

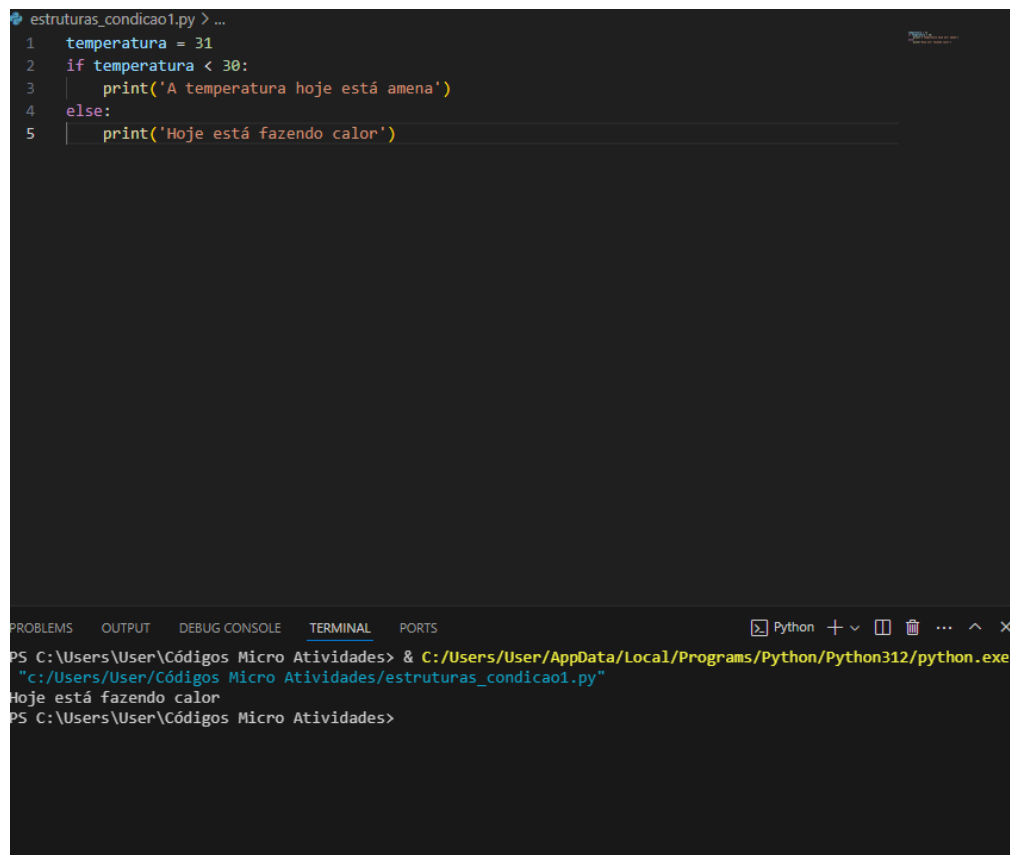
# Micro Atividades de Lógica, Algoritmos e Programação de Computadores

Micro Atividades:

- estruturas\_condicao1:

```
temperatura = 31
if temperatura < 30:
    print('A temperatura hoje está amena')
else:
    print('Hoje está fazendo calor')
```

Print 1:



The screenshot shows a code editor window with a Python script named `estruturas_condicao1.py`. The script contains the following code:

```
1 temperatura = 31
2 if temperatura < 30:
3     print('A temperatura hoje está amena')
4 else:
5     print('Hoje está fazendo calor')
```

Below the code editor is a terminal window. The terminal shows the command to run the script using Python 3.12:

```
PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/User/Códigos Micro Atividades/estruturas_condicao1.py"
```

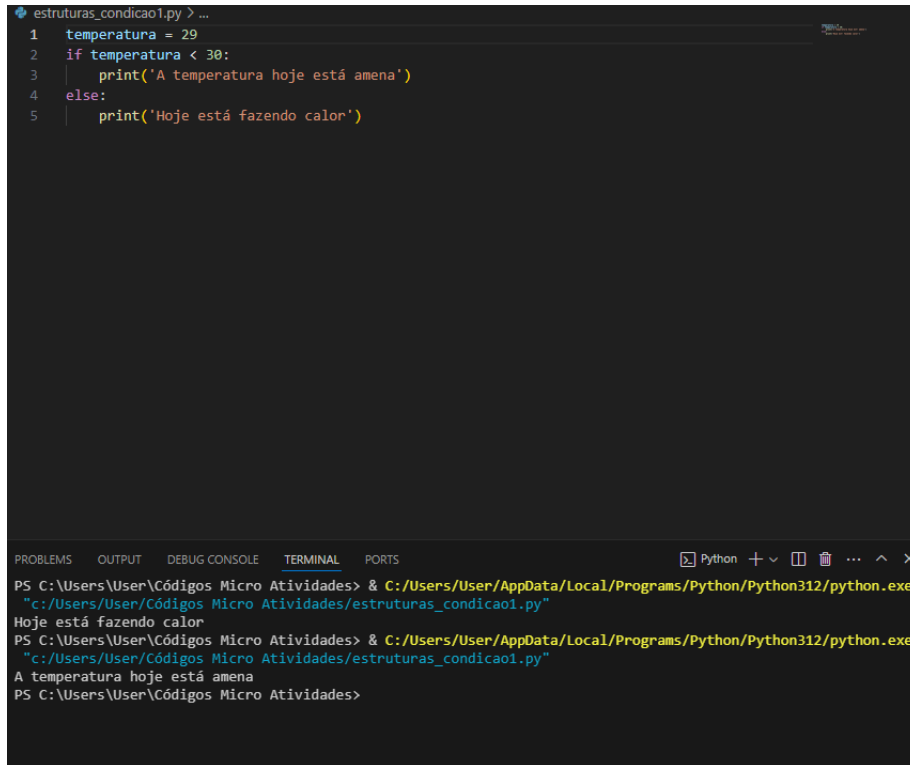
The output of the script is displayed in the terminal:

```
Hoje está fazendo calor
```

The terminal prompt is now ready for the next command:

```
PS C:\Users\User\Códigos Micro Atividades>
```

Print 2:



The screenshot shows a VS Code editor window with a file named `estruturas_condicao1.py`. The code in the editor is as follows:

```
1 temperatura = 29
2 if temperatura < 30:
3     print('A temperatura hoje está amena')
4 else:
5     print('Hoje está fazendo calor')
```

Below the editor, the TERMINAL panel is open, showing the command used to run the script and its output:

```
PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/User/Códigos Micro Atividades/estruturas_condicao1.py"
Hoje está fazendo calor
PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/User/Códigos Micro Atividades/estruturas_condicao1.py"
A temperatura hoje está amena
PS C:\Users\User\Códigos Micro Atividades>
```

- estruturas\_condicao2

```
if tempoExperiencia < 2:
    print('Nível de conhecimento júnior.')
```

```
elif 2 <= tempoExperiencia < 5:
    print('Nível de conhecimento pleno.')
```

```
else:
    print('Nível de conhecimento sênior.')
```

## Print 1

```
estruturas_condicao2.py > ...
1  tempoExperiencia = 1
2  if tempoExperiencia < 2:
3      print('Nível de conhecimento júnior.')
4  elif 2 <= tempoExperiencia < 5:
5      print('Nível de conhecimento pleno.')
6  else:
7      print('Nível de conhecimento sênior.')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... ^ X

PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/User/Códigos Micro Atividades/estruturas\_condicao2.py"

Nível de conhecimento júnior.

PS C:\Users\User\Códigos Micro Atividades>

## Print 2

```
estruturas_condicao2.py > ...
1  tempoExperiencia = 3
2  if tempoExperiencia < 2:
3      print('Nível de conhecimento júnior.')
4  elif 2 <= tempoExperiencia < 5:
5      print('Nível de conhecimento pleno.')
6  else:
7      print('Nível de conhecimento sênior.')
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + - [ ] [ ] ... ^ x

PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/User/Códigos Micro Atividades/estruturas\_condicao2.py"

Nível de conhecimento pleno.

PS C:\Users\User\Códigos Micro Atividades>

## Print 3

```
estruturas_condicao2.py > ...
1  tempoExperiencia = 5
2  if tempoExperiencia < 2:
3      print('Nível de conhecimento júnior.')
4  elif 2 <= tempoExperiencia < 5:
5      print('Nível de conhecimento pleno.')
6  else:
7      print('Nível de conhecimento sênior.')
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + - [ ] [ ] ... ^ x

PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/User/Códigos Micro Atividades/estruturas\_condicao2.py"

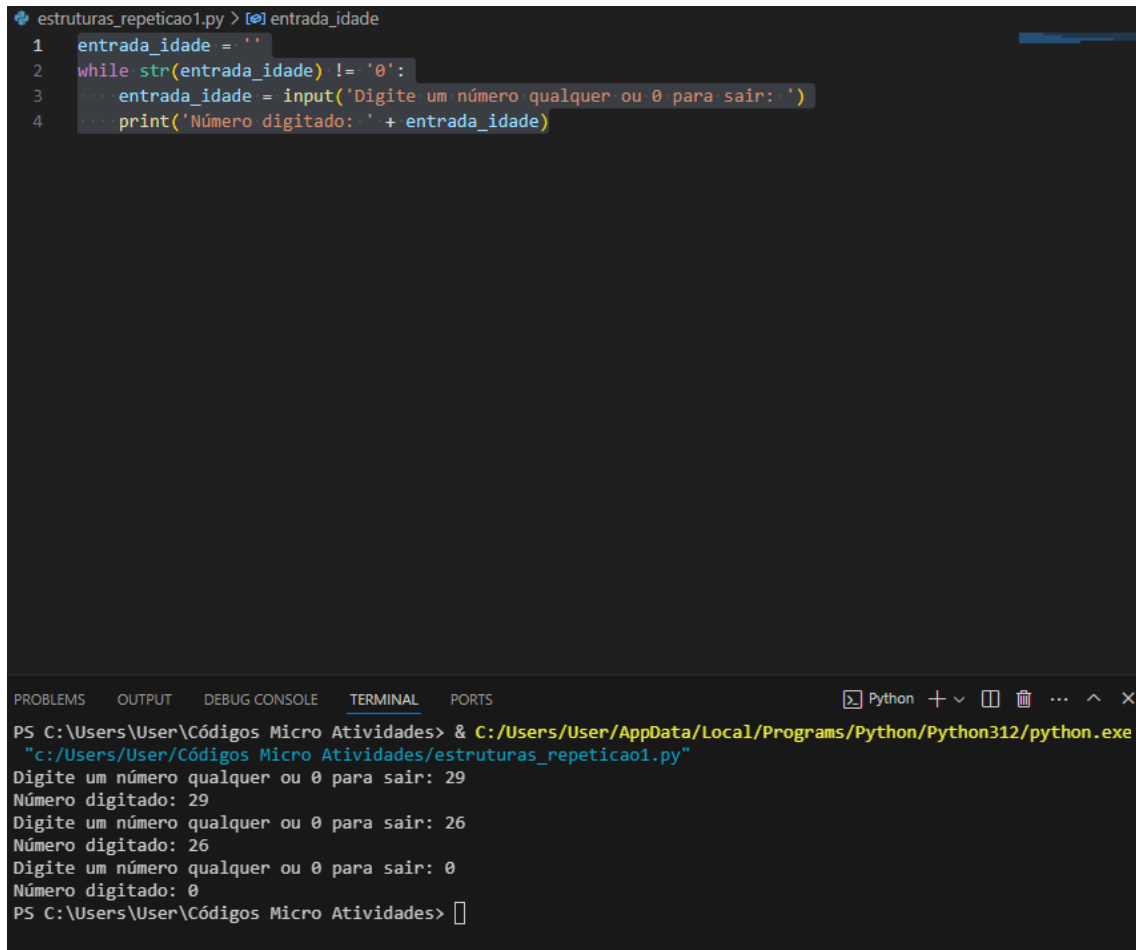
Nível de conhecimento sênior.

PS C:\Users\User\Códigos Micro Atividades>

- estruturas\_repeticao1

```
entrada_idade = ''
while str(entrada_idade) != '0':
    entrada_idade = input('Digite um número qualquer ou 0 para sair:')
    print('Número digitado: ' + entrada_idade)
```

Print 1



The screenshot shows a code editor with the following Python code in `estruturas_repeticao1.py`:

```
1 entrada_idade = ''
2 while str(entrada_idade) != '0':
3     entrada_idade = input('Digite um número qualquer ou 0 para sair: ')
4     print('Número digitado: ' + entrada_idade)
```

Below the code editor is a terminal window showing the execution of the script. The prompt is `PS C:\Users\User\Códigos Micro Atividades>`. The command executed is `& C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/User/Códigos Micro Atividades/estruturas_repeticao1.py"`. The output shows the program running and prompting for input three times:

```
Digite um número qualquer ou 0 para sair: 29
Número digitado: 29
Digite um número qualquer ou 0 para sair: 26
Número digitado: 26
Digite um número qualquer ou 0 para sair: 0
Número digitado: 0
PS C:\Users\User\Códigos Micro Atividades>
```

- estruturas\_repeticao2

```
texto = 'Olá, laço for.'

for item in texto:
    print('Caractere: ' + item)
for numero in range(1, 11):
    print('Número do intervalo: ' + str(numero))
```

## Print 1

```
estruturas_repeticao2.py > texto
1 texto = 'Olá, laço for.'
2
3 for item in texto:
4     print('Caractere: ' + item)
5 for numero in range(1, 11):
6     print('Número do intervalo: ' + str(numero))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... ^ X

```
PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/User/Códigos Micro Atividades/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: .
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 C:\Users\User\Códigos Micro Atividades>
```

- funcoes1

```
def imprimir_variavel():
    texto = 'Olá, funções em Python'
    print(texto)
imprimir_variavel()
```

## Print 1

```
funcoes1.py > ...
1  def imprimir_variavel():
2      texto = 'Olá, funções em Python'
3      print(texto)
4  imprimir_variavel()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

"c:/Users/User/Códigos Micro Atividades/funcoes1.py"  
Olá, funções em Python  
PS C:\Users\User\Códigos Micro Atividades>

- funcoes2

```
def loginUsuario(perfil):
    if perfil.lower() == 'admin':
        print("Bem-vindo, Administrador")
    else:
        print("Bem-vindo, Usuário")

loginUsuario('Admin')      # Bem-vindo, Administrador
loginUsuario('admin')      # Bem-vindo, Administrador
loginUsuario('User')       # Bem-vindo, Usuário
loginUsuario('usuário')    # Bem-vindo, Usuário
loginUsuario('ADMIN')      # Bem-vindo, Administrador
```

## Print 1

```
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  loginUsuario('Admin')      # Bem-vindo, Administrador
8  loginUsuario('admin')     # Bem-vindo, Administrador
9  loginUsuario('User')      # Bem-vindo, Usuário
10 loginUsuario('usuário')    # Bem-vindo, Usuário
11 loginUsuario('ADMIN')     # Bem-vindo, Administrador
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [X] ... ^ X

```
PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/User/Códigos Micro Atividades/funcoes2.py"
Bem-vindo, Administrador
Bem-vindo, Administrador
Bem-vindo, Usuário
Bem-vindo, Usuário
Bem-vindo, Administrador
PS C:\Users\User\Códigos Micro Atividades>
```



# Trabalho Prático

## Calculadora

- calculadora\_v2

```
saida = ''

def adicao(num1, num2):
    return num1 + num2

def subtracao(num1, num2):
    return num1 - num2

def multiplicacao(num1, num2):
    return num1 * num2

def divisao(num1, num2):
    if num2 == 0:
        return "Não foi possível realizar a divisão por 0"
    else:
        return num1 / num2

def calculadora(num1, num2, operacao):
    if operacao in ['+', 'adição']:
        resultado = adicao(num1, num2)
    elif operacao in ['-', 'subtração']:
        resultado = subtracao(num1, num2)
    elif operacao in ['*', 'multiplicação']:
        resultado = multiplicacao(num1, num2)
    elif operacao in ['/', 'divisão']:
        resultado = divisao(num1, num2)
    else:
        return "Operação inválida"

    return resultado

while saida.lower() != 'n':
    primeiro_numero = float(input("Digite o primeiro número: "))
    segundo_numero = float(input("Digite o segundo número: "))
    operacao = input("Digite a operação (+, -, *, / ou o nome da operação): ")

    resultado = calculadora(primeiro_numero, segundo_numero, operacao)

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

    saida = input("Deseja continuar? (S/N): ")
```

Print 1

```

11 calculadora_v2.py > [?] saida
12 def divisao(num1, num2):
13     if num2 == 0:
14         return "Não foi possível realizar a divisão por 0"
15     else:
16         return num1 / num2
17
18 def calculadora(num1, num2, operacao):
19     if operacao in ['+', 'adição']:
20         resultado = adicao(num1, num2)
21     elif operacao in ['-', 'subtração']:
22         resultado = subtracao(num1, num2)
23     elif operacao in ['*', 'multiplicação']:
24         resultado = multiplicacao(num1, num2)
25     elif operacao in ['/', 'divisão']:
26         resultado = divisao(num1, num2)
27     else:
28         return "Operação inválida"
29
30     return resultado
31
32 while saida.lower() != 'n':
33     primeiro_numero = float(input("Digite o primeiro número: "))

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Python

+

⌵

⌵

🗑

⋮

⬆

✖

```

PS C:\Users\User\Códigos Micro Atividades> & C:/Users/User/AppData/Local/Programs/Python/Python312/python.exe
"c:/Users/User/Códigos Micro Atividades/calculadora_v2.py"
Digite o primeiro número: 5
Digite o segundo número: 5
Digite a operação (+, -, *, / ou o nome da operação): /
Resultado da operação: 1.0
Deseja continuar? (S/N): s
Digite o primeiro número: 5
Digite o segundo número: 8
Digite a operação (+, -, *, / ou o nome da operação): +
Resultado da operação: 13.0
Deseja continuar? (S/N): n
PS C:\Users\User\Códigos Micro Atividades>

```