

# Missão Prática | Nível 3 | Mundo 4

Desenvolvimento de Aplicativo Wear OS para Sensores e Áudio

## Sumário

- 1 1. Introdução
- 2 2. Configuração do Ambiente
- 3 3. Criação do Projeto Wear OS
- 4 4. Configuração do Emulador
- 5 5. Implementação de Funcionalidades de Sensores e Áudio
- 6 6. Captura de Telas
- 7 7. Tecnologias Utilizadas
- 8 8. Conclusão

## 1. Introdução

Este documento apresenta, de forma clara e objetiva, o passo a passo do desenvolvimento da Missão Prática Nível 3 - Mundo 4. O foco principal foi a criação de um aplicativo Wear OS capaz de interagir com sensores do dispositivo e gerenciar saídas de áudio.

## 2. Configuração do Ambiente

Para iniciar, instalou-se o Android Studio e configurou-se o emulador para Wear OS. Foram seguidas as instruções do site oficial do Android Developers para download e instalação. Após a instalação, o AVD Manager foi utilizado para criar um dispositivo virtual com API 30.

## 3. Criação do Projeto Wear OS

Utilizou-se o template 'No Activity' no Android Studio, definindo o nome do projeto 'ListaDeTarefas' e o mínimo de SDK como Android 11.0 (API 30). O Android Studio gerou então a estrutura básica de pastas e arquivos do aplicativo.

## 4. Configuração do Emulador

No Device Manager, criou-se um novo dispositivo virtual selecionando o tipo 'Wear OS Small Round' e API 30. As configurações padrão foram mantidas para memória e orientação.

## 5. Implementação de Funcionalidades de Sensores e Áudio

Foi desenvolvida uma classe auxiliar para verificar saídas de áudio disponíveis utilizando o AudioManager. Em seguida, registrou-se callbacks para detectar a conexão e desconexão de fones Bluetooth, melhorando a experiência do usuário ao redirecionar para as configurações do sistema quando necessário.

## 6. Captura de Telas

Realizaram-se capturas de tela diretamente pelo Android Studio e também via app complementar para dispositivos Wear OS, permitindo documentação e validação visual dos testes.

## 7. Tecnologias Utilizadas

- Android Studio IDE
- Android SDK e Wear OS APIs
- AVD Manager (emulador)
- Java/Kotlin e XML
- AudioManager e AudioDeviceInfo
- Git e GitHub para controle de versão
- Sistemas operacionais Windows, Linux e macOS

## **8. Conclusão**

O projeto demonstrou todo o ciclo de desenvolvimento de um aplicativo Wear OS, desde a configuração do ambiente até a implementação de funcionalidades avançadas. Permitiu aplicar conhecimentos de sensores e gerenciamento de áudio, resultando em uma solução inclusiva e acessível.