



Lidando com sensores em dispositivos móveis

Nome: Thomaz Alves Da Costa **Matrícula:** 202303196751

Campus: Campus Virtual (SIA)

Lidando com Sensores em dispositivos móveis – 2023.1 – 4º Semestre

Objetivo da Prática:

- Criar uma aplicação wearable.
- Conhecer e saber alguns princípios básicos.
- Instalação do Android Studio e do emulador.
- Executar um app no emulador.
- Fazer capturas de telas no Android Studio.
- Fazer capturas de telas com app complementar.

Códigos da missão prática:

OnCreate e OnInit:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    textToSpeech = new TextToSpeech(this, this);

    speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
    startListening();
}

@Override
public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {
        int result = textToSpeech.setLanguage(Locale.getDefault());
        if (result == TextToSpeech.LANG_MISSING_DATA || result == TextToSpeech.LANG_NOT_SUPPORTED) {
            speak("Error nao conseguimos processar a mensagem ou alerta");
        }
    }
}

public void speak(String message) {
    textToSpeech.speak(message, TextToSpeech.QUEUE_FLUSH, null, null);
}

@Override
protected void onDestroy() {
    if (textToSpeech != null) {
        textToSpeech.stop();
        textToSpeech.shutdown();
    }
    super.onDestroy();
}
```

Lista de perguntas que podem ser feitas pelo usuário:

```
// Comandos que o usuário pode falar que serão reconhecidos
private void handleCommand(String command) {
    if (command.toLowerCase().contains("ler mensagem")) {
        speak("Você tem uma nova mensagem: Olá, bem-vindo ao DomaAssist!");
    } else if (command.toLowerCase().contains("notificação")) {
        sendNotification("Alerta de Segurança", "Esta é uma notificação de exemplo.");
    } else if (command.toLowerCase().contains("ajuda")) {
        speak("Como posso ajudar você?");
    } else {
        speak("Comando não reconhecido: " + command);
    }
}
}
```

Função que inicializa o speech:

```
// Iniciar o reconhecimento de voz
public void startListening() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    speechRecognizer.setRecognitionListener(new RecognitionListener() {
        @Override
        public void onReadyForSpeech(Bundle params) {}

        @Override
        public void onBeginningOfSpeech() {}

        @Override
        public void onRmsChanged(float rmsdB) {}

        @Override
        public void onBufferReceived(byte[] buffer) {}

        @Override
        public void onEndOfSpeech() {
            // Reiniciar o reconhecimento
            startListening();
        }

        @Override
        public void onError(int error) {
            Log.e("SpeechRecognizer", "Error: " + error);
            startListening(); // Reiniciar após erro
        }

        @Override
        public void onResults(Bundle results) {
            List<String> matches = results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
            if (matches != null && !matches.isEmpty()) {
                String spokenText = matches.get(0);
                handleCommand(spokenText);
            }
            startListening(); // Reiniciar após obter resultados
        }

        @Override
        public void onPartialResults(Bundle partialResults) {}

        @Override
        public void onEvent(int eventType, Bundle params) {}
    });

    speechRecognizer.startListening(intent);
}
```

Função que é responsável por enviar as notificações ao usuário:

```
// Enviar notificações
public void sendNotification(String title, String message) {
    NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    String channelId = "security_alerts";

    NotificationChannel channel = new NotificationChannel(channelId, "Security Alerts", NotificationManager.IMPORTANCE_HIGH);
    notificationManager.createNotificationChannel(channel);

    Intent intent = new Intent(this, MainActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this, channelId)
        .setContentTitle(title)
        .setContentText(message)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true);

    notificationManager.notify(1, builder.build());
}
```

Conclusão:

Criar uma aplicação wearable para mim foi um desafio muito grande pois eu não tinha muito conhecimento, porém eu já sabia da linguagem Java, que me ajudou muito a realizar este desafio. Mais um detalhe q eu gostei e muito, foi a construção da interface, que pode ser via código ou via ferramenta de construção, acho que esta foi uma das coisas que mais me agradou.