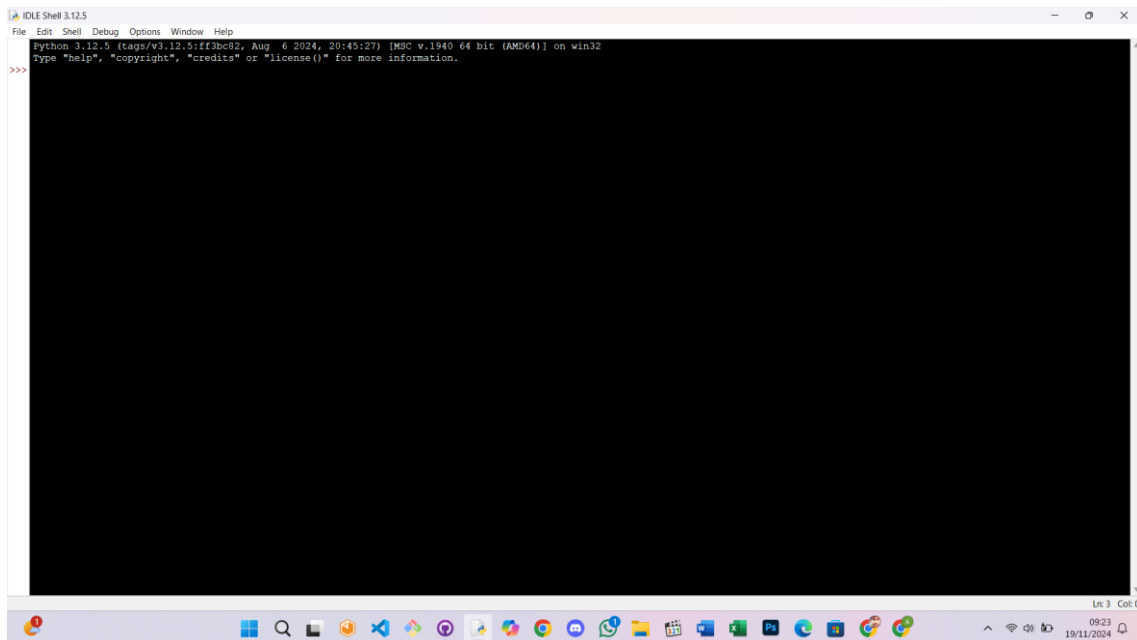
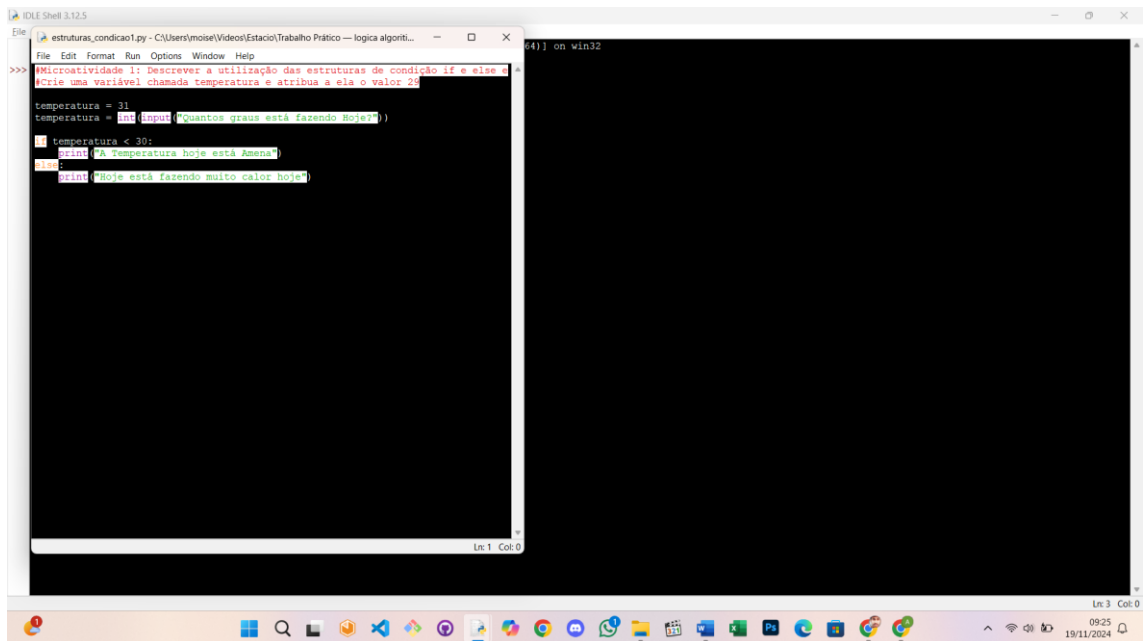
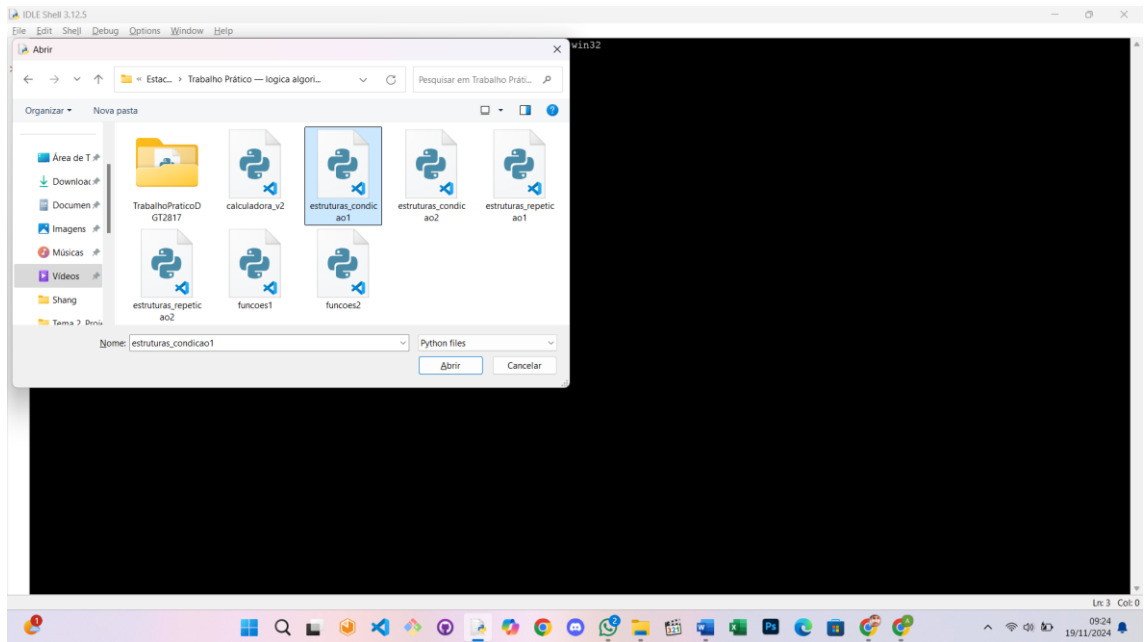


PDF Trabalho Prático | DGT2817

LÓGICA, ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Microatividade 1: Descrever a utilização das estruturas de condição if e else em Python





```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_condicao1.py =====
Quantos graus está fazendo hoje
```

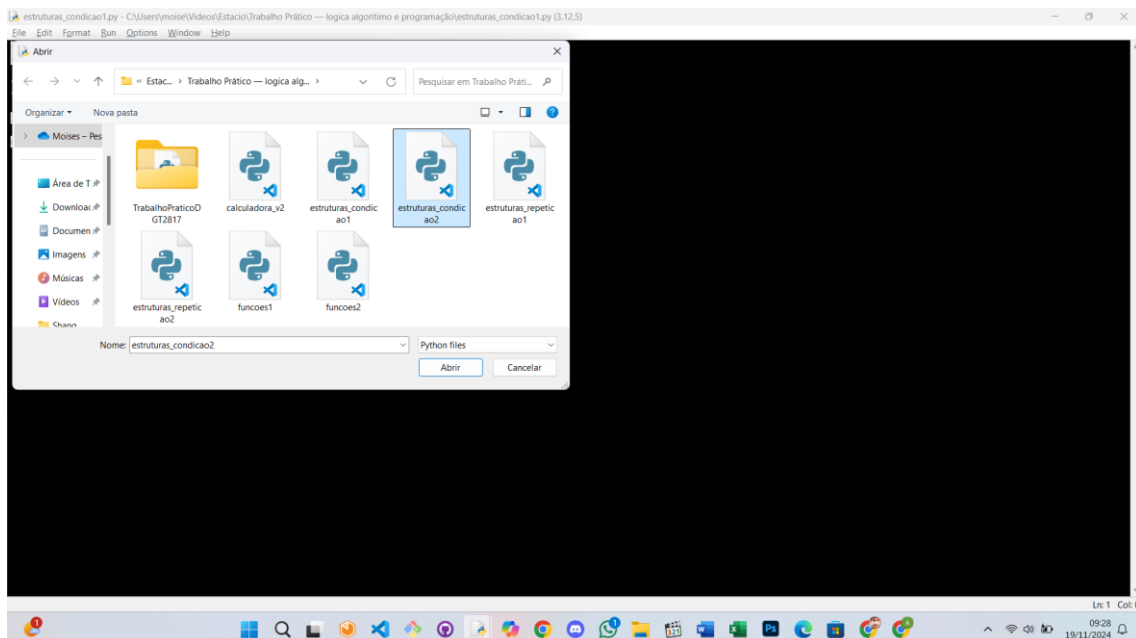
```
Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_condicao1.py =====
Quantos graus está fazendo hoje10
Hoje está fazendo muito calor hoje
```

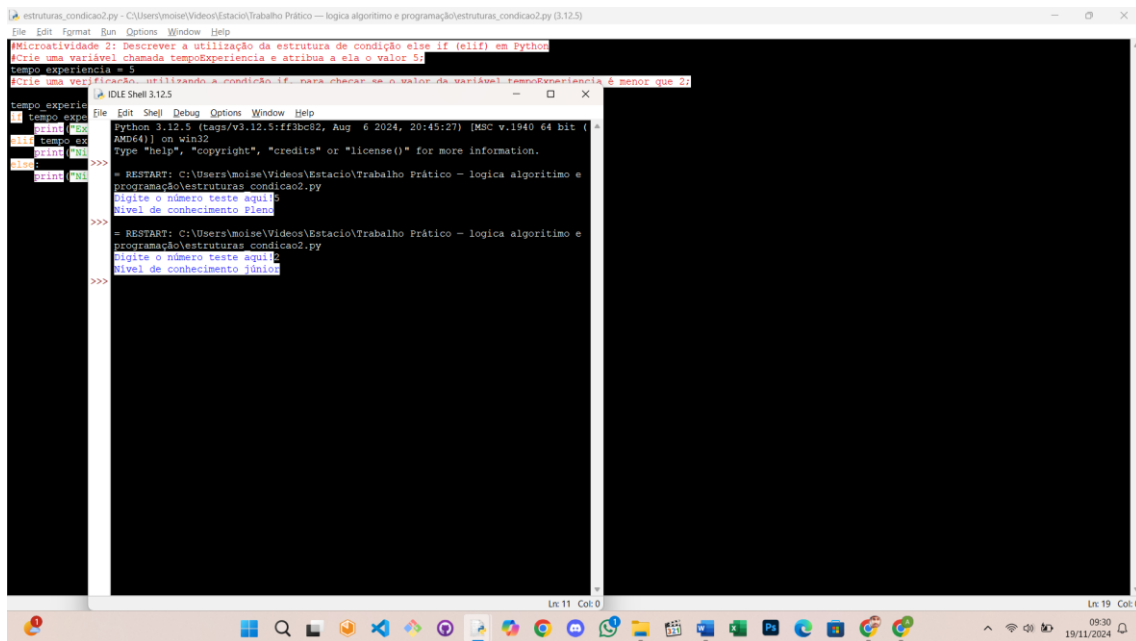
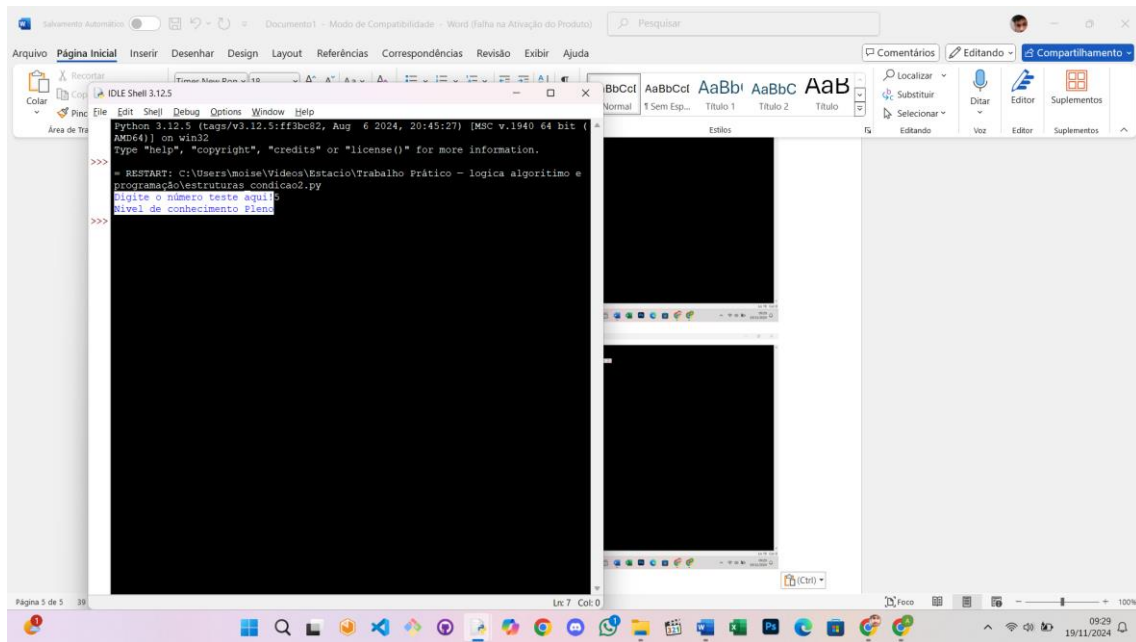
The screenshot shows a Python IDE window titled 'estruturas_condicao1.py'. The code defines a function 'temperatura' that takes a temperature value and returns a string describing the weather. The function uses conditional logic (if-elif-else) to handle different temperature ranges. The code is as follows:

```
def temperatura(temp):  
    if temp > 30:  
        return "Hoje está fazendo muito calor hoje"  
    elif temp > 20:  
        return "Quanto graus está fazendo hoje"  
    else:  
        return "A temperatura hoje está Amena"  
  
# Teste  
temp = 35  
print(temperatura(temp))
```

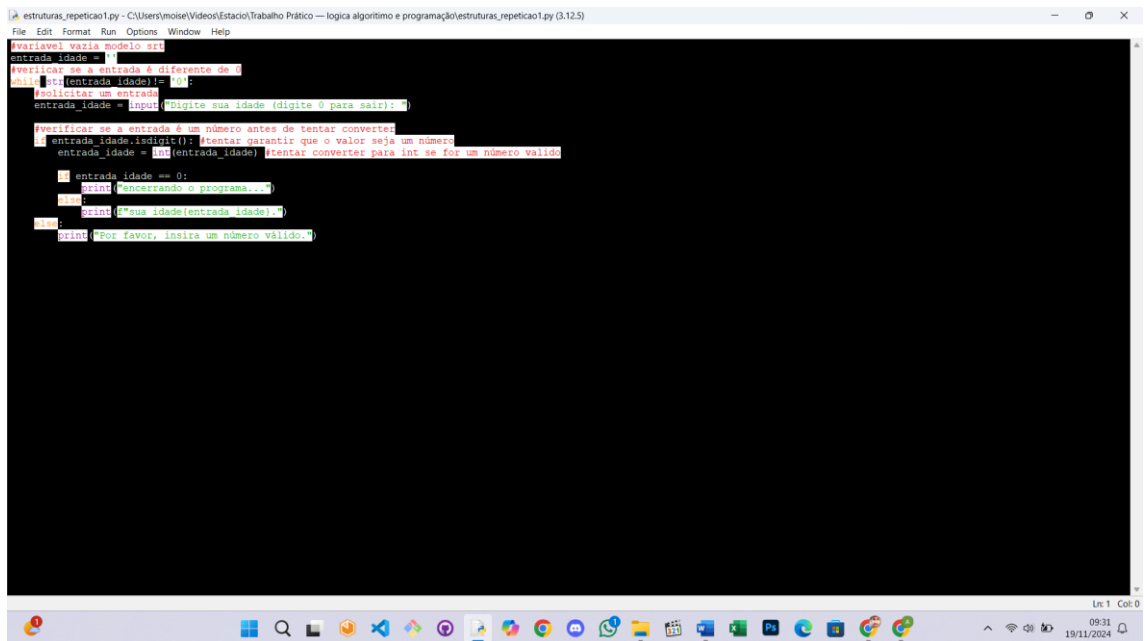
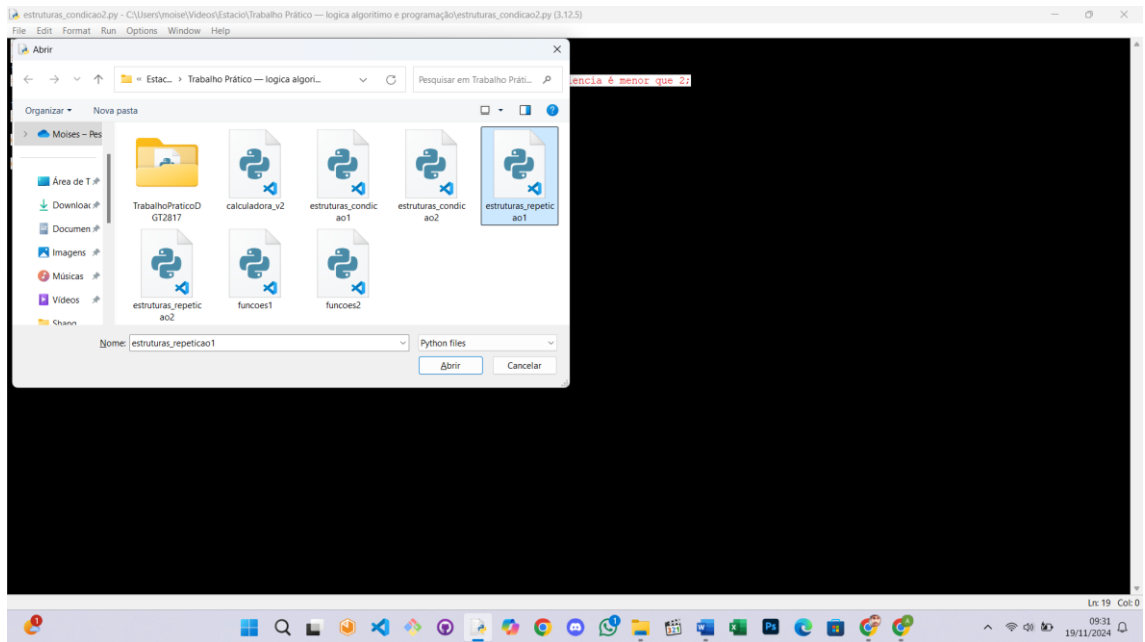
The IDE also shows a terminal window with the output of the script: "Hoje está fazendo muito calor hoje".

Microatividade 2: Descrever a utilização da estrutura de condição else if (elif) em Python





Microatividade 3: Descrever a utilização da estrutura de repetição while em Python



```
estruturas_repeticao1.py - C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_repeticao1.py (3.12.5)
File Edit Format Run Options Window Help

#variavel vazia modelo str
entrada_idade = ''
#verificar se a entrada é diferente de 0
while True:
    entrada_idade = input("Digite sua idade (digite 0 para sair): ")
    #verificar se a entrada é diferente de 0
    if entrada_idade != '0':
        #printar a idade
        print(f"Você tem {entrada_idade} anos")
    else:
        #break
        break

Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_repeticao1.py
Digite sua idade (digite 0 para sair):
Digite sua idade (digite 0 para sair):
```

```
estruturas_repeticao1.py - C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_repeticao1.py (3.12.5)
File Edit Format Run Options Window Help

#variavel vazia modelo str
entrada_idade = ''
#verificar se a entrada é diferente de 0
while True:
    entrada_idade = input("Digite sua idade (digite 0 para sair): ")
    #verificar se a entrada é diferente de 0
    if entrada_idade != '0':
        #printar a idade
        print(f"Você tem {entrada_idade} anos")
    else:
        #break
        break

Python 3.12.5 (tags/v3.12.5:ff3bc82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_repeticao1.py
Digite sua idade (digite 0 para sair): 28
sua idade 28.
Digite sua idade (digite 0 para sair):
```


The screenshot shows a Python IDE with a file named `estruturas_repicao1.py`. The code implements a loop that repeatedly asks for an age until the user enters 0. It then calculates the average of the entered ages. An IDLE Shell window is open, showing the program's execution with several test inputs and the resulting average.

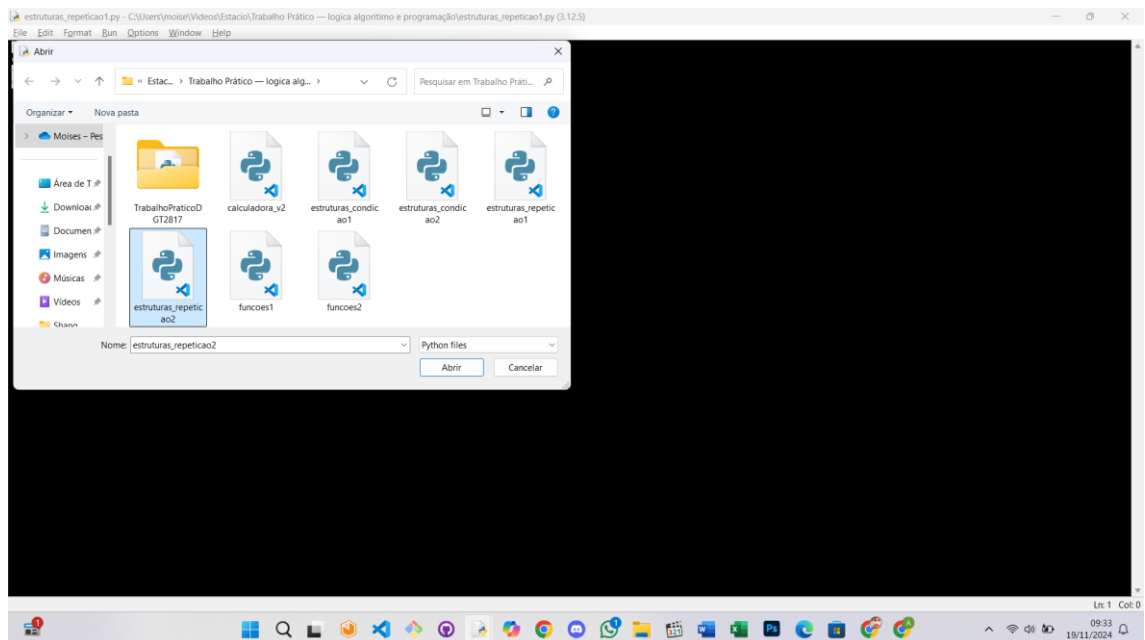
```
#!/usr/bin/env python3
# Verificar se a entrada é diferente de 0
while True:
    entrada_idade = int(input("Digite sua idade (digite 0 para sair): "))
    if entrada_idade == 0:
        break
    soma_idade += entrada_idade
    contador += 1

# Verificar se a soma é diferente de 0
if contador != 0:
    media_idade = soma_idade / contador
    print(f"A média das idades é: {media_idade}")
else:
    print("Nenhuma idade foi digitada.")
```

Execution output in the IDLE Shell:

```
Python 3.12.5 (tags/v3.12.5:ff3bc92, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
- RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\estruturas_repicao1.py
Digite sua idade (digite 0 para sair): 28
sua idade28.
Digite sua idade (digite 0 para sair): 32
sua idade32.
Digite sua idade (digite 0 para sair): 54
sua idade54.
Digite sua idade (digite 0 para sair): 18
sua idade18.
Digite sua idade (digite 0 para sair): 75
sua idade75.
Digite sua idade (digite 0 para sair): 100
sua idade100.
Digite sua idade (digite 0 para sair): 2
sua idade2.
Digite sua idade (digite 0 para sair): 0
encerrando o programa...
>>>
```

Microatividade 4: Descrever a utilização da estrutura de repetição for em Python



The image shows a code editor window with a Python script. The script is designed to demonstrate two types of loops: a 'for' loop for iterating over a string and a 'while' loop for iterating over a range of numbers. The code is as follows:

```
estruturas_repeticao2.py - C:\Users\moise\Videos\Estacio\Trabalho Físico — logica algoritmo e programação\estruturas_repeticao2.py (3,12,5)
File Edit Format Run Options Window Help

texto = "ola, lago for"

#itera sobre cada caractere de texto
for item in texto:
    print(caractere: " + item)
#parte 2: iterar sobre um intervalo numérico

#itera sobre o intervalo de números de 1 a 11
for numero in range(1, 11):
    print("Número do intervalo: " + str(numero))
```

The code is written in a dark-themed editor. The first part uses a 'for' loop to iterate over each character in the string "ola, lago for", printing each character. The second part uses a 'for' loop with the 'range' function to iterate over numbers from 1 to 10 (since range is exclusive of the end value), printing each number. The taskbar at the bottom shows various application icons and the system clock indicating 09:33 on 19/11/2024.

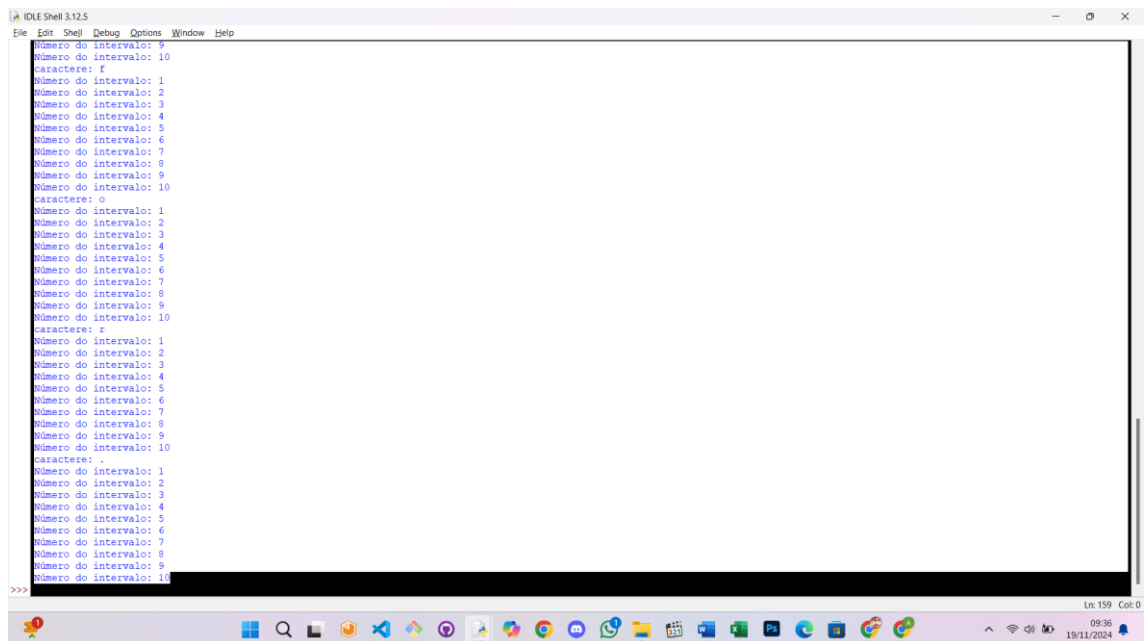
```
IDLE Shell 3.12.5
File Edit Shell Debug Options Window Help
Python 3.12.5 (tags/v12.5.1:ff3b82, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
type "help", "copyright", "credits" or "license()" for more information.
>>>
=> RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programacao\estruturas repeticao2.py
Caractere: 0
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
Caractere: 1
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
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
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
```

```

IDLE Shell 3.12.5
File Edit Shell Debug Options Window Help
Número do intervalo: 8
Número do intervalo: 9
Número do intervalo: 10
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
Caractere: l
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
Caractere: a
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
Caractere: c
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
Ln: 159 Col: 0
09:36 19/11/2024
```

```

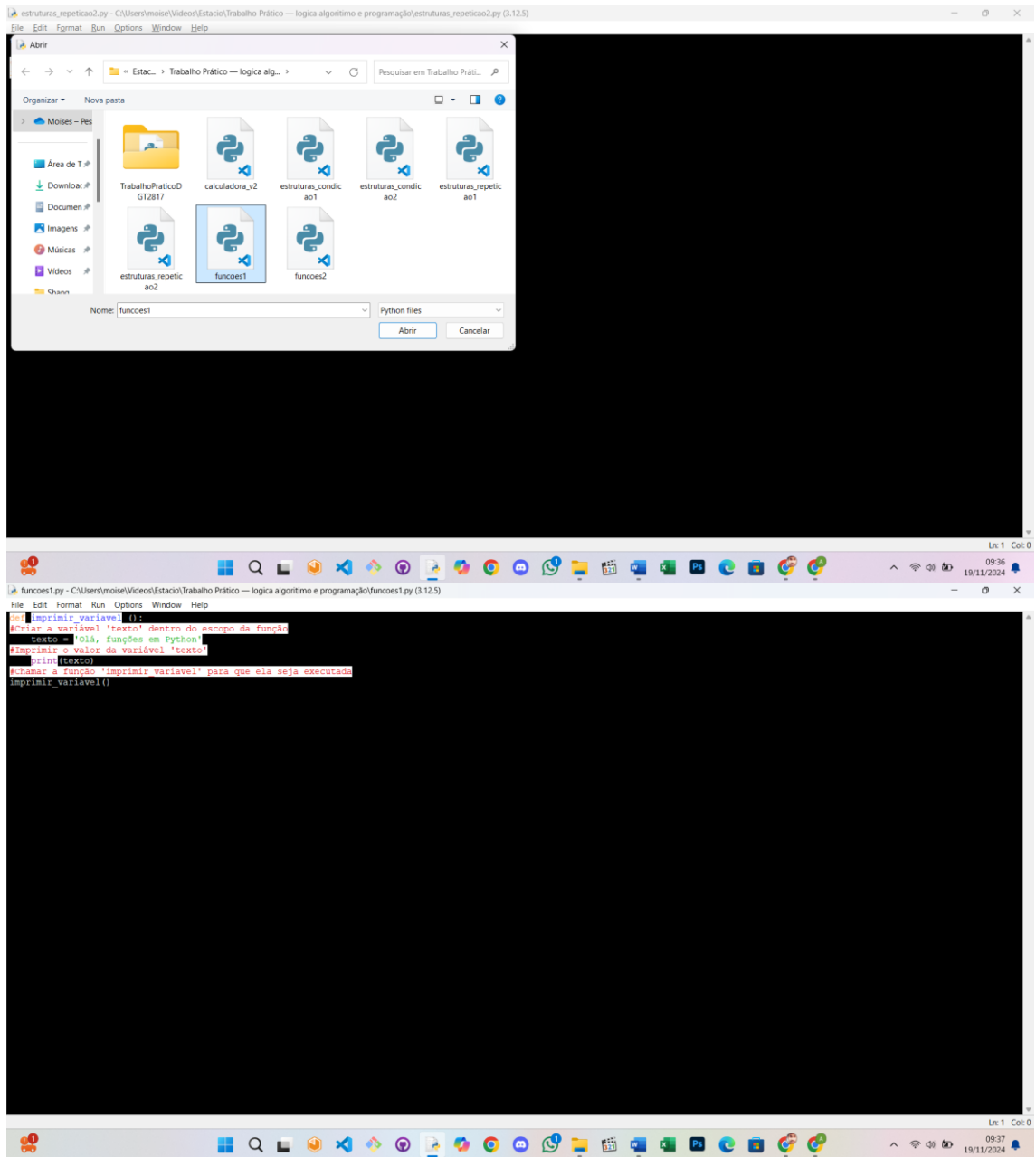
IDLE Shell 3.12.5
File Edit Shell Debug Options Window Help
Número do intervalo: 4
Número do intervalo: 5
Número do intervalo: 6
Número do intervalo: 7
Número do intervalo: 8
Número do intervalo: 9
Número do intervalo: 10
Caractere: o
Número do intervalo: 1
Número do intervalo: 2
Número do intervalo: 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
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
Caractere: f
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
Caractere: o
Número do intervalo: 1
Número do intervalo: 2
Número do intervalo: 3
Número do intervalo: 4
Número do intervalo: 5
Número do intervalo: 6
Ln: 159 Col: 0
09:36 19/11/2024
```

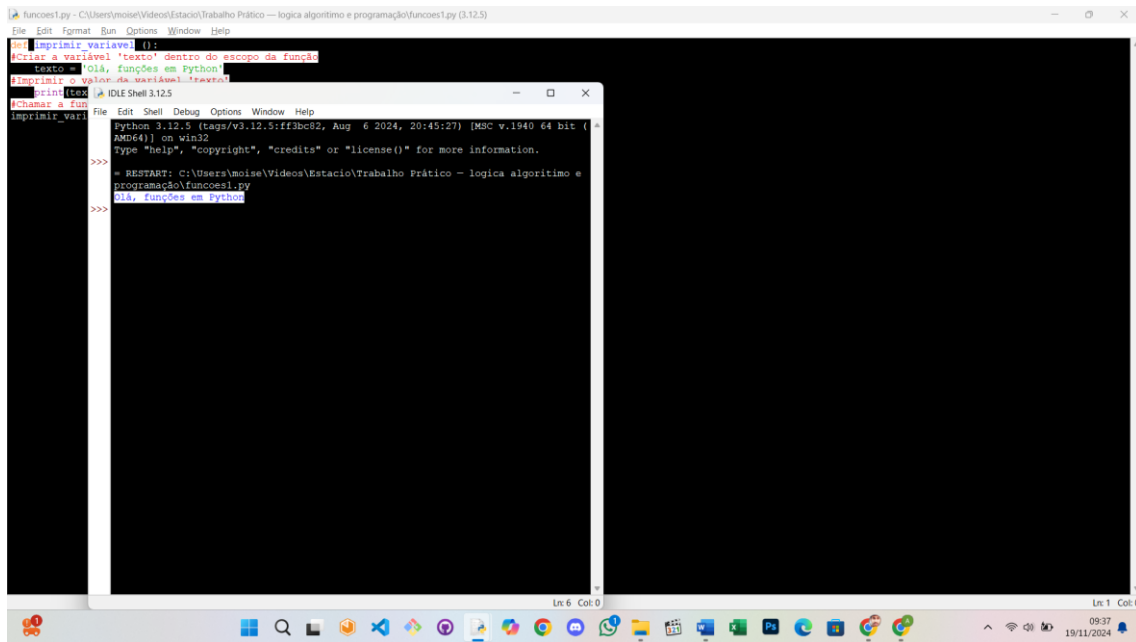


```
IDE Shell 3.12.5
File Edit Shell Debug Options Window Help
Número do intervalo: 9
Número do intervalo: 10
caractere: f
Número do intervalo: 1
Número do intervalo: 2
Número do intervalo: 3
Número do intervalo: 4
Número do intervalo: 5
Número do intervalo: 6
Número do intervalo: 7
Número do intervalo: 8
Número do intervalo: 9
Número do intervalo: 10
caractere: o
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
caractere: e
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
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
>>>
```

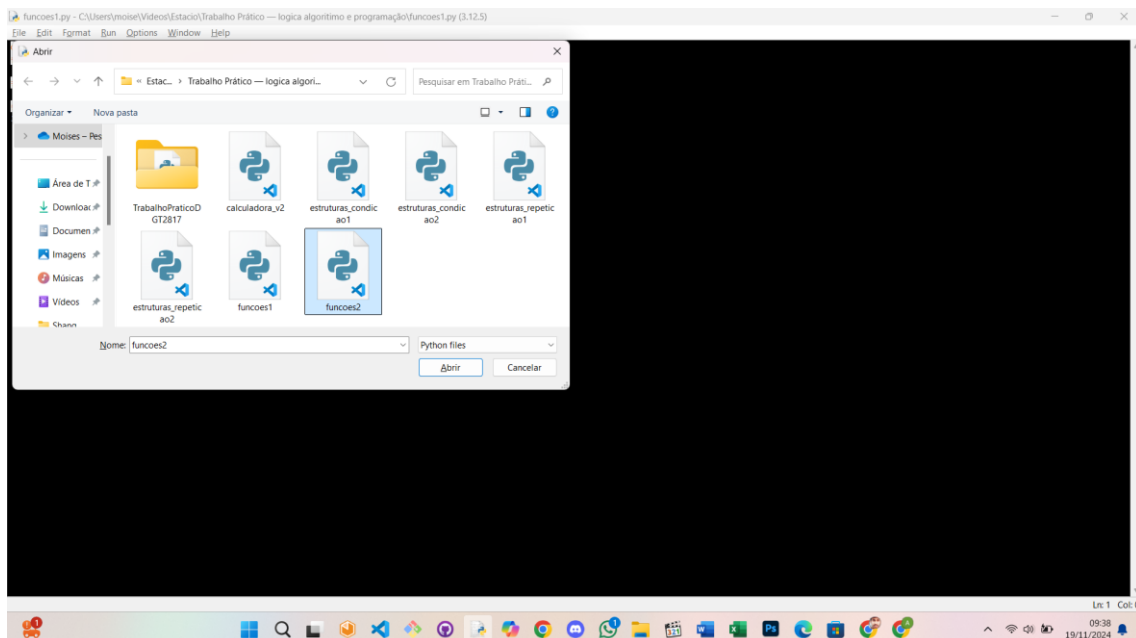
Ln 159 Col 0
09:36 19/11/2024

Microatividade 5: Descrever a utilização de funções em Python





Microatividade 6: Descrever a utilização argumentos de funções no Python



```
funcões2.py - C:\Users\moise\Videos\Estacio\Trabalho Prático — logica algoritmo e programação\funcões2.py (3.12.5)
File Edit Format Run Options Window Help
#tentar definir uma função chamada "loginUsuario" que por sua vez, receberá o parametro "perfil"
def loginUsuario(perfil):
    #tentar converter o valor do parametro para minúscula com lower()
    perfil.lower() == 'admin':
        print('Bem-Vindo, Administrador')
    else:
        print('Bem-Vindo, Usuário')
#tente que a função está passando diferentes valores como parametro
loginUsuario('Admin') #deverá tentar imprimir: 'Bem-Vindo,Administrador'
loginUsuario('admin') #tentar imprimir: 'Bem-Vindo, Administrador'
loginUsuario('User') #tentar imprimir: 'Bem-Vindo, Usuário'
loginUsuario('usuario') #tentopoderá tentar conseguir imprimir: 'Bem-Vindo, Usuário'
```

```
funcões2.py - C:\Users\moise\Videos\Estacio\Trabalho Prático — logica algoritmo e programação\funcões2.py (3.12.5)
File Edit Format Run Options Window Help
#tentar definir uma função chamada "loginUsuario" que por sua vez, receberá o parametro "perfil"
def loginUsuario(perfil):
    #tentar converter o valor do parametro para minúscula com lower()
    perfil.lower() == 'admin':
        print('Bem-Vindo, Administrador')
    else:
        print('Bem-Vindo, Usuário')
#tente que a função está passando diferentes valores como parametro
loginUsuario('Admin')
loginUsuario('Admin')
loginUsuario('User')
loginUsuario('usuario')
>>>
= RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático — logica algoritmo e
programação\funcões2.py
Bem-Vindo, Administrador
Bem-Vindo, Administrador
Bem-Vindo, Usuário
Bem-Vindo, Usuário
>>>
```

Trabalho Prático — “calculadora_v2.py”


```
calculadora_v2.py - C:\Users\moise\Videos\Estacio\Trabalho Prático — logica algoritmo e programação\calculadora_v2.py (3.12.5)
File Edit Format Run Options Window Help

saida = ''
def adicao(a, b):
    return a + b
def subtracao(a, b):
    return a - b
def multiplicacao(a, b):
    return a * b
def divisao(a, b):
    if b == 0:
        raise ValueError("Não foi possível realizar a divisão por 0")
    return a / b
# Criar uma função chamada calculadora
def calculadora(num1, num2, operacao):
    if operacao == '+':
        return adicao(num1, num2)
    elif operacao == '-':
        return subtracao(num1, num2)
    elif operacao == '*':
        return multiplicacao(num1, num2)
    elif operacao == '/':
        return divisao(num1, num2)
    else:
        return "Operação inválida"
# Criar um loop while
while True:
    # Inserir os dois números e a operação desejada
    num1 = float(input("Digite o primeiro número: "))
    num2 = float(input("Digite o segundo número: "))
    operacao = input("Digite a operação (+, -, *, /) ou o nome da operação: ")

    # Chamar a função calculadora e armazenar o resultado
    resultado = calculadora(num1, num2, operacao)

    # Calcular o resultado
    print(f"Resultado da operação: {resultado}")

    # Perguntar ao usuário se deseja continuar ou sair
    saida = input("Deseja continuar? (S/N): ")

    if saida.lower() != 's':
        break
```

```
calculadora_v2.py - C:\Users\moise\Videos\Estacio\Trabalho Prático — logica algoritmo e programação\calculadora_v2.py (3.12.5)
File Edit Shell Debug Options Window Help
Python 3.12.5 (tags/v3.12.5:ff3bcb2, Aug 6 2024, 20:45:27) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\calculadora_v2.py
Digite o primeiro número:
```

```
calculadora_v2.py - C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\calculadora_v2.py (3.12.5)
File Edit Format Run Options Window Help

saida = 0
def calcula(a, b):
    return a + b
def subtra(a, b):
    return a - b
def mult(a, b):
    return a * b
def div(a, b):
    return a / b
def main():
    while True:
        print("C - RESTART: C:\Users\moise\Videos\Estacio\Trabalho Prático - logica algoritmo e programação\calculadora_v2.py")
        print("Digite o primeiro número: ")
        num1 = input()
        print("Digite o segundo número: ")
        num2 = input()
        print("Digite a operação (+, -, *, / ou o nome da operação): ")
        oper = input()
        result = calcula(num1, num2) if oper == "+" else subtra(num1, num2) if oper == "-" else mult(num1, num2) if oper == "*" else div(num1, num2) if oper == "/" else None
        print("Resultado da operação: ", result)
        print("Deseja continuar? (S/N): ")
        cont = input()
        if cont == "N":
            break
    print("Fim do programa.")
if __name__ == '__main__':
    main()
# Calculadora
# Print
# Fim do programa
saida
```