



Estácio

Missão Prática | Nível 3 | Mundo 4

RPG0025 - Lidando com sensores em dispositivos móveis!

Rian Joseph Ramos Felizardo - 202202923931

POLO BARREIRO - Belo Horizonte, MG

Nível 3 – Lidando com sensores em dispositivos móveis – 2025.1 – 5º Semestre Letivo

Objetivo da Prática

- Instalação do Android Studio e do emulador;
- Criar um app para Wear OS;
- Executar um app no emulador;
- Fazer capturas de telas no Android Studio;
- Fazer capturas de telas com app complementar.

Contextualização

Para uma melhoria na eficiência e na comunicação interna, a empresa “Doma” quer desenvolver um aplicativo Wear OS para assistência aos funcionários que têm necessidades especiais, uma forma de solidificar a interação entre os mesmos. Assim, com os aplicativos wearables podem usar áudio para fornecer informações em tempo real, como leitura de mensagens de texto, notificações, lembretes e respostas a comandos de voz. Isso pode ser especialmente útil para pessoas com deficiência visual. Além de serem úteis para treinamento e educação. Aplicativos podem usar áudio para fornecer instruções, dicas e feedbacks durante o aprendizado ou a prática de novas habilidades. Outra funcionalidade que a empresa quer adotar, é um aplicativo wearable que pode usar o áudio para fornecer alertas de segurança, como notificações de emergência, alertas de tempestades, notícias importantes ou informações críticas.

- Nesta atividade a seguir compreenderemos que os apps do Wear OS podem funcionar como um dos principais frameworks para o desenvolvimento de aplicações mobile. Um aplicativo Wearable pode ter várias especialidades, desde entretenimento e comunicação.

Componente Principal Kotlin MainActivity.dart

Classe principal do app. Responsável por inicializar os botões, lidar com eventos de clique, sintetizar fala com TTS, receber comandos de voz e enviar notificações ao usuário.

Funções principais

- **speak(text: String)**
Utiliza o mecanismo de Text-to-Speech (TTS) para falar mensagens em voz alta.
- **startVoiceInput()**
Inicia o reconhecimento de voz do Android para capturar comandos falados pelo usuário.
- **handleVoiceCommand(command: String?)**
Interpreta o comando de voz recebido. Executa ações como:
Falar um alerta se o comando contiver “alerta”.
Informar uma tarefa se o comando contiver “tarefa”.
Informar que o comando não foi reconhecido, caso contrário.
- **sendAlert(message: String)**
Fala o alerta em voz alta e envia uma notificação com o conteúdo.
- **sendNotification(title: String, content: String)**
Cria e exibe uma notificação com título e conteúdo personalizados.
- **onInit(status: Int)**
Inicializa o TTS com o idioma padrão do sistema, caso não haja erro.
- **onDestroy()**
Libera os recursos do TTS ao encerrar a atividade.

Layout/Seções de estilização

- **BoxInsetLayout**
Container de layout que garante o ajuste automático para dispositivos Wear OS com diferentes formatos de tela.
- **LinearLayout (vertical)**
Organiza os botões centralizados verticalmente, com espaçamento entre eles.

Componentes da interface

- **Botão "Ler Mensagem"**
Aciona a leitura de uma mensagem via áudio (Text-to-Speech).
- **Botão "Comando de Voz"**
Inicia o reconhecimento de comandos de voz do usuário.
- **Botão "Alerta de Segurança"**
Dispara um alerta de voz e envia uma notificação.
- **Botão "Alerta Personalizado"**
Reservado para alertas personalizados (comportamento definido no código Kotlin).

Estilo visual

- Todos os botões usam o fundo personalizado `@drawable/rounded_button` e texto branco (`#FFFFFF`), com margens verticais para espaçamento.

Dependências do projeto

Parte das principais dependências do projeto.

Principais

- **libs.play.services.wearable**
Suporte às funcionalidades do Google Play Services para dispositivos Wear OS.
- **libs.appcompat**
Suporte à compatibilidade de componentes visuais com versões antigas do Android.
- **libs.material**
Componentes visuais baseados no Material Design.
- **libs.activity**
Gerenciamento do ciclo de vida e interações com atividades.
- **libs.constraintlayout**
Gerenciamento avançado de layout para interface de usuário responsiva.

Dependências diretas adicionais

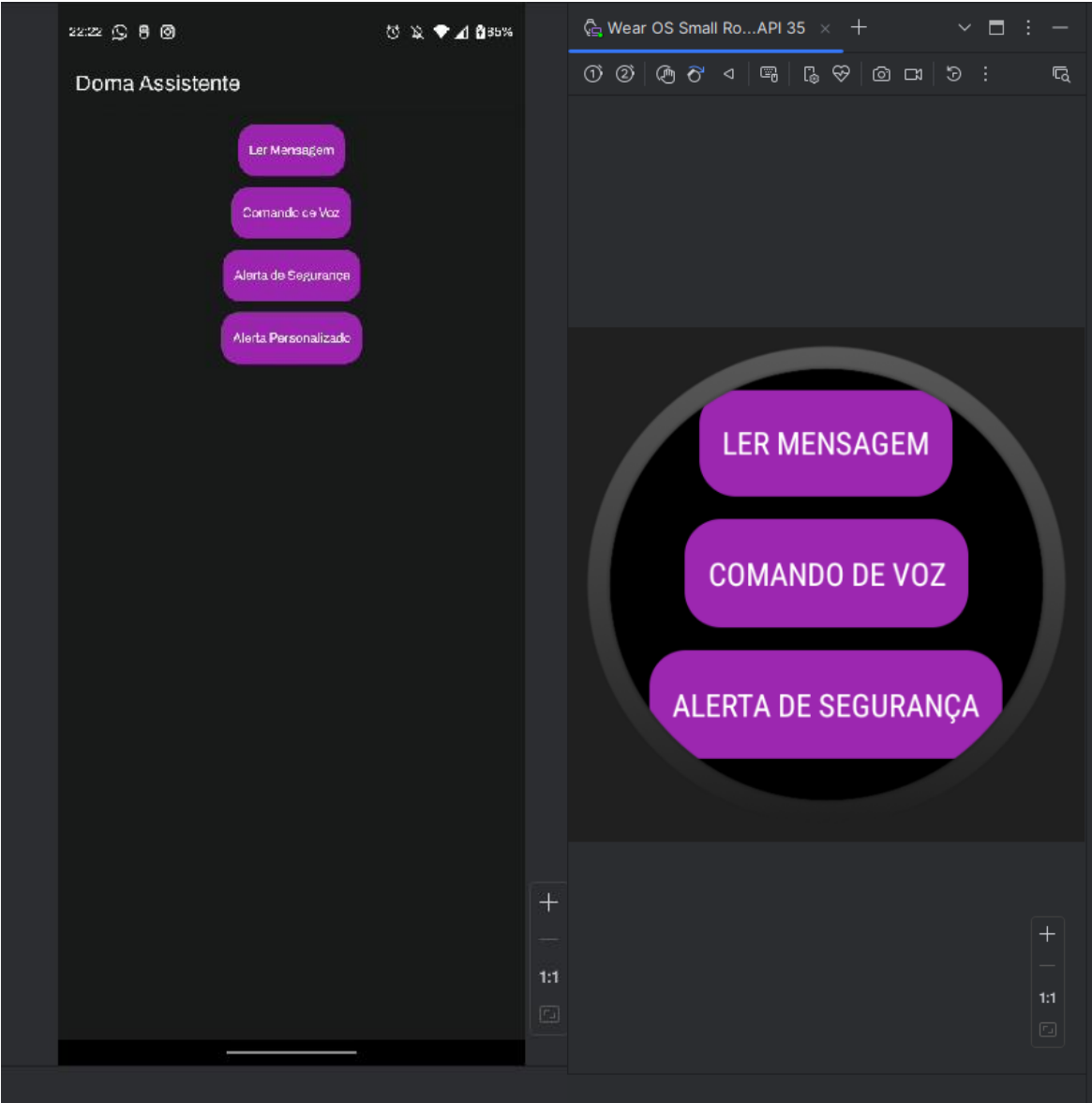
- **androidx.wear:wear:1.3.0**
Biblioteca específica para o desenvolvimento de interfaces e interações em dispositivos Wear OS.
- **androidx.constraintlayout:constraintlayout:2.1.0**
Versão específica da ConstraintLayout usada para definir layouts complexos.
- **com.google.android.material:material:1.6.0**
Implementação direta da biblioteca Material Design para interfaces modernas e responsivas.

Resultados esperados ✨

Ao concluir esta missão, os alunos terão desenvolvido um aplicativo Wear OS que proporciona uma comunicação eficaz e assistência para funcionários com necessidades especiais. O aplicativo deverá ser capaz de ler mensagens e notificações em voz alta, responder a comandos de voz e fornecer alertas de segurança e instruções através de áudio. Este aplicativo não apenas melhora a eficiência e a comunicação interna na empresa "Doma", mas também demonstra a aplicação prática de tecnologias wearables para criar soluções acessíveis e inclusivas no local de trabalho

Tela do App

Tela de Exibição conforme enunciado do trabalho.



Repositório

[rianjsp/missao-pratica-3-mundo-4: Missão pratica nivel 3 mundo 4](#)