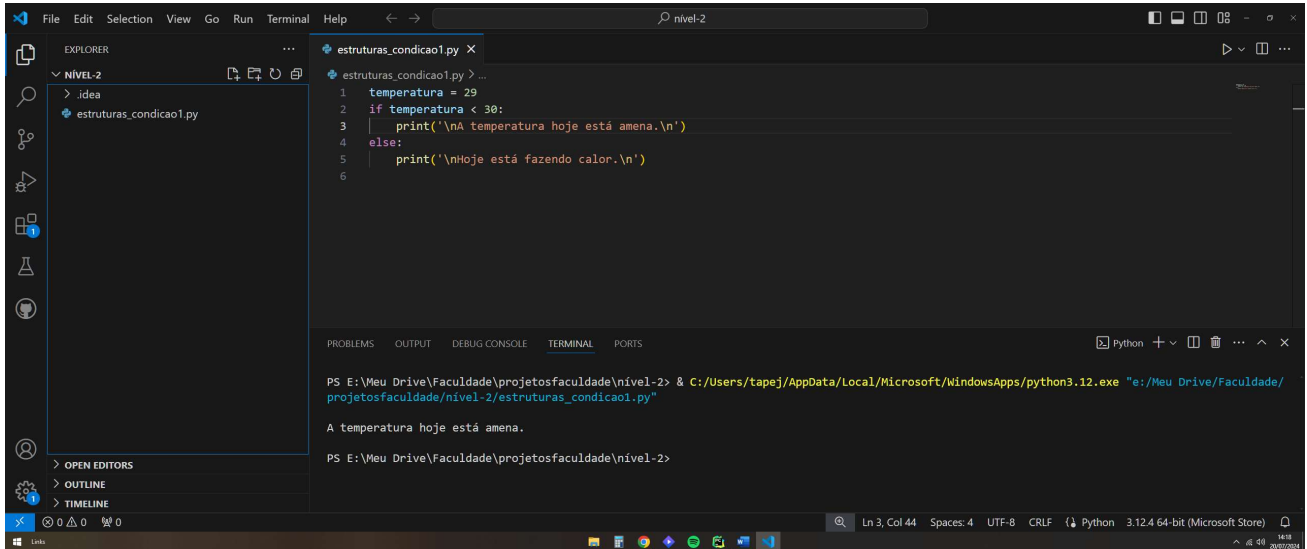


# Missão Prática - Nível 2 – Mundo 1

## Microatividade 1:

## Utilização das estruturas de condição if e else em Python



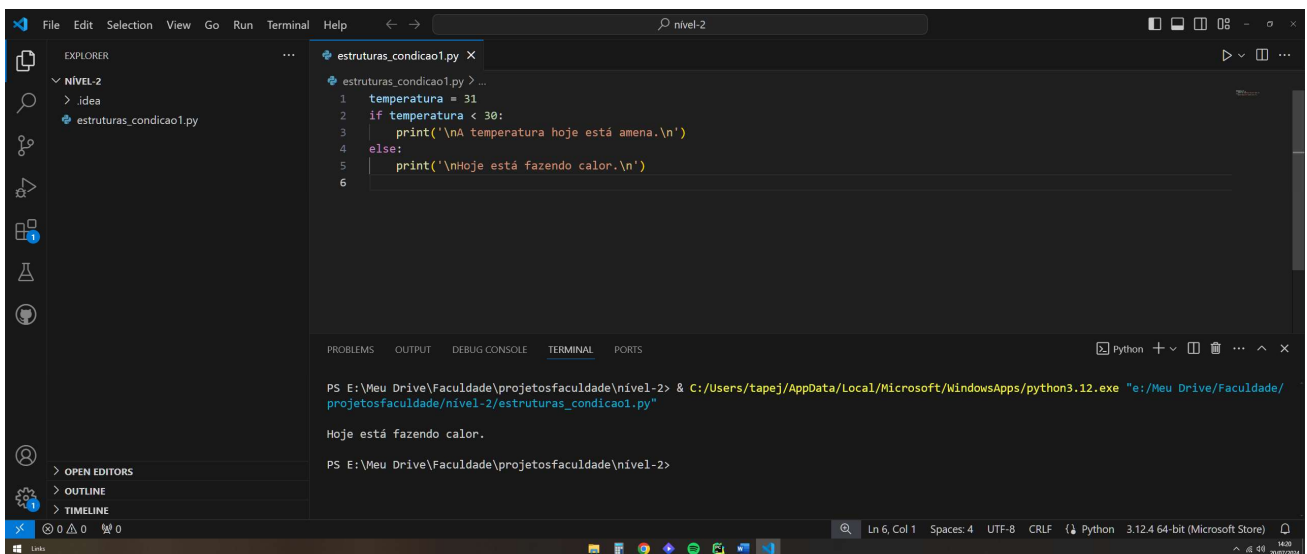
The screenshot shows the Visual Studio Code editor with a file named `estruturas_condicao1.py` open. The code defines a variable `temperatura = 29` and uses an `if` statement to check if it is less than 30. If true, it prints "A temperatura hoje está amena.", otherwise it prints "Hoje está fazendo calor.". The terminal at the bottom shows the command to run the script and the resulting output.

```
1 temperatura = 29
2 if temperatura < 30:
3     print('\nA temperatura hoje está amena.\n')
4 else:
5     print('\nHoje está fazendo calor.\n')
6
```

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_condicao1.py"

A temperatura hoje está amena.

PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```



This screenshot shows the same code as the previous one, but with the value of `temperatura` changed to 31. The terminal output now reflects this change, printing "Hoje está fazendo calor.".

```
1 temperatura = 31
2 if temperatura < 30:
3     print('\nA temperatura hoje está amena.\n')
4 else:
5     print('\nHoje está fazendo calor.\n')
6
```

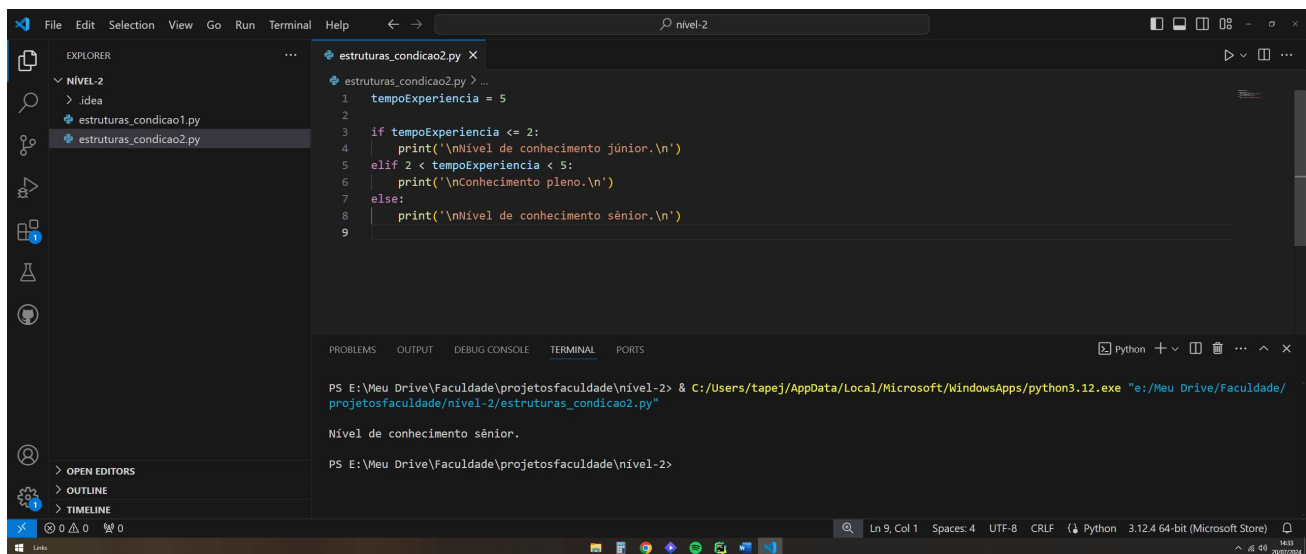
```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_condicao1.py"

Hoje está fazendo calor.

PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

## Microatividade 2:

### Utilização da estrutura de condição else if (elif) em Python



The screenshot shows the Visual Studio Code editor with a file named `estruturas_condicao2.py` open. The code defines a variable `tempoExperiencia` with the value 5 and uses an `if-elif-else` structure to determine the knowledge level. The terminal output shows the result of running the script.

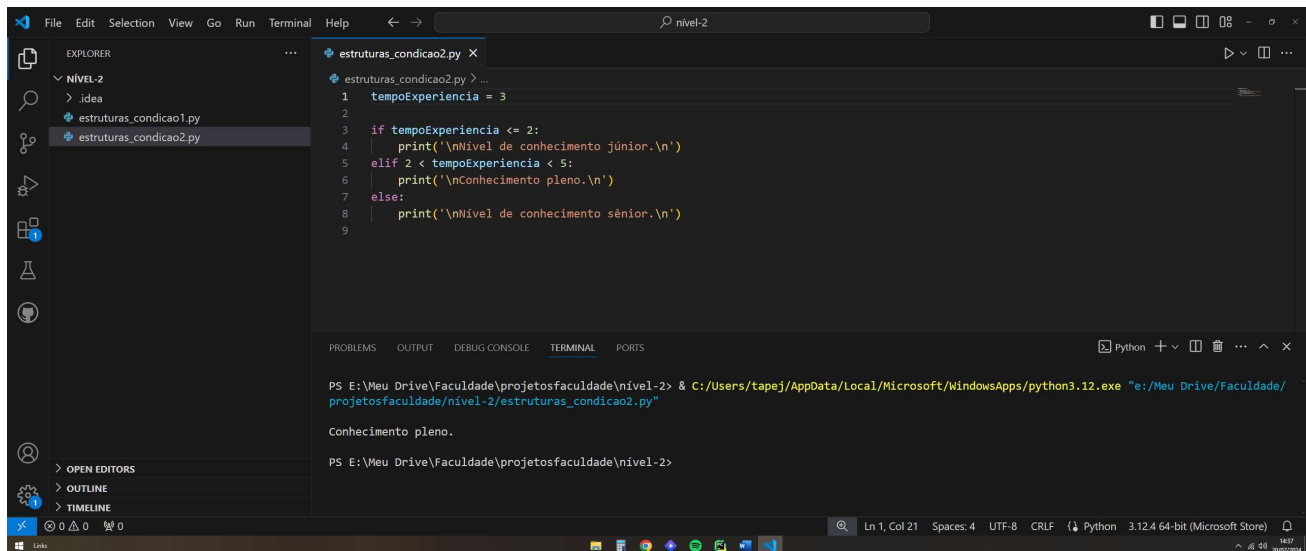
```
1 tempoExperiencia = 5
2
3 if tempoExperiencia <= 2:
4     print('\nNível de conhecimento júnior.\n')
5 elif 2 < tempoExperiencia < 5:
6     print('\nConhecimento pleno.\n')
7 else:
8     print('\nNível de conhecimento sênior.\n')
9
```

Terminal Output:

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_condicao2.py"

Nível de conhecimento sênior.

PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```



The screenshot shows the same Visual Studio Code editor with the file `estruturas_condicao2.py`. In this instance, the variable `tempoExperiencia` is set to 3. The terminal output shows the result of running the script.

```
1 tempoExperiencia = 3
2
3 if tempoExperiencia <= 2:
4     print('\nNível de conhecimento júnior.\n')
5 elif 2 < tempoExperiencia < 5:
6     print('\nConhecimento pleno.\n')
7 else:
8     print('\nNível de conhecimento sênior.\n')
9
```

Terminal Output:

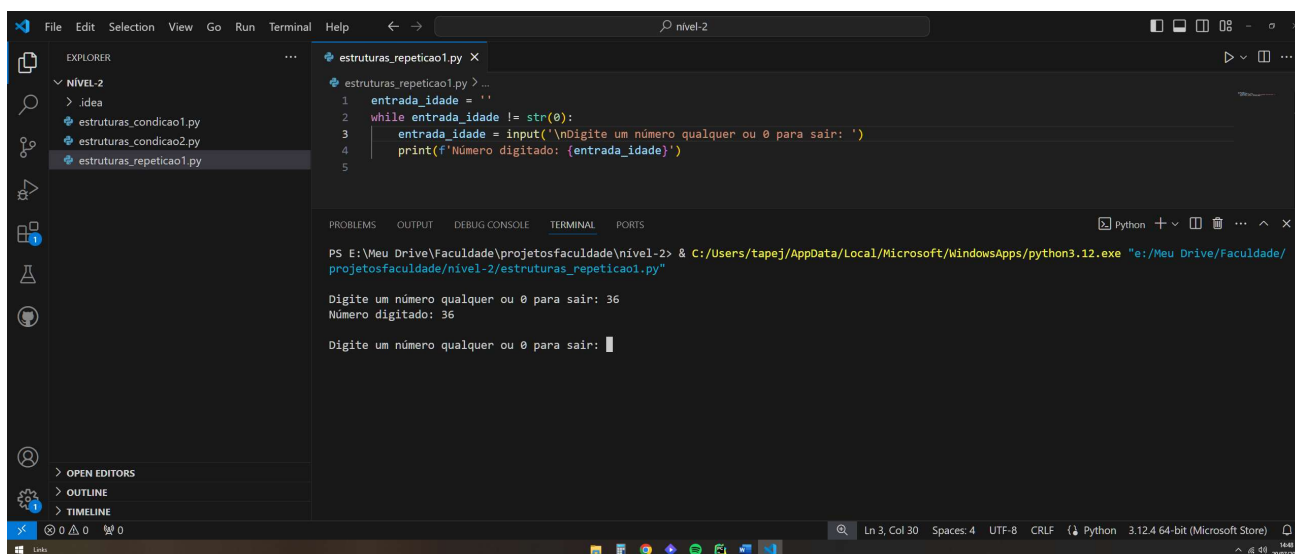
```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_condicao2.py"

Conhecimento pleno.

PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

## Microatividade 3:

### Utilização da estrutura de repetição while em Python



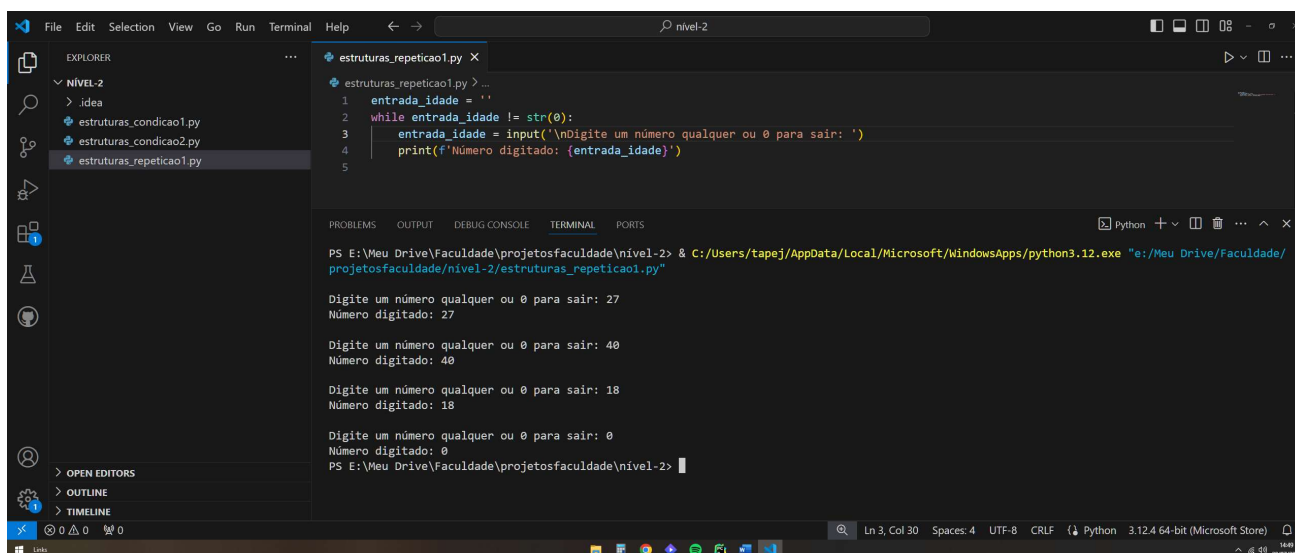
The screenshot shows the Visual Studio Code interface with a Python file named `estruturas_repeticao1.py` open. The code defines a variable `entrada_idade` and enters a `while` loop that continues as long as `entrada_idade` is not an empty string. Inside the loop, it prompts the user for input and prints the entered value. The terminal shows the command to run the script, followed by the user entering '36' and the program printing 'Número digitado: 36'.

```
estruturas_repeticao1.py
1 entrada_idade = ''
2 while entrada_idade != str(0):
3     entrada_idade = input('\nDigite um número qualquer ou 0 para sair: ')
4     print(f'Número digitado: {entrada_idade}')
5
```

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_repeticao1.py"

Digite um número qualquer ou 0 para sair: 36
Número digitado: 36

Digite um número qualquer ou 0 para sair: 
```



This screenshot shows the same code as the first image, but the terminal shows multiple iterations of the loop. The user enters '27', '40', '18', and finally '0' to exit the loop. The program prints the entered values in each iteration and then returns to the command prompt after the user enters '0'.

```
estruturas_repeticao1.py
1 entrada_idade = ''
2 while entrada_idade != str(0):
3     entrada_idade = input('\nDigite um número qualquer ou 0 para sair: ')
4     print(f'Número digitado: {entrada_idade}')
5
```

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/estruturas_repeticao1.py"

Digite um número qualquer ou 0 para sair: 27
Número digitado: 27

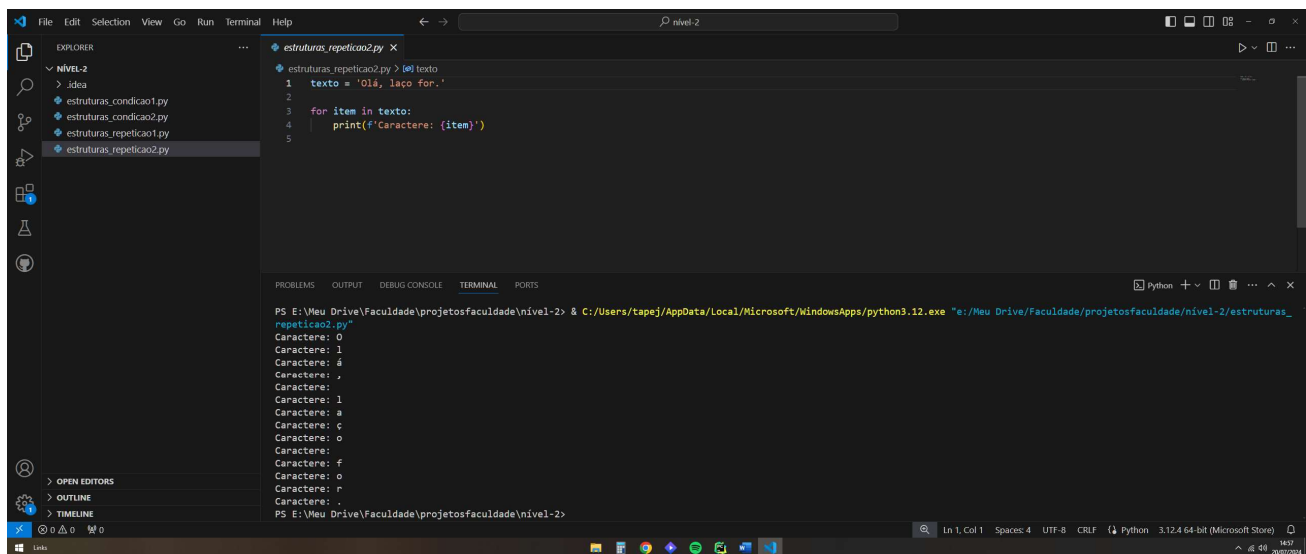
Digite um número qualquer ou 0 para sair: 40
Número digitado: 40

Digite um número qualquer ou 0 para sair: 18
Número digitado: 18

Digite um número qualquer ou 0 para sair: 0
Número digitado: 0
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

## Microatividade 4:

### Utilização da estrutura de repetição for em Python

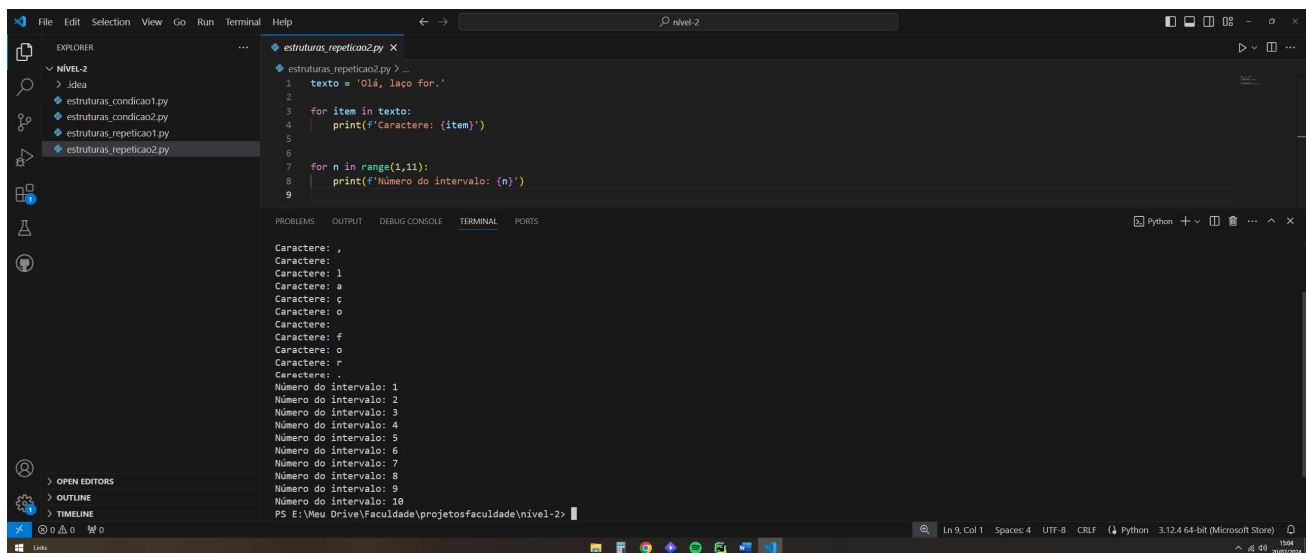


The screenshot shows a Python IDE with a file explorer on the left containing files named `estruturas_condicao1.py`, `estruturas_condicao2.py`, `estruturas_repeticao1.py`, and `estruturas_repeticao2.py`. The main editor displays the following Python code in `estruturas_repeticao2.py`:

```
1 texto = 'Olá, láço for.'
2
3 for item in texto:
4     print(f'Caractere: {item}')
5
```

The terminal at the bottom shows the execution output:

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:\Users\tapej\AppData\Local\Microsoft\WindowsApps\python3.12.exe "e:\Meu Drive\Faculdade\projetosfaculdade\nivel-2\estruturas_repeticao2.py"
Caractere: O
Caractere: l
Caractere: á
Caractere: ,
Caractere: 
Caractere: l
Caractere: a
Caractere: ç
Caractere: o
Caractere: 
Caractere: f
Caractere: o
Caractere: r
Caractere: .
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```



The screenshot shows the same Python IDE with the same file explorer. The main editor displays the following Python code in `estruturas_repeticao2.py`:

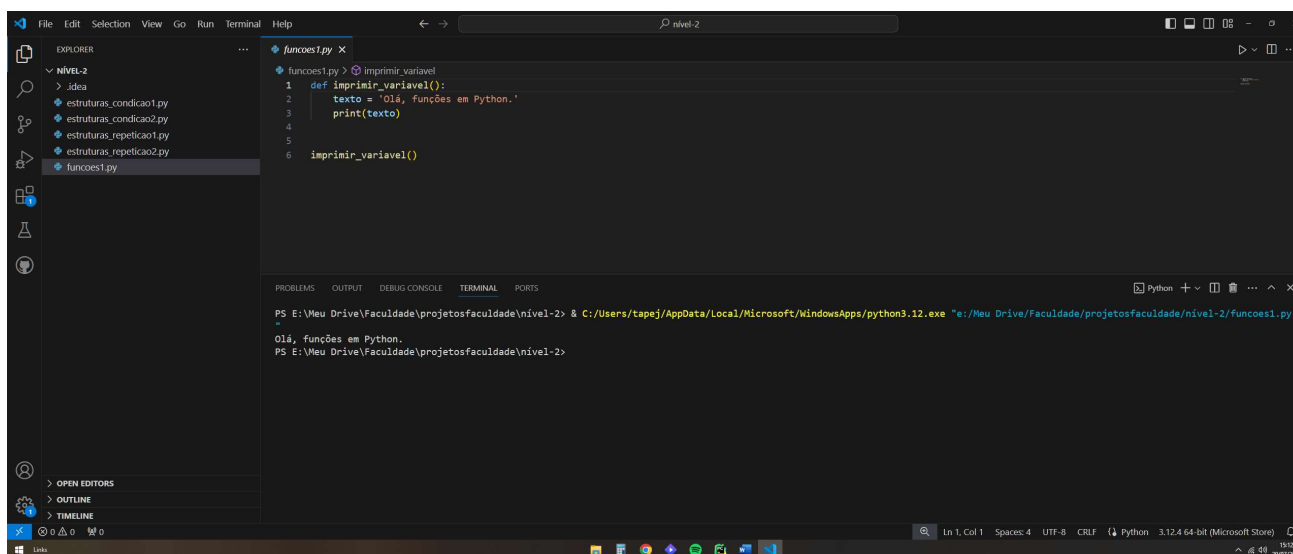
```
1 texto = 'Olá, láço for.'
2
3 for item in texto:
4     print(f'Caractere: {item}')
5
6
7 for n in range(1,11):
8     print(f'Número do intervalo: {n}')
9
```

The terminal at the bottom shows the execution output:

```
Caractere: ,
Caractere: 
Caractere: l
Caractere: a
Caractere: ç
Caractere: o
Caractere: 
Caractere: f
Caractere: o
Caractere: r
Caractere: .
Número do intervalo: 1
Número do intervalo: 2
Número do intervalo: 3
Número do intervalo: 4
Número do intervalo: 5
Número do intervalo: 6
Número do intervalo: 7
Número do intervalo: 8
Número do intervalo: 9
Número do intervalo: 10
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

## Microatividade 5:

### Utilização de funções em Python



The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying a project named 'nível-2'. The file 'funcoes1.py' is selected. The main editor window shows the code for 'funcoes1.py':

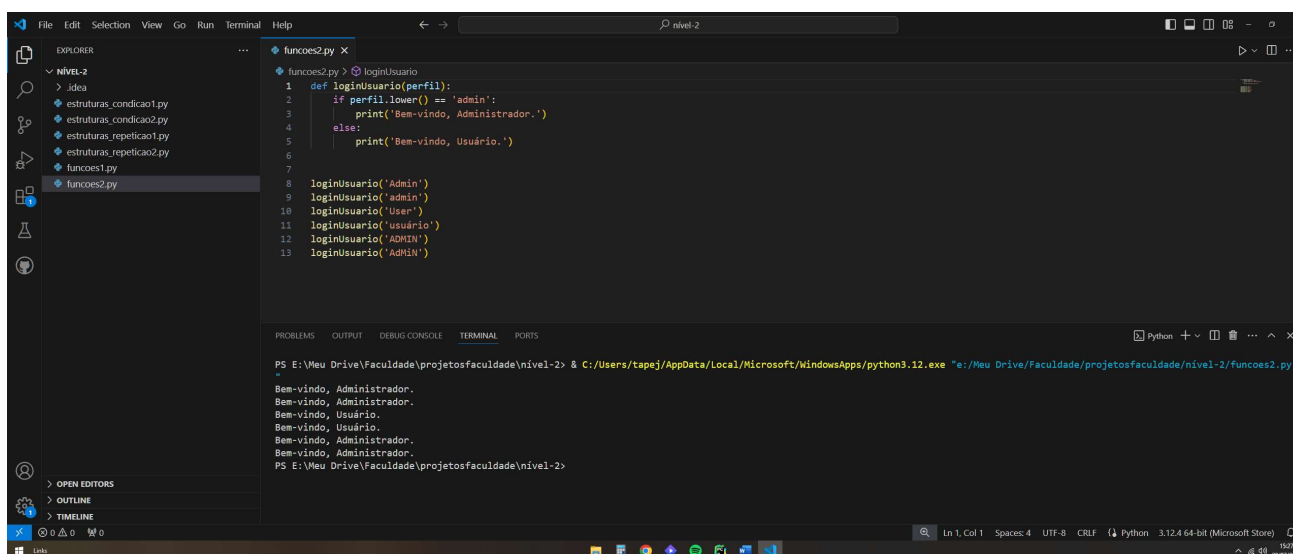
```
1 def imprimir_variavel():
2     texto = 'Olá, funções em Python.'
3     print(texto)
4
5
6 imprimir_variavel()
```

The terminal at the bottom shows the command to run the script and its output:

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/funcoes1.py"
Olá, funções em Python.
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

## Microatividade 6:

### Utilização argumentos de funções no Python



The screenshot shows the Visual Studio Code interface with a file explorer on the left displaying a project named 'nível-2'. The file 'funcoes2.py' is selected. The main editor window shows the code for 'funcoes2.py':

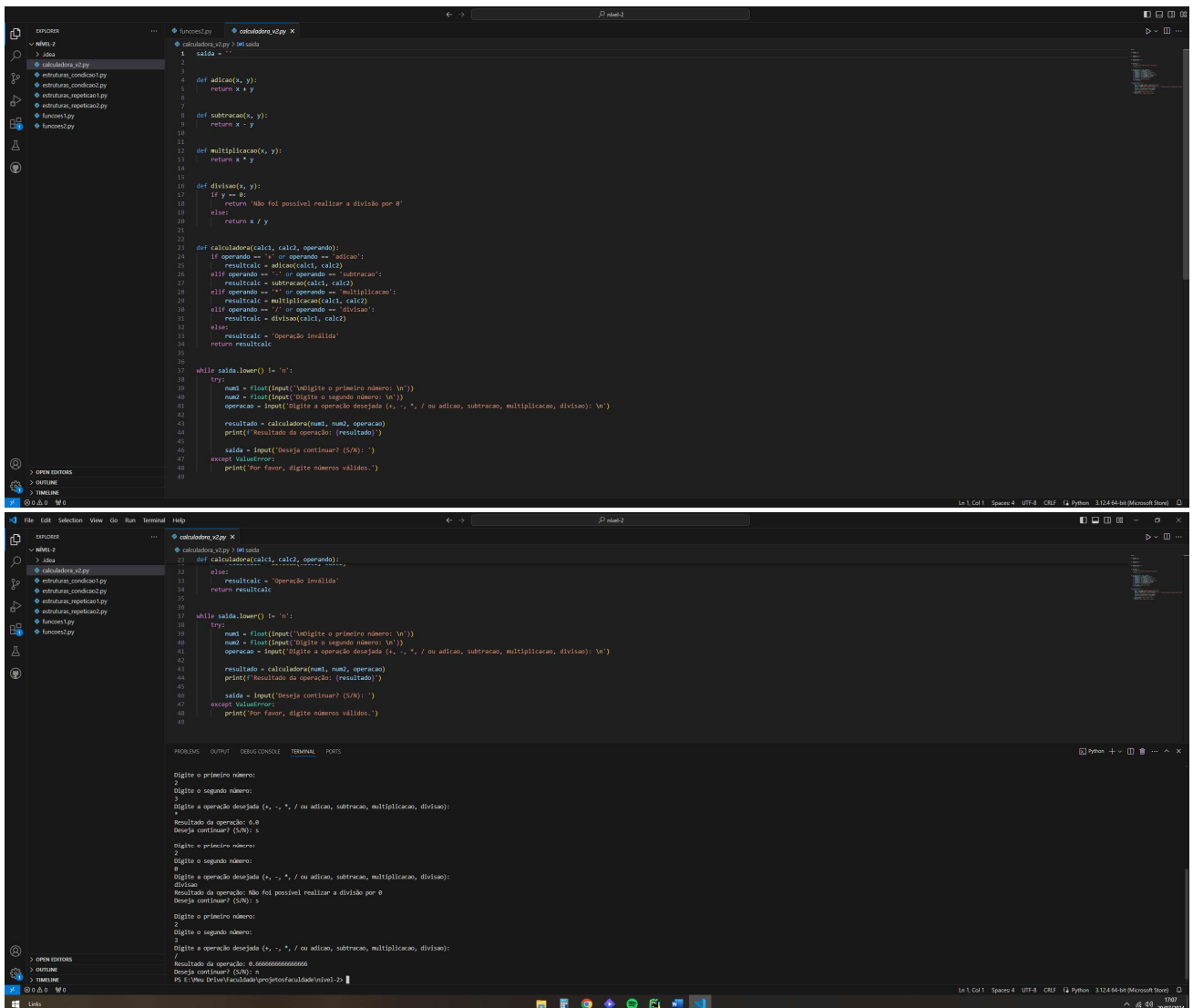
```
1 def loginUsuario(perfil):
2     if perfil.lower() == 'admin':
3         print('Bem-vindo, Administrador.')
4     else:
5         print('Bem-vindo, Usuário.')
6
7
8 loginUsuario('Admin')
9 loginUsuario('admin')
10 loginUsuario('User')
11 loginUsuario('usuário')
12 loginUsuario('ADMIN')
13 loginUsuario('AdMin')
```

The terminal at the bottom shows the command to run the script and its output:

```
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel-2/funcoes2.py"
Bem-vindo, Administrador.
Bem-vindo, Administrador.
Bem-vindo, Usuário.
Bem-vindo, Usuário.
Bem-vindo, Administrador.
Bem-vindo, Administrador.
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel-2>
```

# Missão Prática:

## Vamos Iniciar a Jornada



The image shows a VS Code editor with a Python file named `calculadora_v2.py` and its execution output in the terminal.

**Python Code:**

```
1 saida = ''
2
3
4 def adiciona(x, y):
5     return x + y
6
7
8 def subtracao(x, y):
9     return x - y
10
11
12 def multiplicacao(x, y):
13     return x * y
14
15
16 def divisao(x, y):
17     if y == 0:
18         return 'Não foi possível realizar a divisão por 0'
19     else:
20         return x / y
21
22
23 def calculadora(calcul1, calcul2, operando):
24     if operando == '+' or operando == 'adicao':
25         resultcalc = adiciona(calcul1, calcul2)
26     elif operando == '-' or operando == 'subtracao':
27         resultcalc = subtracao(calcul1, calcul2)
28     elif operando == '*' or operando == 'multiplicacao':
29         resultcalc = multiplicacao(calcul1, calcul2)
30     elif operando == '/' or operando == 'divisao':
31         resultcalc = divisao(calcul1, calcul2)
32     else:
33         resultcalc = 'Operação Inválida'
34     return resultcalc
35
36
37 while saida.lower() != 'n':
38     try:
39         num1 = float(input('Digite o primeiro número: '))
40         num2 = float(input('Digite o segundo número: '))
41         operacao = input('Digite a operação desejada (+, -, *, / ou adicao, subtracao, multiplicacao, divisao): ')
42
43         resultado = calculadora(num1, num2, operacao)
44         print('Resultado da operação: (resultado)')
45
46         saida = input('Deseja continuar? (S/N): ')
47     except ValueError:
48         print('Por favor, digite números válidos.')
49
```

**Terminal Output:**

```
Digite o primeiro número:
2
Digite o segundo número:
3
Digite a operação desejada (+, -, *, / ou adicao, subtracao, multiplicacao, divisao):
+
Resultado da operação: 5.0
Deseja continuar? (S/N): s

Digite o primeiro número:
2
Digite o segundo número:
3
Digite a operação desejada (+, -, *, / ou adicao, subtracao, multiplicacao, divisao):
-
Resultado da operação: Não foi possível realizar a divisão por 0
Deseja continuar? (S/N): s

Digite o primeiro número:
2
Digite o segundo número:
3
Digite a operação desejada (+, -, *, / ou adicao, subtracao, multiplicacao, divisao):
/
Resultado da operação: 0.6666666666666666
Deseja continuar? (S/N): n
PS E:\Des\Trabalho\aulas\projeto\faculdade\trabal\2>
```