

Assistente Wear OS - Documentação Completa

Descrição do Projeto

O **Assistente Wear OS** é um aplicativo projetado para dispositivos **Wear OS** que proporciona uma comunicação eficaz e assistência para funcionários com necessidades especiais. O app permite a **leitura de mensagens e notificações em voz alta**, responde a comandos de voz e fornece **alertas de segurança e instruções** através de áudio.

O projeto visa melhorar a eficiência e a comunicação interna na empresa **Doma**, utilizando tecnologias wearables para criar soluções acessíveis e inclusivas no local de trabalho.

Principais Funcionalidades

- **Leitura de Mensagens e Notificações:** O aplicativo lê mensagens em voz alta para os funcionários.
 - **Comandos de Voz:** Permite interagir com o app por meio de comandos de voz.
 - **Alertas de Segurança:** Fornece notificações para emergências, tempestades e outras situações críticas.
 - **Facilidade na Conexão Bluetooth:** Permite conectar fones de ouvido diretamente pelo aplicativo.
 - **Compatibilidade com Saídas de Áudio:** Suporte para alto-falante integrado e fones de ouvido Bluetooth.
-

Tecnologias Utilizadas

Frontend

- **Flutter SDK** (para desenvolvimento do aplicativo)
- **Dart** (linguagem de programação utilizada no Flutter)
- **Audioplayers** (biblioteca para reprodução de áudio)

Backend

- **MethodChannel** (para comunicação nativa com Android)
- **Android AudioManager API** (para detectar e manipular dispositivos de áudio)

Ambiente de Desenvolvimento

- **VS Code** ou **Android Studio**
- **Wear OS Emulator** ou **Dispositivo Real Wear OS**
- **Flutter SDK** configurado no sistema

Configuração do Ambiente

1 Instalar Flutter e Dependências

Caso ainda não tenha o **Flutter** instalado, siga os passos:

```
# Baixar o Flutter SDK
<https://flutter.dev/docs/get-started/install>

# Verificar se o Flutter está configurado corretamente
flutter doctor
```

2 Criar o Projeto

```
flutter create wearable_assistant
cd wearable_assistant
code .
```

3 Adicionar Dependências

No arquivo `pubspec.yaml`, adicione:

```
dependencies:
  flutter:
    sdk: flutter
  audioplayers: ^0.20.1
```

E execute:

```
flutter pub get
```



Implementação do Código



Verificação de Saídas de Áudio

Arquivo: `lib/audio_helper.dart`

```
import 'package:flutter/services.dart';

class AudioHelper {
  static const MethodChannel _channel = MethodChannel('audio_helper');

  static Future<bool> isAudioOutputAvailable(int type) async {
    final bool available = await _channel.invokeMethod('audioOutputAvailable', {"type": type});
    return available;
  }
}
```



Reprodução de Áudio

Arquivo: `lib/main.dart`

```
import 'package:flutter/material.dart';
import 'package:audioplayers/audioplayers.dart';
import 'audio_helper.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

```

        debugShowCheckedModeBanner: false,
        home: AudioScreen(),
    );
}
}

class AudioScreen extends StatefulWidget {
    @override
    _AudioScreenState createState() => _AudioScreenState();
}

class _AudioScreenState extends State<AudioScreen> {
    final AudioPlayer _audioPlayer = AudioPlayer();

    void playAudio() async {
        await _audioPlayer.play('<https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3>');
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text("Wear OS Assistente")),
            body: Center(
                child: ElevatedButton(
                    onPressed: playAudio,
                    child: Text("Reproduzir Áudio"),
                ),
            ),
        );
    }
}

```

Configuração Bluetooth no Android

Arquivo: `android/app/src/main/kotlin/com/example/wearable_assistant/MainActivity.kt`

```

import android.content.Intent
import android.provider.Settings
import io.flutter.embedding.android.FlutterActivity
import io.flutter.plugin.common.MethodChannel

class MainActivity: FlutterActivity() {
    private val BLUETOOTH_CHANNEL = "bluetooth_helper"

    override fun configureFlutterEngine(flutterEngine: io.flutter.embedding.e
ngine.FlutterEngine) {
        super.configureFlutterEngine(flutterEngine)
        MethodChannel(flutterEngine.dartExecutor.binaryMessenger, BLUETO
OTH_CHANNEL).setMethodCallHandler { call, result →
            if (call.method == "openBluetoothSettings") {
                openBluetoothSettings()
                result.success(null)
            } else {
                result.notImplemented()
            }
        }
    }

    private fun openBluetoothSettings() {
        val intent = Intent(Settings.ACTION_BLUETOOTH_SETTINGS)
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)
        startActivity(intent)
    }
}

```

Subindo o Projeto no GitHub

```

git init
git add .
git commit -m "Primeira versão do Assistente Wear OS"
git branch -M main
git remote add origin <https://github.com/SEU_USUARIO/wearable_assistan

```

```
t.git>  
git push -u origin main
```

Conclusão

Este projeto demonstra como um aplicativo **Wear OS** pode ser utilizado para fornecer **assistência acessível e inclusiva**. Utilizando sensores de áudio e Bluetooth, o app permite **leitura de mensagens, comandos de voz e alertas de segurança**. A implementação foi feita com **Flutter e MethodChannel**, garantindo a comunicação entre **Dart e APIs nativas do Android**.

Agora o app está pronto para ser testado em um **simulador Wear OS** ou em um **dispositivo real!** 