



Trabalho Prático | Interação com sensores de smartphones e wearables

Aluno: Luiz Carlos Marinho Junior

Matrícula: 202311175571

Polo Centro, Maricá - RJ

Interação com sensores de smartphones e wearables – 9001 – 2025.4

1. Objetivo da Prática

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.

Este relatório descreve a análise do projeto Android chamado "doma-business", que está sendo desenvolvido especificamente para dispositivos Wear OS (relógios inteligentes).

O objetivo principal do aplicativo é atuar como um assistente pessoal no pulso do usuário, oferecendo funcionalidades de comando de voz, leitura de notificações e alertas de emergência.

2. Repositório do Projeto

Link do projeto no Github: <https://github.com/luizmarinhojr/interacao-com-sensores-de-smartphones-e-wearebles>

3. Visão Geral do Projeto

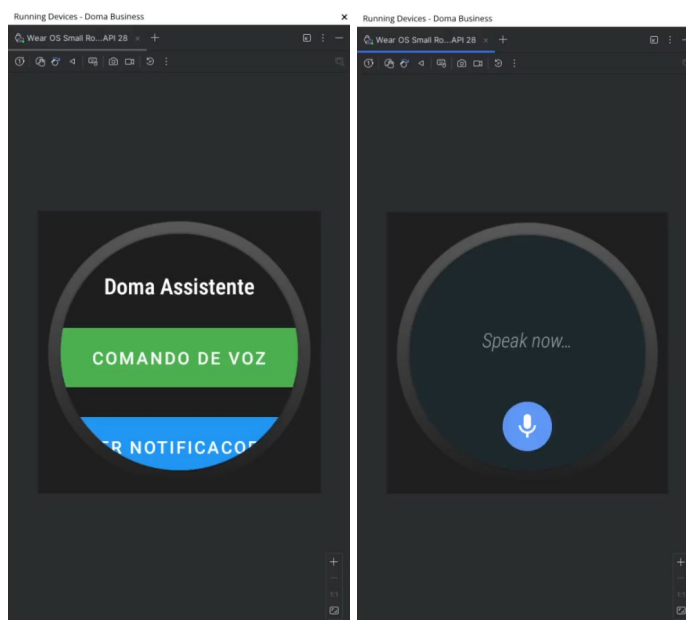
O projeto "doma-business" é um aplicativo Android feito para Wear OS que busca fornecer uma experiência de assistência inteligente e interativa. Ele visa aprimorar a interação do usuário com seu relógio inteligente, permitindo que ele execute ações e receba informações importantes através de comandos de voz e feedback de áudio. A inclusão de um serviço de leitura de notificações e um recurso de alerta de emergência sugere um foco em produtividade, acessibilidade e segurança para o usuário em movimento.

4. Tecnologias Utilizadas

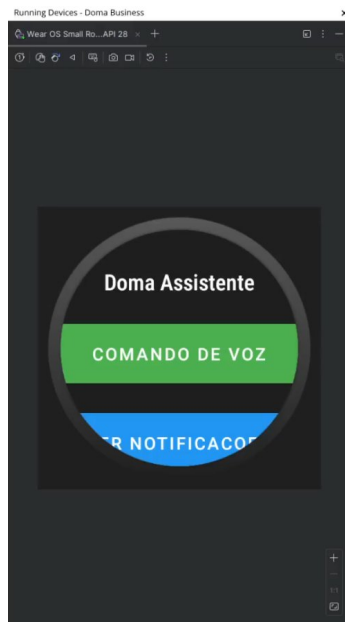
O aplicativo é desenvolvido em Kotlin, a linguagem preferida para o desenvolvimento Android moderno, e utiliza o Gradle com Kotlin DSL (`build.gradle.kts`) para gerenciamento de dependências e configuração do projeto. Ele se baseia fortemente no ecossistema AndroidX, incluindo bibliotecas como `androidx.core.ktx`, `androidx.appcompat`, e `material`, indicando uma adesão às práticas de desenvolvimento atuais e um design de interface do usuário moderno. A versão mínima do SDK é 28, e a compilação é direcionada para o SDK 36, garantindo compatibilidade com uma ampla gama de dispositivos Wear OS, mantendo acesso às funcionalidades mais recentes da plataforma.

5. Funcionalidades Principais

1. Comandos de Voz Inteligentes: O coração do aplicativo reside em sua capacidade de receber e processar comandos de voz. Através do reconhecimento de fala, o usuário pode interagir com o assistente para, por exemplo, ler notificações, acionar um alerta de emergência, obter instruções de uso ou saber a hora atual. A `MainActivity` é responsável por iniciar o reconhecimento de voz e interpretar os comandos.



2. Conversão de Texto em Fala (TTS - Text-to-Speech): O aplicativo utiliza o serviço TTS para fornecer feedback de áudio ao usuário. Isso inclui saudações, confirmações de comandos, leitura das notificações e a transmissão de mensagens de alerta de emergência e instruções. O suporte ao português do Brasil (pt-BR) é uma prioridade, com fallback para o inglês dos EUA, garantindo que a comunicação seja clara e natural para o público-alvo.



3. Leitura de Notificações: Uma funcionalidade crucial para um assistente de Wear OS é a capacidade de ler notificações. O NotificationReaderService implementa o NotificationListenerService do Android para interceptar notificações de outros aplicativos. Ele extrai informações como título, texto e nome do aplicativo remetente, e as pronuncia em voz alta. O serviço também pode ler uma lista de notificações pendentes a pedido do usuário. Há um mecanismo para orientar o usuário a conceder a permissão necessária para esta funcionalidade.



- Alerta de Emergência: O aplicativo inclui um botão e um comando de voz dedicados para acionar um alerta de emergência. Quando ativado, o dispositivo emite uma vibração prolongada e o assistente fala repetidamente uma mensagem de alerta predefinida, fornecendo informações de segurança e instruções.



- Feedback Vibratório: Para melhorar a experiência do usuário, o aplicativo incorpora feedback vibratório em resposta a interações, como cliques em botões e o acionamento de alertas de emergência.



- Configurações ('SettingsActivity'): Existe uma SettingsActivity dedicada, que atualmente serve como um placeholder. Embora a funcionalidade de configurações ainda não esteja implementada, sua presença sugere planos futuros para permitir que os usuários personalizem o comportamento do assistente.

6. Permissões Requeridas:

O aplicativo solicita várias permissões essenciais para suas funcionalidades:

- `android.permission.RECORD_AUDIO`: Necessária para o reconhecimento de voz.
- `android.permission.INTERNET`: Provavelmente para comunicação com serviços externos (se houver) ou para o funcionamento do TTS/reconhecimento de voz.
- `android.permission.VIBRATE`: Para fornecer feedback tátil ao usuário.
- `android.permission.POST_NOTIFICATIONS`: Para o aplicativo postar suas próprias notificações (por exemplo, confirmações ou informações).
- `android.permission.READ_NOTIFICATION_POLICY`: Relacionado à gestão de políticas de notificação.
- `android.permission.BIND_NOTIFICATION_LISTENER_SERVICE`: A permissão mais crítica, que permite ao `NotificationReaderService` ler as notificações de outros aplicativos.

7. Observações e Próximos Passos (Implícito):

O projeto "doma-business" está bem encaminhado para ser um assistente de Wear OS funcional. A lógica principal para reconhecimento de voz, TTS e leitura de notificações está presente. No entanto, a `SettingsActivity` está atualmente vazia, indicando que a implementação de opções configuráveis pelo usuário ainda não foi iniciada ou concluída. Além disso, a função `getNotificationService()` na `MainActivity` atualmente retorna `null`, o que significa que a comunicação direta entre a `MainActivity` e o `NotificationReaderService` para solicitar notificações pendentes ainda precisa ser devidamente estabelecida ou refatorada para usar um mecanismo de comunicação entre serviços mais robusto.

No geral, o projeto demonstra um design claro e um esforço para criar um aplicativo Wear OS útil e interativo, focado na assistência por voz e na gestão de notificações.