



**Campus:** 202301037751 - POLO CENTRO - PORTO REAL - RJ

**Curso:** Desenvolvimento Full Stack

**Disciplina:** Nível 3: Lidando com sensores em dispositivos móveis

**Turma:** 9001

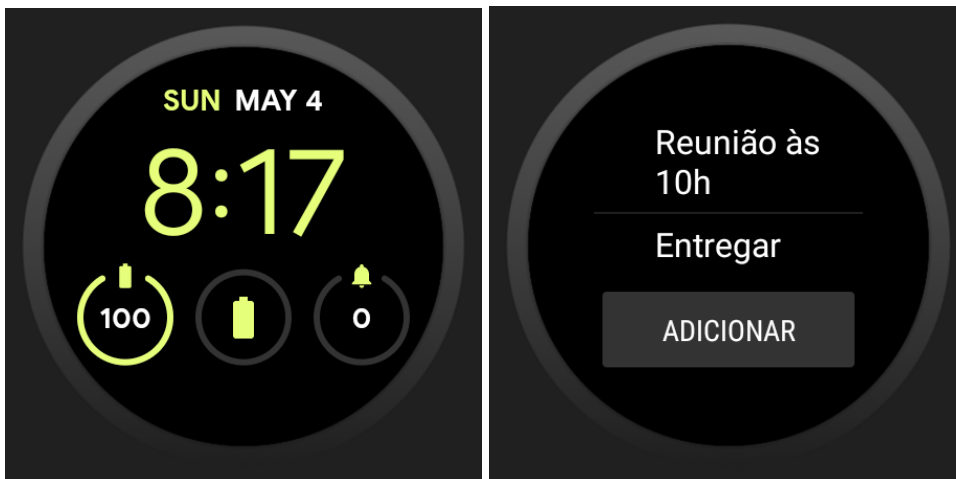
**Semestre letivo:** 5º Semestre

**Nome:** Leonardo Naves de Lima Araujo

Missão Prática | Nível 3 | Mundo 4

**Github do projeto:** <https://github.com/Navesz/Estacio-N3M4>

### Sobre o Projeto



O Lista de Tarefas para Wear OS é um aplicativo desenvolvido para dispositivos vestíveis que utiliza recursos de reconhecimento de voz e reprodução de áudio para criar uma experiência acessível de gerenciamento de tarefas. O aplicativo foi criado como parte da Missão Prática do Nível 3 do Mundo 4, focando na integração de sensores e recursos de áudio disponíveis em dispositivos Wear OS para melhorar a acessibilidade e experiência do usuário.

O projeto foi concebido como uma solução para a empresa fictícia "Doma", que desejava desenvolver um aplicativo Wear OS para assistência a funcionários com necessidades especiais, melhorando a comunicação e eficiência no ambiente de trabalho.

## **Funcionalidades Implementadas**

### **1. Sistema de Reconhecimento de Voz**

- Entrada por Comando de Voz: O usuário pode adicionar novas tarefas simplesmente falando, sem necessidade de digitação na pequena tela do relógio
- Feedback Sonoro: Um som de notificação é reproduzido para indicar o início e fim do reconhecimento
- Processamento Inteligente: Converte automaticamente o áudio em texto e adiciona à lista de tarefas

### **2. Sistema de Reprodução de Áudio**

- Leitura de Tarefas: Ao selecionar uma tarefa, o sistema reproduz seu conteúdo em áudio
- Compatibilidade Múltipla: Suporta reprodução via alto-falante integrado ou fones de ouvido Bluetooth
- Feedback Adaptativo: Oferece feedback visual e sonoro durante o processo de reprodução

### **3. Gerenciamento Inteligente de Dispositivos de Áudio**

- Detecção Automática: Identifica dispositivos de áudio disponíveis em tempo real
- Alternância Transparente: Alterna entre alto-falante e dispositivos Bluetooth sem interrupção da experiência
- Assistência Contextual: Guia o usuário para configuração de Bluetooth quando necessário

### **4. Interface Otimizada para Wearables**

- Design Adaptativo: Otimizado para dispositivos redondos e quadrados
- Elementos Visuais Claros: Botões grandes e texto legível para fácil interação em telas pequenas
- Navegação Simplificada: Interface minimalista que prioriza as principais funções

### **5. Integração com Sensores**

- Sensores Corporais: Utiliza sensores disponíveis no dispositivo para monitorar estado do usuário
- Wake Lock: Mantém o aplicativo ativo mesmo com a tela desligada para funcionalidades críticas
- Otimização de Energia: Gerencia recursos de hardware de forma eficiente

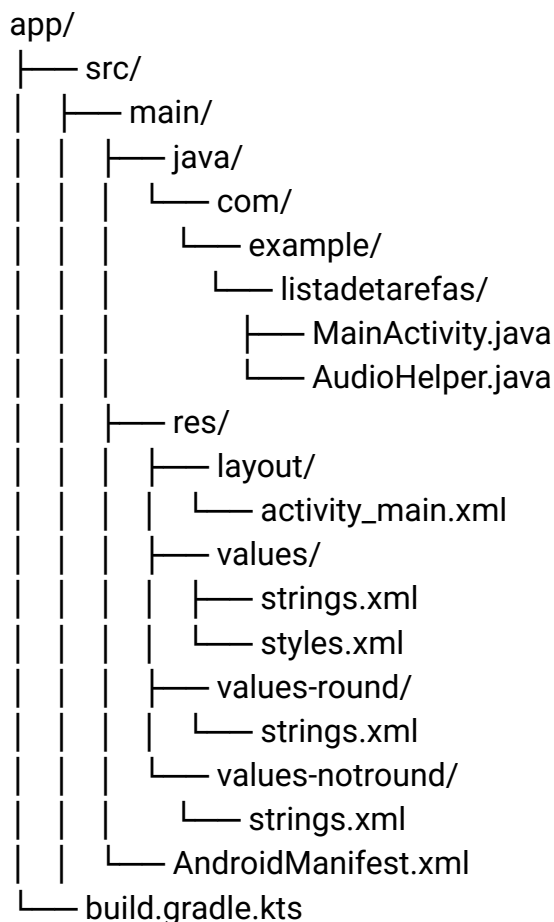
## **Tecnologias e APIs Utilizadas**

- Android SDK: Base de desenvolvimento com Java
- Wear OS SDK: APIs específicas para desenvolvimento de aplicativos vestíveis
- SpeechRecognizer: Sistema avançado de reconhecimento de voz do Android
- MediaPlayer: API para gerenciamento e reprodução de áudio
- AudioManager e AudioDeviceCallback: Monitoramento e controle de dispositivos de áudio

- BODY\_SENSORS Permission: Permissão para acesso a sensores corporais
- BoxInsetLayout: Layout especializado para interfaces adaptáveis em dispositivos Wear OS
- ConstraintLayout: Para designs responsivos e complexos

## Arquitetura do Projeto

### Estrutura de Arquivos



## Componentes Principais

1. MainActivity: Centro de controle do aplicativo, gerencia a interface do usuário e o fluxo de interação
2. AudioHelper: Classe utilitária para gerenciamento de dispositivos de áudio e verificação de compatibilidade
3. Layout XML: Definições de interface específicas para dispositivos Wear OS
4. Recursos Customizados: Strings e estilos adaptados para diferentes formatos de dispositivos

## Processo de Desenvolvimento

O desenvolvimento do aplicativo foi dividido em cinco microatividades principais:

### Microatividade 1: Configuração do Ambiente

- Instalação e configuração do Android Studio
- Configuração de dependências e SDKs
- Estudo das melhores práticas para acesso a sensores em Wear OS

### Microatividade 2: Criação do Projeto Base

- Criação de novo projeto com configurações específicas para Wear OS
- Configuração de build.gradle com dependências necessárias
- Definição da estrutura inicial de pacotes e classes

### Microatividade 3: Implementação da Lógica Básica

- Desenvolvimento de arquivos de configuração (AndroidManifest.xml)
- Implementação da classe MainActivity
- Criação da classe AudioHelper para gerenciamento de áudio

### Microatividade 4: Emulação e Testes

- Configuração de emulador Wear OS
- Testes de interface em diferentes tamanhos e formatos de tela
- Debugging inicial de funcionalidades

### Microatividade 5: Captura de Telas e Refinamentos

- Captura de telas para documentação
- Refinamentos de interface e experiência do usuário
- Otimizações de performance

## Desafios e Soluções

### Desafios Enfrentados:

1. Limitações de Tela: Desenvolver uma interface funcional para telas extremamente pequenas

Solução: Uso de layouts específicos para Wear OS (BoxInsetLayout) e simplificação da interface

2. Reconhecimento de Voz em Ambiente Ruidoso:

Solução: Implementação de feedback visual e sonoro para indicar o estado do reconhecimento

3. Gerenciamento de Energia: Balancear funcionalidades com consumo de bateria

Solução: Uso criterioso de Wake Lock e gerenciamento eficiente de listeners

4. Compatibilidade com Diferentes Dispositivos de Áudio:

Solução: Implementação de AudioDeviceCallback para detecção dinâmica de dispositivos

## 5. Interface Adaptativa:

Solução: Criação de recursos específicos para dispositivos redondos e quadrados

## Resultados e Aprendizados

### Resultados Alcançados:

- Aplicativo funcional que atende às necessidades especificadas
- Interface adaptativa que funciona bem em diferentes dispositivos Wear OS
- Implementação bem-sucedida de reconhecimento de voz e reprodução de áudio
- Gerenciamento eficiente de recursos de hardware

### Principais Aprendizados:

- Desenvolvimento específico para dispositivos vestíveis
- Integração com APIs de reconhecimento de voz e reprodução de áudio
- Criação de interfaces adaptativas para telas pequenas
- Gerenciamento de permissões e sensores em Wear OS
- Otimização de aplicativos para dispositivos com recursos limitados

## Como Executar o Projeto

### 1. Clone o repositório:

```
git clone https://github.com/Navesz/Estacio-N3M4.git  
cd Estacio-N3M4
```

### 2. Abra o projeto no Android Studio

### 3. Configure um emulador Wear OS ou conecte um dispositivo físico:

- No Android Studio, acesse "Tools > Device Manager"
- Selecione "Create Device" e escolha um dispositivo Wear OS (recomendado: "Wear OS Small Round")
- Selecione a imagem do sistema com API 30 ou superior

### 4. Execute o aplicativo:

- Selecione a configuração "Android App"
- Clique em "Run" (▶)

## Considerações Finais

O desenvolvimento deste aplicativo para Wear OS proporcionou uma excelente oportunidade para explorar o desenvolvimento para dispositivos vestíveis, com foco especial em acessibilidade e uso de sensores. A experiência adquirida com este projeto será valiosa para

futuros desenvolvimentos mobile, especialmente aqueles voltados para dispositivos com interfaces não convencionais e foco em acessibilidade.

O aplicativo Lista de Tarefas para Wear OS demonstra como a tecnologia vestível pode ser utilizada para melhorar a produtividade e assistência a pessoas com necessidades especiais, alinhando-se com as tendências atuais de desenvolvimento de aplicativos inclusivos e acessíveis.