirecionar o usuário diretamente para as configurações de Bluetooth.

```
val intent = with(Intent(Settings.ACTION_BLUETOOTH_SETTINGS)) {  
    addFlags(Intent.FLAG_ACTIVITY_NEW_TASK or  
    Intent.FLAG_ACTIVITY_CLEAR_TASK) putExtra("EXTRA_CONNECTION_ONLY", true)  
    putExtra("EXTRA_CLOSE_ON_CONNECT", true)  
    putExtra("android.bluetooth.devicepicker.extra.FILTER_TYPE", 1)  
}  
startActivity(intent)
```

## Reprodução de Áudio

Após a detecção de uma saída de áudio adequada, o áudio pode ser reproduzido da mesma forma que em dispositivos móveis.

## Uso de Alto-falantes

Para dispositivos Wear OS com alto-falantes, funcionalidades de áudio podem ser incorporadas para melhorar a experiência do usuário, como notificações sonoras, instruções de voz para exercícios, etc.

## TESTES

1. Testes Funcionais: Verificar se todas as funcionalidades do aplicativo estão operando corretamente.

2. Testes de Usabilidade: Garantir que a interface é acessível e fácil de usar por pessoas com necessidades especiais.

3. Testes de Desempenho: Avaliar a eficiência do aplicativo em simuladores de dispositivos Wear OS.

## CONCLUSÃO

Este projeto visa criar um aplicativo Wear OS que fornece suporte eficaz e acessível a funcionários com deficiência na empresa "Doma". Com a conclusão bem-sucedida, espera-se uma melhoria significativa na comunicação na segurança interna, demonstrando o potencial dos wearables em soluções inclusivas no local de trabalho.