



Missão Prática | Nível 3 | Mundo 4

Filipe Alvim Santos - 202208291325

Campus Polo Sulacap – RJ / Desenvolvimento Full Stack

**Nível 3: Lidando com sensores em dispositivos móveis – Turma: 22.3
– 4º Semestre**

Objetivo da Prática

Os objetivos da prática são: Instalação do Android Studio e do emulador. Criar um app para Wear OS. Executar um app no emulador e fazer capturas de telas no Android Studio;

Missão Prática – Lidando com sensores em dispositivos móveis

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.

MainActivity e seu layout, no qual tem dois buttons que direcionam para a MessagesActivity ou NotificationsActivity:

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        val messagesButton = findViewById<ImageButton>(R.id.messages_button)  
        val notificationsButton = findViewById<ImageButton>(R.id.notifications_button)  
  
        messagesButton.setOnClickListener {  
            val intent = Intent(this, MessagesActivity::class.java)  
            startActivity(intent)  
        }  
  
        notificationsButton.setOnClickListener {  
            val intent = Intent(this, NotificationsActivity::class.java)  
            startActivity(intent)  
        }  
    }  
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/header_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/app_name"
        android:textColor="@color/textColor"
        android:textSize="34sp"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:orientation="horizontal"
        >

        <ImageButton
            android:id="@+id/messages_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:contentDescription="@string/message_btn"
            android:src="@drawable/messages_icon"
            android:layout_marginEnd="30dp"
            android:background="?android:attr/selectableItemBackground" />

        <ImageButton
            android:id="@+id/notifications_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:contentDescription="@string/notifications_btn"
            android:src="@drawable/notifications_icon"
            android:background="?android:attr/selectableItemBackground" />
    </LinearLayout>
</RelativeLayout>
```

MessagesActivity, o layout do caixa de diálogo (modal), layout de mensagens(listView) e o layout de textView de mensagens :

```
class MessagesActivity : AppCompatActivity() {

    private lateinit var textToSpeech: TextToSpeech

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_messages)

        textToSpeech = TextToSpeech(this) { status →
            if (status ≠ TextToSpeech.ERROR) {
                textToSpeech.setLanguage(Locale("pt", "BR"))
            }
        }

        val messages = mapOf(
            "Filipe" to "Reunião às 10 horas amanhã",
            "João" to "Entrega do projeto na sexta-feira",
            "José" to "Me liga depois",
            "Gabriel" to "PR para revisar",
            "Pedro" to "Fechar contrato com a empresa X"
        )
        val listView = findViewById<ListView>(R.id.listView)
        val adapter = ArrayAdapter<this, R.layout.list_item, messages.keys.toList>()
        listView.adapter = adapter
        listView.setOnItemClickListener { _, _, position, _ →
            val name = messages.keys.elementAt(position)
            val message = messages[name]

            val dialogView = LayoutInflater.from(this).inflate(R.layout.dialog_layout, null)
            textToSpeech.speak("Gostaria de ouvir a mensagem de $name?", TextToSpeech.QUEUE_FLUSH, null, "")
            val dialogMessage = dialogView.findViewById<TextView>(R.id.dialog_message)
            val positiveButton = dialogView.findViewById<Button>(R.id.dialog_positive_button)
            val negativeButton = dialogView.findViewById<Button>(R.id.dialog_negative_button)

            dialogMessage.text = "Gostaria de ouvir a mensagem de $name?"

            val alertDialog = AlertDialog.Builder(this).apply {
                setView(dialogView)
                create()
            }.show()

            positiveButton.setOnClickListener {
                textToSpeech.speak("Ouvindo a mensagem de $name: $message", TextToSpeech.QUEUE_FLUSH, null, "")
                alertDialog.dismiss()
            }
            negativeButton.setOnClickListener {
                alertDialog.dismiss()
            }
        }
    }

    override fun onDestroy() {
        if (textToSpeech.isSpeaking) {
            textToSpeech.stop()
        }
        textToSpeech.shutdown()
        super.onDestroy()
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:background="@color/background_color"
>

    <TextView
        android:paddingTop="30dp"
        android:id="@+id/dialog_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:gravity="center"
        android:textColor="@color/text_color"
        android:fontFamily="serif-monospace"
        android:textFontWeight="700"
    />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="55dp">

        <Button
            android:id="@+id/dialog_positive_button"
            android:layout_width="60dp"
            android:layout_height="50dp"
            android:text="@string/yes_btn"
            android:fontFamily="serif-monospace"
            android:textFontWeight="700"
            android:textColor="@color/text_color"
            android:background="?android:attr/selectableItemBackground"
        />

        <Button
            android:id="@+id/dialog_negative_button"
            android:layout_width="60dp"
            android:layout_height="50dp"
            android:background="?android:attr/selectableItemBackground"
            android:text="@string/no_btn"
            android:textColor="@color/text_color"
            android:fontFamily="serif-monospace"
            android:textFontWeight="700" />

    </LinearLayout>

</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MessagesActivity">

    <ListView
      android:id="@+id/listView"
      android:background="@color/background_color"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" />

  </RelativeLayout>
```

```
/
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/textItemView"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:padding="16dp"
  android:textSize="20sp"
  android:textColor="@color/text_color"
  android:fontFamily="serif-monospace"
  android:textFontWeight="500"
  android:gravity="center" />
```

NotificationsActivity, seu layout e o layout do textView de notificações:

```
class NotificationsActivity : AppCompatActivity() {
    private lateinit var textToSpeech: TextToSpeech
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_notifications)

        textToSpeech = TextToSpeech(this) { status →
            if (status ≠ TextToSpeech.ERROR) {
                textToSpeech.setLanguage(Locale("pt", "BR"))
            }
        }

        val notifications = mapOf(
            "Alerta - Clima" to "Previsão de temporal hoje",
            "Alerta - Casa" to "Problema com a fiação elétrica",
            "Alerta - Segurança" to "Roubo na vizinhança",
            "Alerta - Trânsito" to "Acidente na avenida principal",
            "Alerta - Educação" to "Suspensão das aulas presenciais",
            "Alerta - Saúde" to "Vacinação contra a COVID-19 disponível"
        )

        val listView = findViewById<ListView>(R.id.listNotificationView)
        val adapter = ArrayAdapter(this, R.layout.list_item_notification, notifications.keys.toList())
        listView.adapter = adapter

        listView.setOnItemClickListener { _, _, position, _ →
            val alerta = notifications.keys.elementAt(position)
            val message = notifications[alerta]
            textToSpeech.speak(
                "$alerta: $message",
                TextToSpeech.QUEUE_FLUSH,
                null,
                ""
            )
        }
    }

    override fun onDestroy() {
        if (textToSpeech.isSpeaking) {
            textToSpeech.stop()
        }
        textToSpeech.shutdown()
        super.onDestroy()
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NotificationsActivity">

    <ListView
      android:id="@+id/listNotificationView"
      android:background="@color/background_color"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      />

  </RelativeLayout>
```

```
/
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/textItemNotificationView"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:padding="16dp"
  android:textSize="15sp"
  android:textColor="@color/alert"
  android:fontFamily="serif-monospace"
  android:textFontWeight="700"
  android:gravity="center" />
```