



Estácio

Missão Prática | Nível 3 | Mundo 4

RPG0025 - Lidando com sensores em dispositivos móveis

Aluna: Simone Ramos de Jesus.

Matricula: 202208290965.

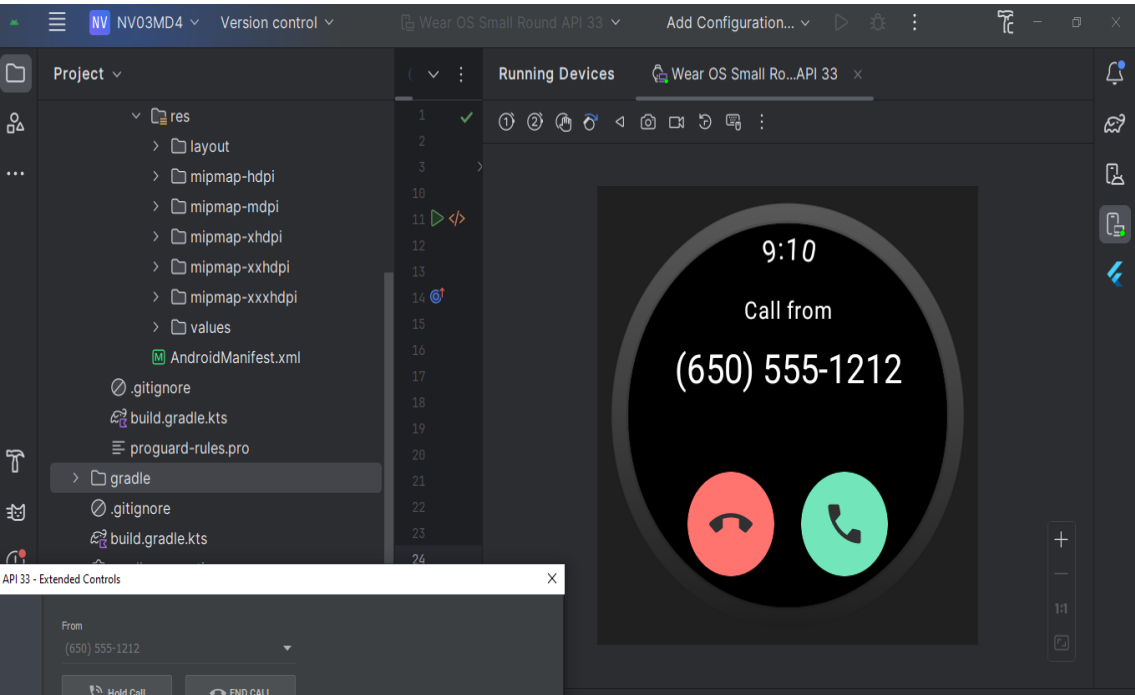
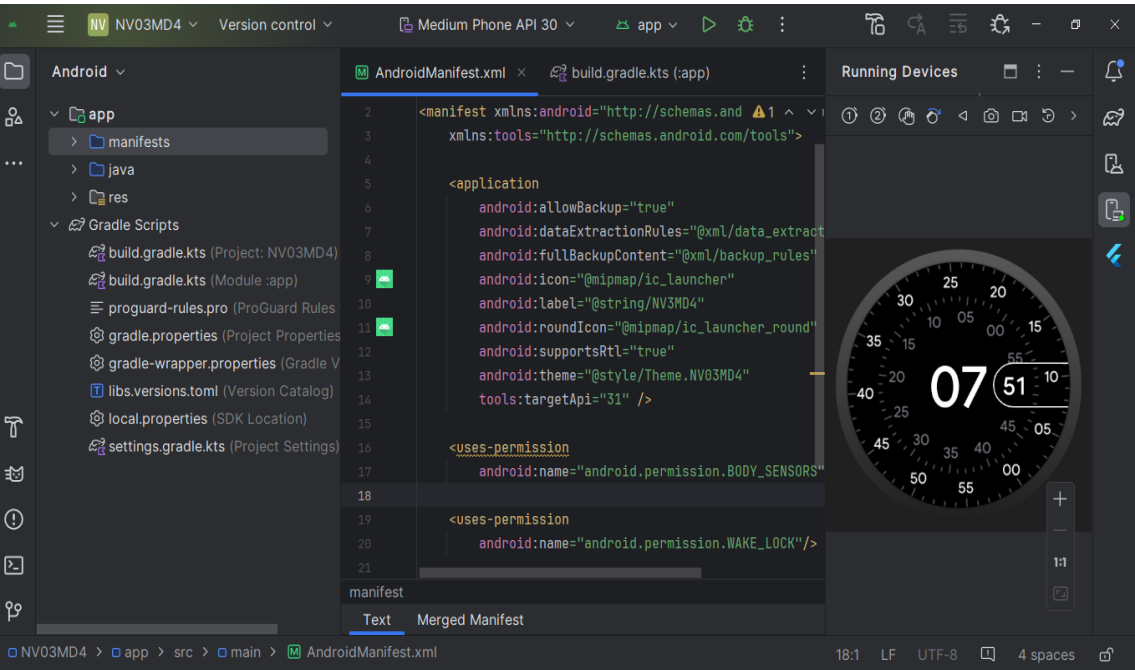
Curso: Desenvolvimento Full Stack.

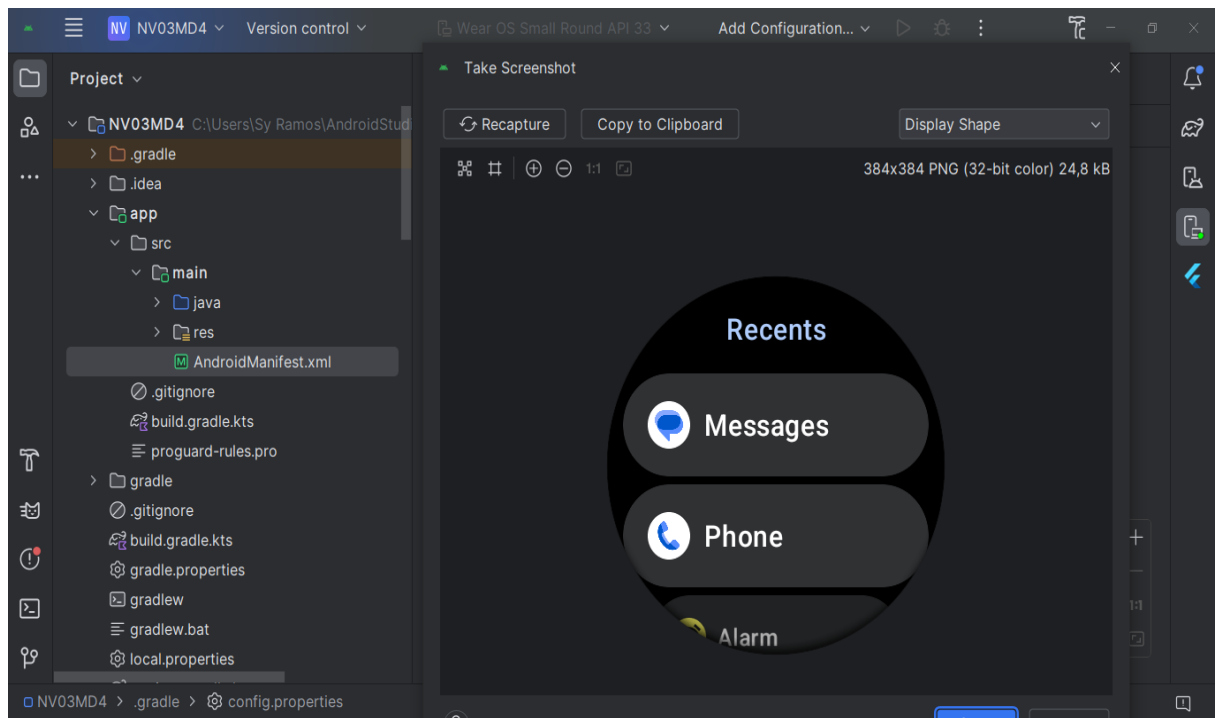
Campus: Polo Prado – Belo Horizonte – MG.

Objetivos da prática:

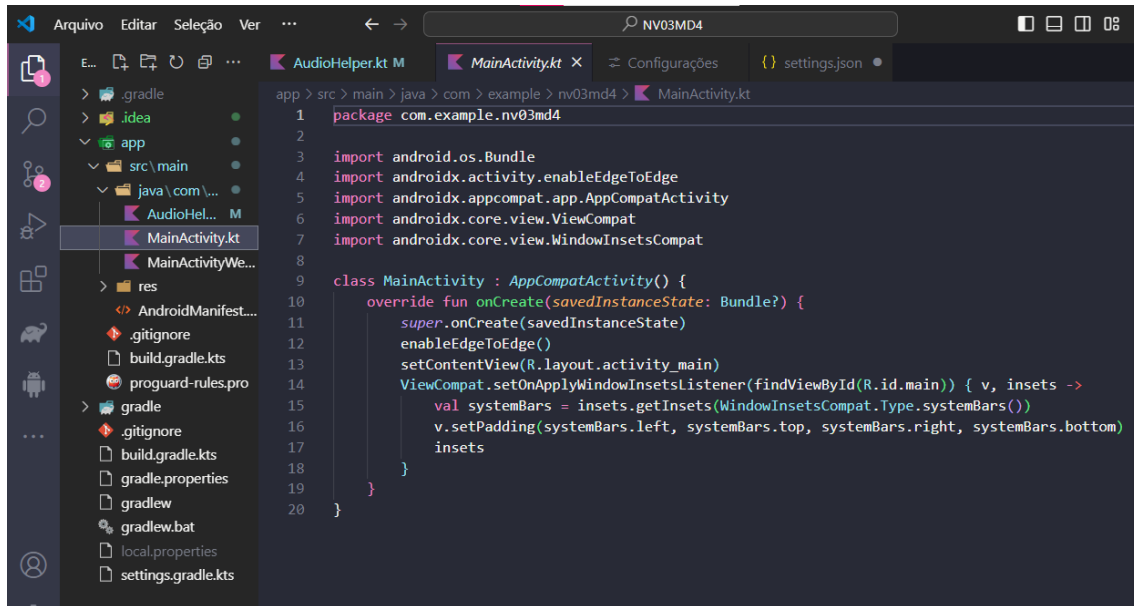
- Instalação do Android Studio e do emulador;
- Criar um app para Wear OS;
- Executar um app no emulador;
- Fazer capturas de telas no Android Studio;
- Fazer capturas de telas com app complementar.

Android Studio:



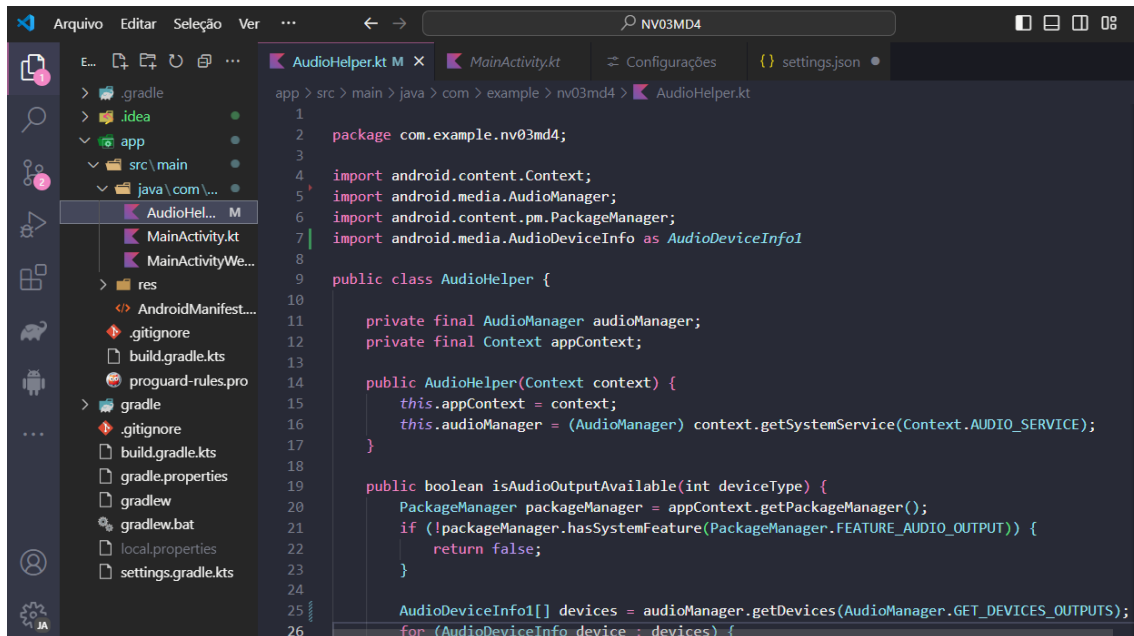


Visual Studio Code:



The screenshot shows the Visual Studio Code interface with the 'MainActivity.kt' file open. The file is located in the 'app > src > main > java > com > example > nv03md4' package. The code defines the 'MainActivity' class, which is a subclass of 'AppCompatActivity'. It overrides the 'onCreate' method, calling 'super.onCreate', enabling edge-to-edge, setting the content view to 'R.layout.activity_main', and setting up window insets for the main view.

```
1 package com.example.nv03md4
2
3 import android.os.Bundle
4 import androidx.activity.enableEdgeToEdge
5 import androidx.appcompat.app.AppCompatActivity
6 import androidx.core.view.ViewCompat
7 import androidx.core.view.WindowInsetsCompat
8
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         enableEdgeToEdge()
13         setContentView(R.layout.activity_main)
14         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
15             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
16             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
17             insets
18         }
19     }
20 }
```



The screenshot shows the Visual Studio Code interface with the 'AudioHelper.kt' file open. The file is located in the 'app > src > main > java > com > example > nv03md4' package. The code defines the 'AudioHelper' class, which is a public class. It has two private final fields: 'audioManager' of type 'AudioManager' and 'appContext' of type 'Context'. It has a constructor that takes a 'Context' object and initializes 'appContext' and 'audioManager'. It also has a public method 'isAudioOutputAvailable' that takes an 'int' device type and returns a 'boolean'. The method checks if the package manager has the 'FEATURE_AUDIO_OUTPUT' feature. If not, it returns 'false'. Otherwise, it gets the audio devices from the 'audioManager' and iterates over them.

```
1 package com.example.nv03md4;
2
3 import android.content.Context;
4 import android.media.AudioManager;
5 import android.content.pm.PackageManager;
6 import android.media.AudioDeviceInfo as AudioDeviceInfo1
7
8 public class AudioHelper {
9
10     private final AudioManager audioManager;
11     private final Context appContext;
12
13     public AudioHelper(Context context) {
14         this.appContext = context;
15         this.audioManager = (AudioManager) context.getSystemService(Context.AUDIO_SERVICE);
16     }
17
18     public boolean isAudioOutputAvailable(int deviceType) {
19         PackageManager packageManager = appContext.getPackageManager();
20         if (!packageManager.hasSystemFeature(PackageManager.FEATURE_AUDIO_OUTPUT)) {
21             return false;
22         }
23
24         AudioDeviceInfo1[] devices = audioManager.getDevices(AudioManager.GET_DEVICES_OUTPUTS);
25         for (AudioDeviceInfo device : devices) {
26
```

