



Centro Universitário Estácio
Polo São Lourenço da Mata – PE

Curso: Desenvolvimento Full Stack

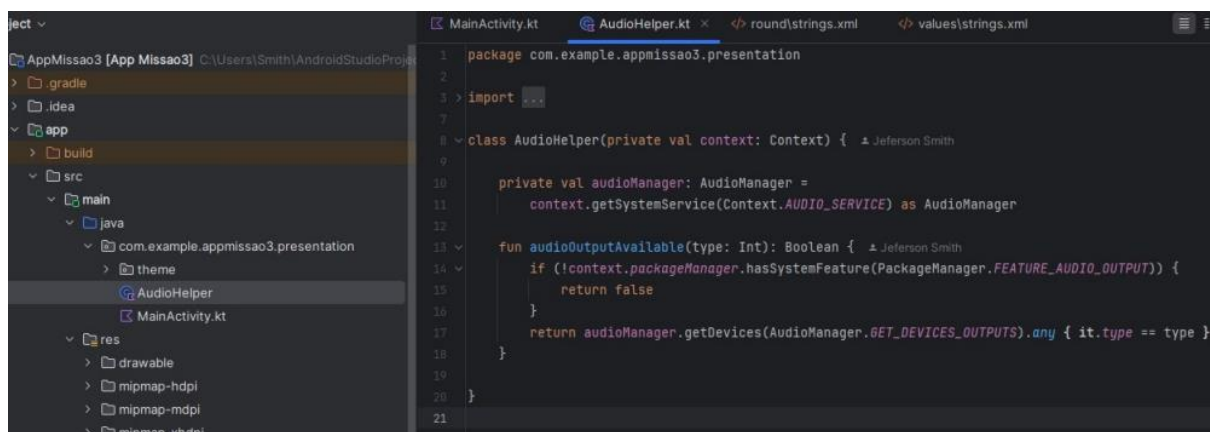
Disciplina: Lidando com sensores em dispositivos móveis - Nível 3

Semestre: 4º

Aluno: Jonnatha Walben Saldanha da Silva

Audio Helper

A classe `AudioHelper` é responsável por verificar se o dispositivo possui uma saída de áudio disponível. Ela utiliza o `AudioManager` para acessar informações sobre dispositivos de áudio conectados e valida se um tipo específico de saída está presente. Adicionalmente, realiza uma verificação preliminar para garantir que o dispositivo oferece suporte à funcionalidade de saída de áudio.



MainActivity

A classe `MainActivity` em um aplicativo para Wear OS lida com a gestão de conexões de áudio, especialmente dispositivos Bluetooth A2DP. Sempre que um fone Bluetooth é conectado ou desconectado, o aplicativo exibe uma mensagem ao usuário (Toast).

Em caso de desconexão, ele também direciona automaticamente para as configurações de Bluetooth, facilitando a reconexão.

A interface do aplicativo, desenvolvida com Jetpack Compose, apresenta um relógio posicionado no topo da tela e uma mensagem de saudação centralizada ("Hello, World"), estilizada com um tema personalizado. A lógica de monitoramento dos dispositivos Bluetooth é implementada por meio de um `AudioDeviceCallback`, que detecta a adição e remoção de dispositivos. Além disso, o aplicativo exibe uma tela de splash durante o carregamento inicial e adota o tema visual padrão do Wear OS.

```
1 MainActivity.kt 2 AudioHelper.kt 3 roundStrings.xml 4 valuesStrings.xml
package com.example.appmissao3.presentation

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.media.AudioManager
import android.media.AudioDeviceInfo
import android.media.AudioDeviceCallback
import android.media.AudioDeviceChangeListener
import android.media.AudioDeviceCallback
import android.media.AudioDeviceChangeListener
import android.media.AudioDeviceCallback
import android.media.AudioDeviceChangeListener

class MainActivity : AppCompatActivity() {
    private lateinit var audioManager: AudioManager
    private lateinit var audioHelper: AudioHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        installSplashScreen()
        super.onCreate(savedInstanceState)

        setTheme(android.R.style.Theme_DeviceDefault)

        audioManager = getSystemService(Context.AUDIO_SERVICE) as AudioManager
        audioHelper = AudioHelper(this)

        val audioDeviceCallback = object : AudioDeviceCallback() {
            override fun onAudioDevicesAdded(addedDevices: Array<out AudioDeviceInfo>?) {
                super.onAudioDevicesAdded(addedDevices)
                if (audioHelper.audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2DP)) {
                    showToast("Fone de ouvido Bluetooth conectado")
                }
            }

            override fun onAudioDevicesRemoved(removedDevices: Array<out AudioDeviceInfo>?) {
                super.onAudioDevicesRemoved(removedDevices)
                if (removedDevices != null) {
                    for (device in removedDevices) {
                        if (device.type == AudioDeviceInfo.TYPE_BLUETOOTH_A2DP) {
                            promptBluetoothConnection()
                            showToast("Fone de ouvido Bluetooth desconectado")
                        }
                    }
                }
            }
        }

        audioManager.registerAudioDeviceCallback(audioDeviceCallback, handler = null)
    }
}
```

```

setContent {
    WearApp(greetingName: "Android")
}

private fun showToast(message: String) {
    Toast.makeText(context, message, Toast.LENGTH_SHORT).show()
}

private fun promptBluetoothConnection() {
    val intent = Intent(Settings.ACTION_BLUETOOTH_SETTINGS)
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK)
    startActivity(intent)
}

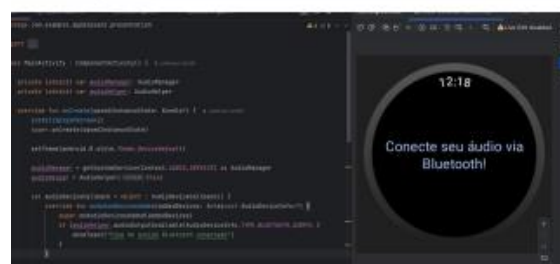
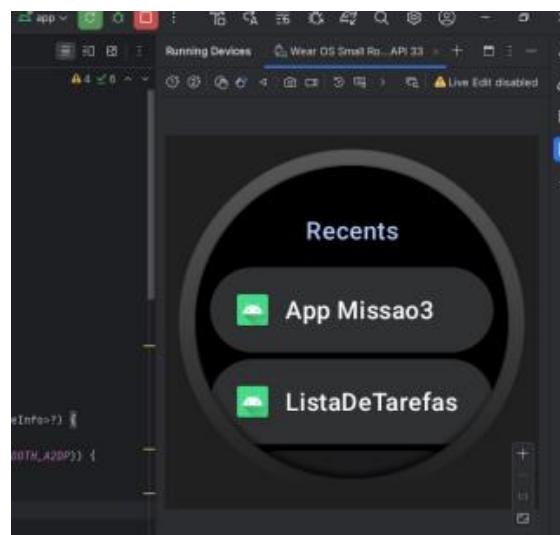
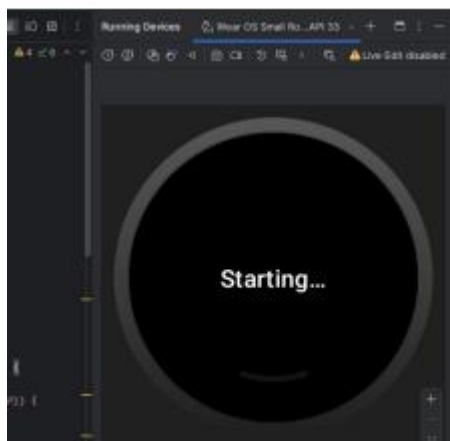
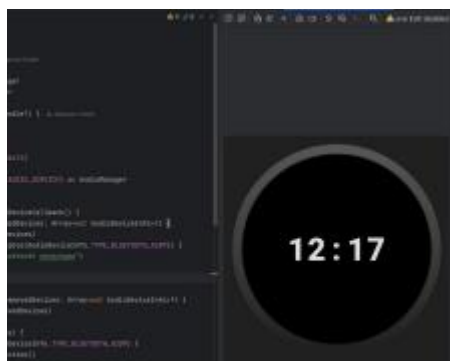
@Composable
fun WearApp(greetingName: String) {
    AppMissaoTheme {
        Box(
            modifier = Modifier
                .fillMaxSize()
                .background(MaterialTheme.colors.background),
            contentAlignment = Alignment.Center
        ) {
            TimeText()
            Greeting(greetingName = greetingName)
        }
    }
}

@Composable
fun Greeting(greetingName: String) {
    Text(
        modifier = Modifier.fillMaxWidth(),
        textAlign = TextAlign.Center,
        color = MaterialTheme.colors.primary,
        text = stringResource(id = R.string.conecte_seu_audio_via_bluetooth)
    )
}

@Preview(device = Devices.WEAR_OS_SMALL_ROUND, showSystemUI = true)
@Composable
fun DefaultPreview() {
    WearApp(greetingName: "Preview Android")
}

```

Executando o aplicativo



A empresa *Doma* pretende desenvolver um aplicativo Wear OS com o objetivo de melhorar a eficiência e a comunicação interna, especialmente voltada ao apoio de colaboradores com necessidades especiais, promovendo uma interação mais inclusiva e integrada.

Com a utilização de tecnologias vestíveis, o áudio desempenha um papel essencial ao fornecer informações em tempo real, como leitura de mensagens de texto, notificações, lembretes e respostas por comando de voz — funcionalidades particularmente vantajosas para pessoas com deficiência visual.

Além disso, esses aplicativos podem ser aproveitados para treinamentos e capacitações, oferecendo orientações, dicas e feedbacks em áudio durante o aprendizado ou a prática de novas habilidades, tornando o processo mais dinâmico e acessível.

Por fim, a empresa também pretende incluir no aplicativo alertas de segurança, garantindo notificações rápidas em situações críticas, como emergências, condições climáticas adversas, ou outras informações urgentes, reforçando o bem-estar e a segurança dos funcionários.