



Estácio

Estácio - Mundo 4 - Missão Nível 3

Faculdade Estácio - Polo Itaipava - Petrópolis/RJ.

Curso: Desenvolvimento Full Stack.

Disciplina: Nível 3: Lidando com sensores em dispositivos móveis
RPG0025.

Semestre Letivo: 4.

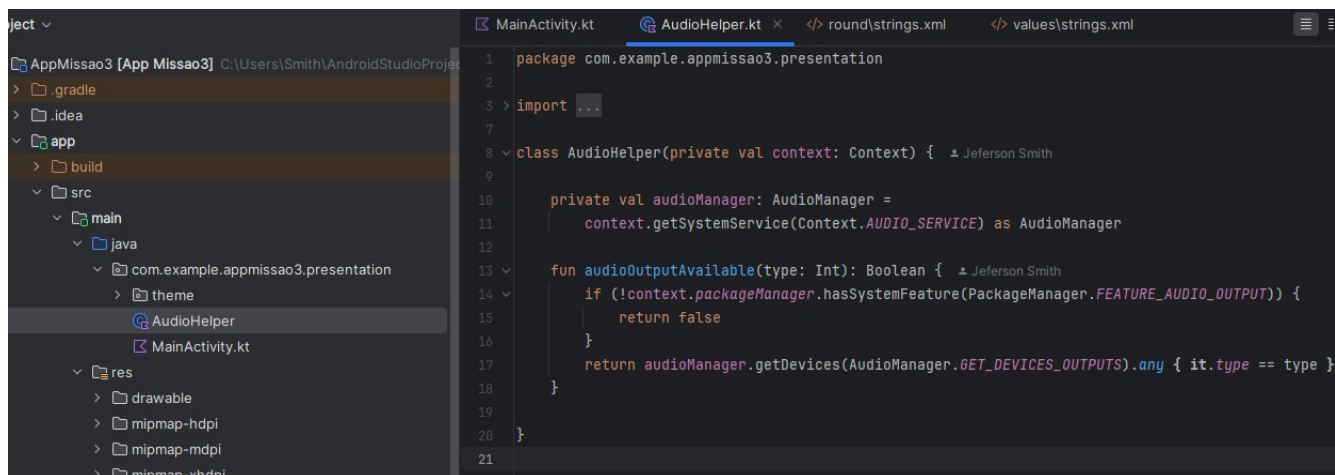
Integrante: Jeferson Jones Smith da Rocha.

Repositório: <https://github.com/JefersonSmith/estacio-mundo4-nivel3>

IDE Utilizada: Android Studio.

Audio Helper

Essa classe `AudioHelper` verifica se há uma saída de áudio disponível no dispositivo. Ela utiliza o `AudioManager` para obter informações sobre os dispositivos de saída de áudio conectados e verifica se o tipo de saída de áudio especificado está presente. Além disso, faz uma checagem inicial se o dispositivo suporta a funcionalidade de saída de áudio.



MainActivity

Essa classe `MainActivity` de um aplicativo Wear OS gerencia conexões de áudio, detectando dispositivos Bluetooth A2DP. Ao conectar ou desconectar um fone de ouvido Bluetooth, ela exibe um `Toast` para o usuário e, ao desconectar, também abre as configurações de Bluetooth. A interface do aplicativo exibe um texto centralizado e a hora usando o `WearApp` com o tema do aplicativo.

Este aplicativo Wear OS monitora dispositivos de áudio Bluetooth, especificamente fones de ouvido Bluetooth A2DP. Ele exibe uma notificação (Toast) para o usuário quando um fone de ouvido Bluetooth é conectado ou desconectado. No caso de desconexão, o aplicativo também abre automaticamente as configurações de Bluetooth do dispositivo para facilitar a reconexão. O monitoramento dos dispositivos de áudio é feito por meio de um `AudioDeviceCallback`, que detecta eventos de adição e remoção de dispositivos. A interface do usuário, construída com Jetpack Compose, apresenta um relógio na parte superior da tela e um texto de saudação centralizado, exibindo "Hello, World", estilizado com um tema personalizado. O aplicativo também implementa uma splash screen ao iniciar, e define o tema visual para o padrão do Wear OS.

```
1 package com.example.appmissao3.presentation
2
3 > import ...
31
32 class MainActivity : ComponentActivity() {
33
34     private lateinit var audioManager: AudioManager
35     private lateinit var audioHelper: AudioHelper
36
37     override fun onCreate(savedInstanceState: Bundle?) {
38         installSplashScreen()
39         super.onCreate(savedInstanceState)
40
41         setTheme(android.R.style.Theme_DeviceDefault)
42
43         audioManager = getSystemService(Context.AUDIO_SERVICE) as AudioManager
44         audioHelper = AudioHelper(context = this)
45
46         val audioDeviceCallback = object : AudioDeviceCallback() {
47             override fun onAudioDevicesAdded(addedDevices: Array<out AudioDeviceInfo>?) {
48                 super.onAudioDevicesAdded(addedDevices)
49                 if (audioHelper.audioOutputAvailable(AudioDeviceInfo.TYPE_BLUETOOTH_A2DP)) {
50                     showToast("Fone de ouvido Bluetooth conectado")
51                 }
52             }
53
54             override fun onAudioDevicesRemoved(removedDevices: Array<out AudioDeviceInfo>?) {
55                 super.onAudioDevicesRemoved(removedDevices)
56                 if (removedDevices != null) {
57                     for (device in removedDevices) {
58                         if (device.type == AudioDeviceInfo.TYPE_BLUETOOTH_A2DP) {
59                             promptBluetoothConnection()
60                             showToast("Fone de ouvido Bluetooth desconectado")
61                         }
62                     }
63                 }
64             }
65         }
66
67         audioManager.registerAudioDeviceCallback(audioDeviceCallback, handler = null)
```

```

        setContent {
            WearApp( greetingName: "Android")
        }
    }

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

    private fun promptBluetoothConnection() { ⚡ Jefferson Smith
        val intent = Intent(Settings.ACTION_BLUETOOTH_SETTINGS)
        intent.addFlags( flags: Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TA
        startActivity(intent)
    }
}

@Composable ⚡ Jefferson Smith
fun WearApp(greetingName: String) {
    Appmissao3Theme {
        Box(
            modifier = Modifier
                .fillMaxSize()
                .background(MaterialTheme.colors.background),
            contentAlignment = Alignment.Center
        ) {
            TimeText()
            Greeting(greetingName = greetingName)
        }
    }
}

@Composable ⚡ Jefferson Smith
fun Greeting(greetingName: String) {
    Text(
        modifier = Modifier.fillMaxWidth(),
        textAlign = TextAlign.Center,
        color = MaterialTheme.colors.primary,
        text = stringResource(id = "Conecte seu áudio via Bluetooth!")
    )
}

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

```

Executando o aplicativo

