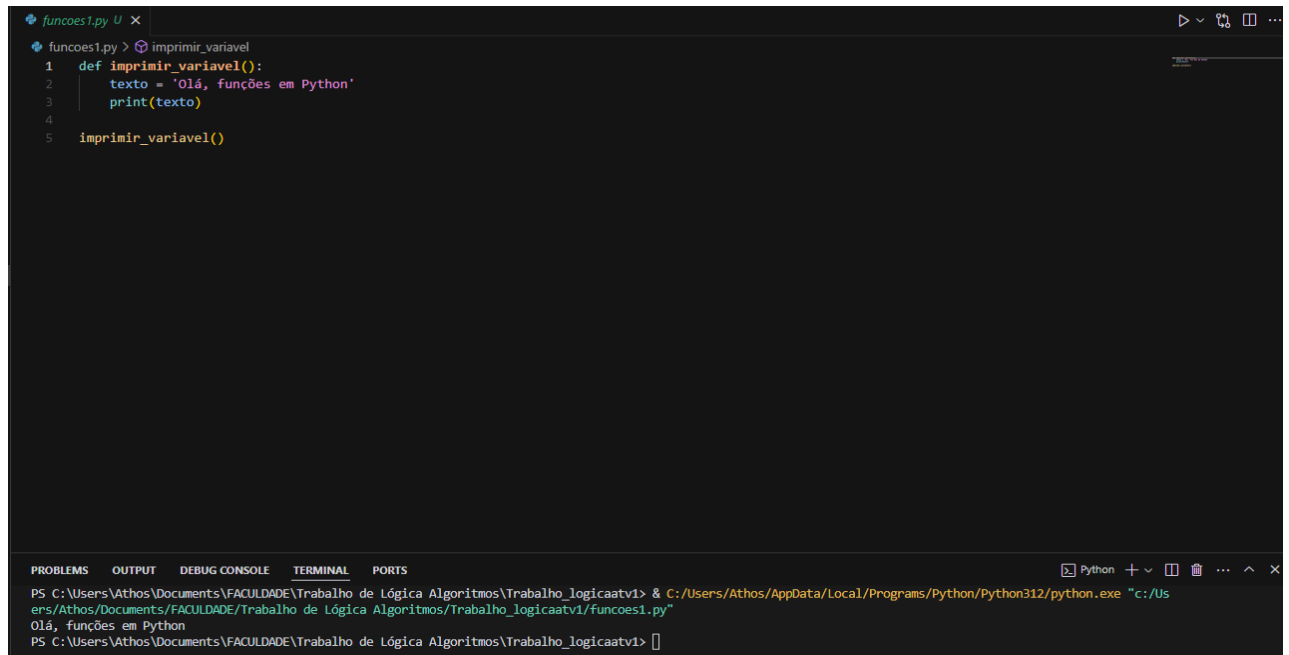


Este documento é parte integrante do curso de Desenvolvimento Full-Stack, oferecido pela faculdade Estácio. O objetivo deste trabalho é apresentar uma série de exemplos de código em Python, abordando diferentes estruturas de controle e funções.

Através desta documentação, será apresentado a prática do conhecimento adquirido durante o curso, essa documentação irá conter o código, lógica e capturas de telas da execução dos scripts.

1. Função para Imprimir Variável

Descrição: Este script define uma função que imprime uma mensagem simples.



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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> & C:/Users/Athos/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Athos/Documents/FACULDADE/Trabalho de Lógica Algoritmos/Trabalho_logicaatv1/funcoes1.py"
Olá, funções em Python
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>
```

```
def imprimir_variavel():
```

```
    texto = "Olá, funções em Python"
```

```
    print(texto)
```

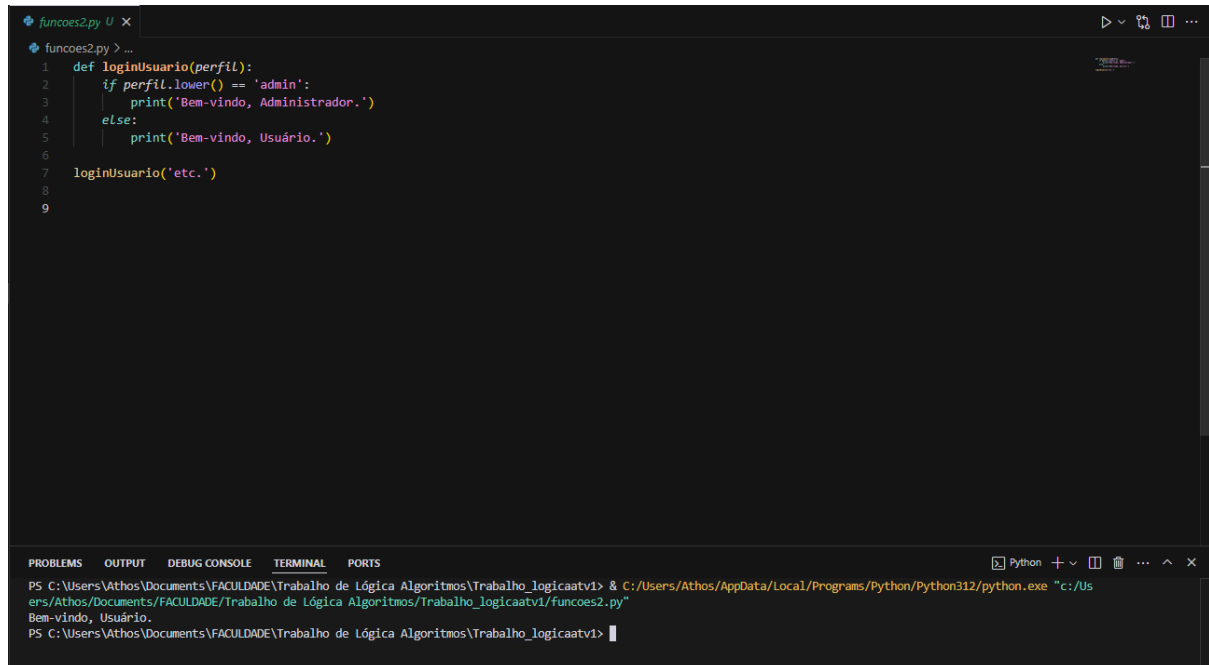
```
imprimir_variavel()
```

Explicação:

1. Definição da Função: A função `imprimir_variavel` é definida sem parâmetros.
2. Atribuição de Valor: A variável `texto` recebe a string "Olá, funções em Python".
3. Impressão: A função `print` é usada para exibir o valor da variável `texto`.
4. Chamada da Função: A função `imprimir_variavel` é chamada para executar o código dentro dela.

2. Função com Parâmetro

Descrição: Este script define uma função que recebe um parâmetro e imprime uma mensagem de saudação com base no valor do parâmetro.



```
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('etc.')
8
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> & C:/Users/Athos/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Athos/Documents/FACULDADE/Trabalho de Lógica Algoritmos/Trabalho_logicaatv1/funcoes2.py"

Bem-vindo, Usuário.

PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>

Código:

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

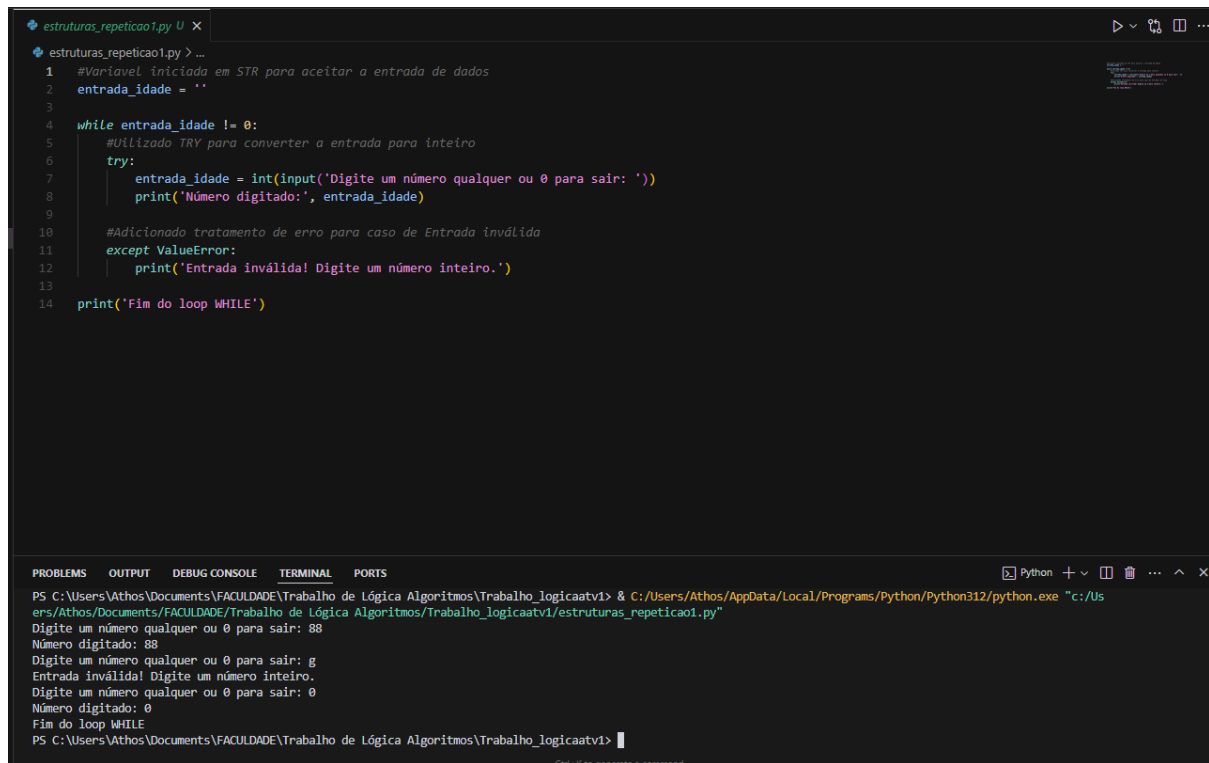
loginUsuario('etc')
```

Explicação:

1. Definição da Função: A função `loginUsuario` é definida com um parâmetro `perfil`.
2. Estrutura Condicional: A estrutura `if-else` verifica se o valor do parâmetro `perfil` é igual a `'admin'`.
3. Impressão: Se a condição for verdadeira, imprime "Bem-vindo, Administrador.". Caso contrário, imprime "Bem-vindo, Usuário.".
4. Chamada da Função: A função `loginUsuario` é chamada com o argumento `'etc'`.

3. Estrutura de Repetição 01

Descrição: Este script solicita ao usuário que digite um número qualquer ou 0 para sair. O programa continua a solicitar números até que o usuário digite 0.



The screenshot shows a Python IDE with a file named `estruturas_repeticao1.py`. The code is as follows:

```
1 #Variável iniciada em STR para aceitar a entrada de dados
2 entrada_idade = ''
3
4 while entrada_idade != 0:
5     #Utilizado TRY para converter a entrada para inteiro
6     try:
7         entrada_idade = int(input('Digite um número qualquer ou 0 para sair: '))
8         print('Número digitado:', entrada_idade)
9
10    #Adicionado tratamento de erro para caso de Entrada inválida
11    except ValueError:
12        print('Entrada inválida! Digite um número inteiro.')
13
14 print('Fim do loop WHILE')
```

The terminal output shows the execution of the script:

```
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> & C:/Users/Athos/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Athos/Documents/FACULDADE/Trabalho de Lógica Algoritmos/Trabalho_logicaatv1/estruturas_repeticao1.py"
Digite um número qualquer ou 0 para sair: 88
Número digitado: 88
Digite um número qualquer ou 0 para sair: g
Entrada inválida! Digite um número inteiro.
Digite um número qualquer ou 0 para sair: 0
Número digitado: 0
Fim do loop WHILE
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>
```

Código:

```
# Variável iniciada em STR para aceitar a entrada de dados
entrada_idade = ''

while entrada_idade != '0':

    # Utilizado TRY para converter a entrada para inteiro
    try:

        entrada_idade = int(input("Digite um número qualquer ou 0 para sair:
"))

        print("Número digitado: ", entrada_idade)

    # Dicionado tratamento de erro para caso de Entrada inválida
    except ValueError:

        print("Entrada inválida! Digite um número inteiro.")

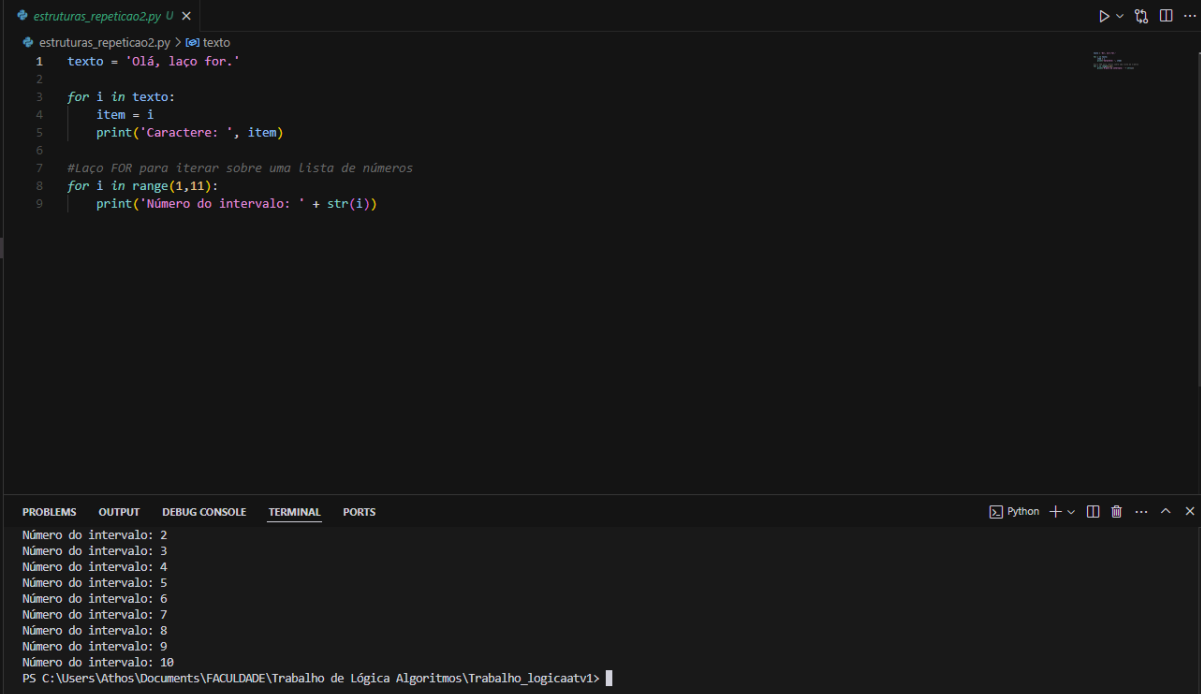
print("Fim do loop WHILE")
```

Explicação:

1. Inicialização da Variável: A variável `entrada_idade` é inicializada como uma string vazia.
2. Laço WHILE: O laço `while` continua a executar enquanto `entrada_idade` for diferente de '0'.
3. Entrada de Dados: Dentro do laço, o usuário é solicitado a digitar um número.
4. Conversão e Tratamento de Erro: A entrada do usuário é convertida para inteiro. Se a conversão falhar, uma mensagem de erro é exibida.
5. Impressão do Número: Se a conversão for bem-sucedida, o número digitado é impresso.
6. Fim do Laço: Quando o usuário digita 0, o laço termina e uma mensagem de fim é impressa.

4. Estrutura de Repetição 02

Descrição: Este script utiliza um laço FOR para iterar sobre uma string e um intervalo de números.



```
estruturas_repeticao2.py > texto
1 texto = 'Olá, faça for.'
2
3 for i in texto:
4     item = i
5     print('Caractere: ', item)
6
7 #Laço FOR para iterar sobre uma lista de números
8 for i in range(1,11):
9     print('Número do intervalo: ' + str(i))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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
P5 C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>

Código:

```
texto = "Olá, faça for."

for i in texto:

    item = i

    print("Caractere: ", item)

# Loop FOR para iterar sobre uma lista de números
for i in range(1, 11):

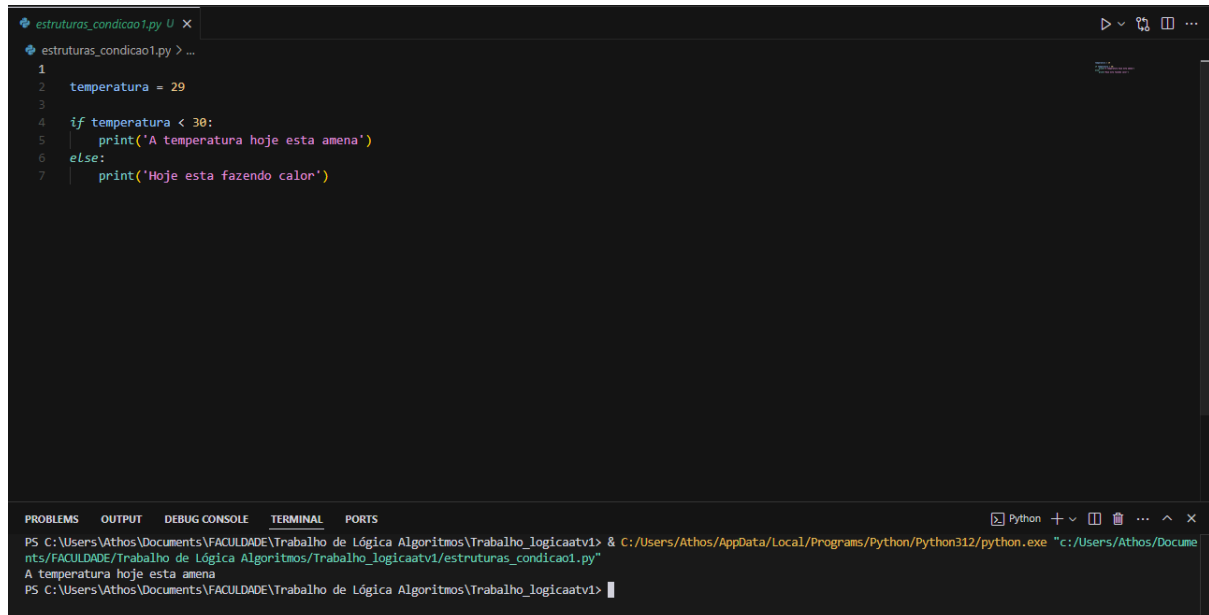
    print("Número do intervalo: " + str(i))
```

Explicação:

1. Atribuição de Valor: A variável `texto` recebe a string "Olá, faça for."
2. Laço FOR: O laço `for` itera sobre cada caractere da string `texto`.
3. Impressão: Para cada caractere, a variável `item` recebe o valor do caractere atual e é impresso.
4. Laço FOR para Intervalo: O segundo laço `for` itera sobre os números de 1 a 10.
5. Impressão de Números: Para cada número no intervalo, ele é impresso.

5. Estrutura Condicional 01

Descrição: Este script verifica a temperatura e imprime uma mensagem indicando se está amena ou fazendo calor.



```
estruturas_condicao1.py U x
estruturas_condicao1.py > ...
1
2     temperatura = 29
3
4     if temperatura < 30:
5         print('A temperatura hoje esta amena')
6     else:
7         print('Hoje esta fazendo calor')
```

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

```
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> & C:/Users/Athos/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Athos/Docume
nts/FACULDADE/Trabalho de Lógica Algoritmos/Trabalho_logicaatv1/estruturas_condicao1.py"
A temperatura hoje esta amena
PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> |
```

Código:

```
temperatura = 29

if temperatura < 30:

    print("A temperatura hoje está amena")

else:

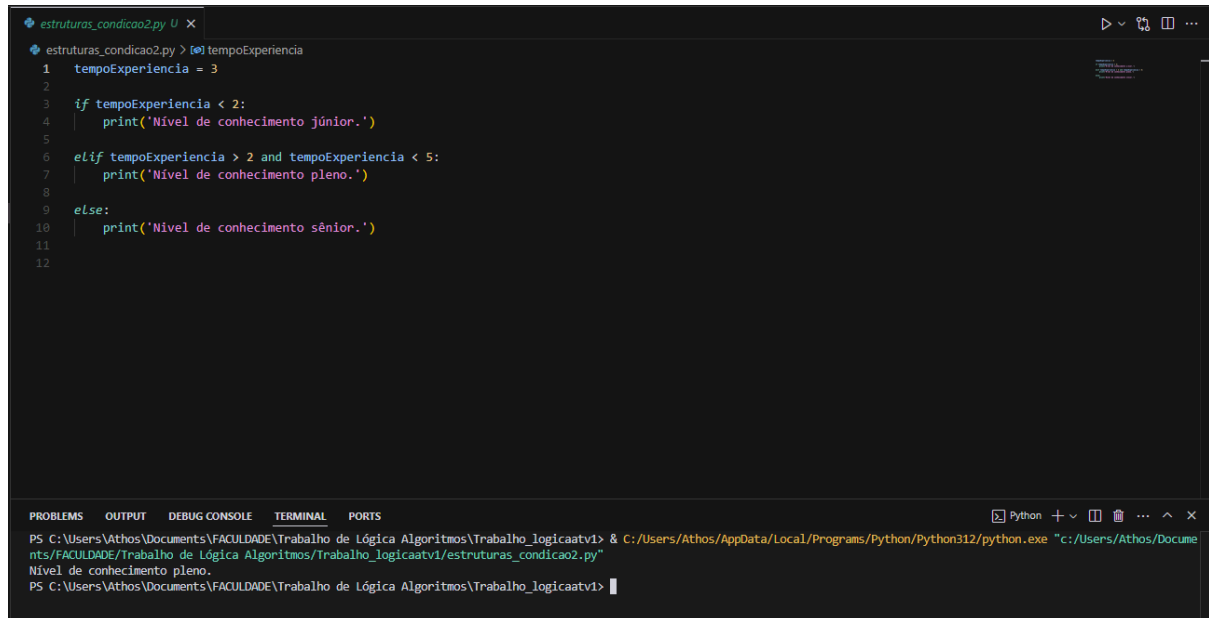
    print("Hoje está fazendo calor")
```

Explicação:

1. Atribuição de Valor: A variável `temperatura` recebe o valor 29.
2. Estrutura Condicional: A estrutura `if-else` verifica se a temperatura é menor que 30.
3. Impressão: Se a condição for verdadeira, imprime "A temperatura hoje está amena". Caso contrário, imprime "Hoje está fazendo calor".

6. Estrutura Condicional 02

Descrição: Este script verifica o tempo de experiência de um funcionário e imprime o nível de conhecimento correspondente.



```
estruturas_condicao2.py X
estruturas_condicao2.py > tempoExperiencia
1 tempoExperiencia = 3
2
3 if tempoExperiencia < 2:
4     print('Nível de conhecimento júnior.')
5
6 elif tempoExperiencia > 2 and tempoExperiencia < 5:
7     print('Nível de conhecimento pleno.')
8
9 else:
10    print('Nível de conhecimento sênior.')
11
12
```

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

PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1> & C:/Users/Athos/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Athos/Documents/FACULDADE/Trabalho de Lógica Algoritmos/Trabalho_logicaatv1/estruturas_condicao2.py"

Nível de conhecimento pleno.

PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>

Código:

```
tempoExperiencia = 3

if tempoExperiencia < 2:

    print("Nível de conhecimento júnior.")

elif tempoExperiencia >= 2 and tempoExperiencia < 5:

    print("Nível de conhecimento pleno.")

else:

    print("Nível de conhecimento sênior.")
```

Explicação:

1. Atribuição de Valor: A variável `tempoExperiencia` recebe o valor 3.
2. Estrutura Condicional Composta: A estrutura `if-elif-else` verifica o valor de `tempoExperiencia`.
3. Impressão: Dependendo do valor de `tempoExperiencia`, uma mensagem correspondente é impressa.

7. Calculadora Simples

Descrição: Este script implementa uma calculadora simples que realiza operações de adição, subtração, multiplicação e divisão. O usuário pode inserir dois números e a operação desejada.

Código:

```
# Variável de controle para saída
saida = ''

# Função para soma
def adicao(a, b):
    return a + b

# Função para subtração
def subtracao(a, b):
    return a - b

# Função para multiplicação
def multiplicacao(a, b):
    return a * b

# Função para divisão
def divisao(a, b):
    if b == 0:
        return "Não foi possível realizar a divisão por 0"
    return a / b

# Função calculadora
def calculadora(num1, num2, operacao):
    if operacao in ('+', 'soma'):
        resultado = adicao(num1, num2)
```

```

elif operacao in ('-', 'subtracao'):
    resultado = subtracao(num1, num2)
elif operacao in ('*', 'multiplicacao'):
    resultado = multiplicacao(num1, num2)
elif operacao in ('/', 'divisao'):
    resultado = divisao(num1, num2)
else:
    resultado = "Operação inválida"
return resultado

# Laço principal
while saida.lower() != 'n':
    try:
        # Solicita os números e a operação ao usuário
        num1 = float(input("Digite o primeiro número: "))
        num2 = float(input("Digite o segundo número: "))
        operacao = input("Digite a operação desejada (+, -, *, / ou o nome da
operação): ").strip().lower()

        # Calcula o resultado
        resultado = calculadora(num1, num2, operacao)

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

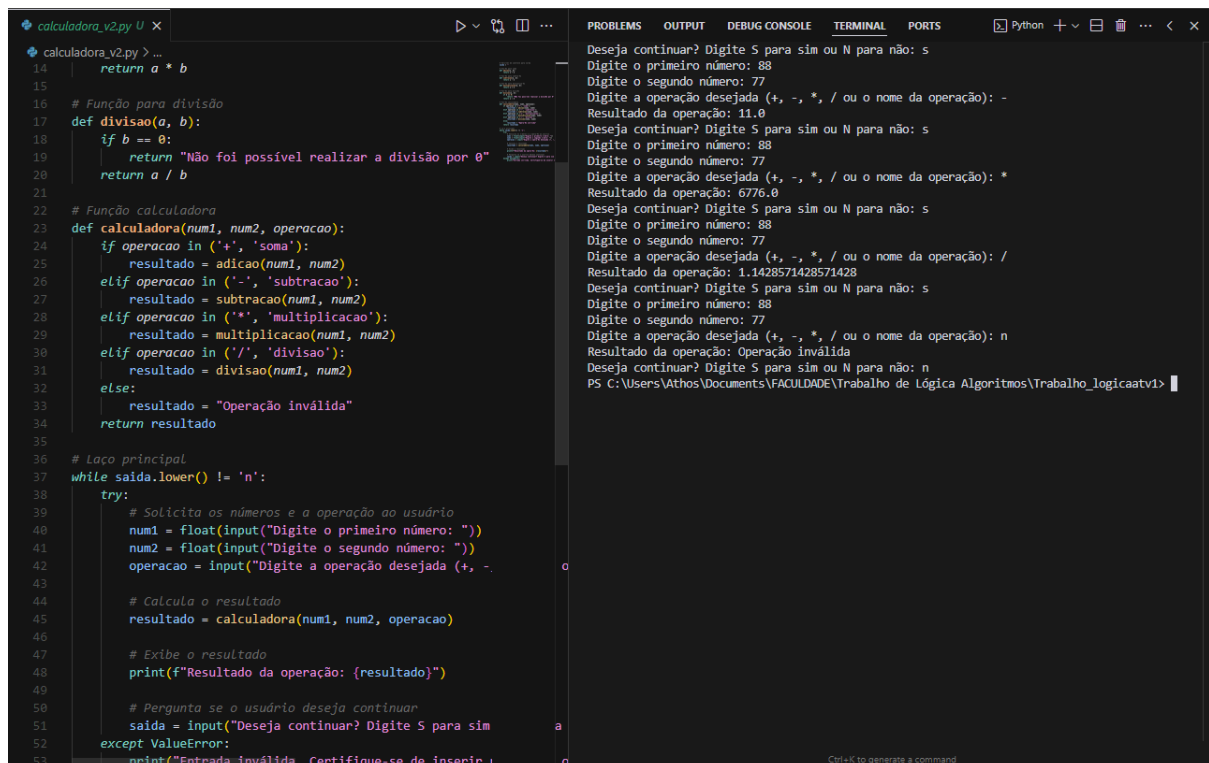
        # Pergunta se o usuário deseja continuar
        saida = input("Deseja continuar? Digite S para sim ou N para não:
").strip().lower()
    except ValueError:
        print("Entrada inválida. Certifique-se de inserir números e operações
válidas.")

```

Explicação:

1. Inicialização da Variável de Controle: A variável `saida` é inicializada como uma string vazia.
2. Definição das Funções de Operações: As funções `adicao`, `subtracao`, `multiplicacao` e `divisao` são definidas para realizar as operações matemáticas.
3. Função Calculadora: A função `calculadora` recebe dois números e uma operação, determina a operação desejada e chama a função correspondente.
4. Laço Principal: O laço `while` continua a executar enquanto `saida` for diferente de 'n'.
5. Entrada de Dados: Dentro do laço, o usuário é solicitado a digitar dois números e a operação desejada.
6. Cálculo e Impressão do Resultado: A função `calculadora` é chamada com os valores inseridos e o resultado é impresso.
7. Continuação do Programa: O usuário é perguntado se deseja continuar. Se a resposta for 'n', o laço termina.

Execução:



```
calculadora_v2.py X
14     return a * b
15
16 # Função para divisão
17 def divisao(a, b):
18     if b == 0:
19         return "Não foi possível realizar a divisão por 0"
20     return a / b
21
22 # Função calculadora
23 def calculadora(num1, num2, operacao):
24     if operacao in ('+', 'soma'):
25         resultado = adicao(num1, num2)
26     elif operacao in ('-', 'subtracao'):
27         resultado = subtracao(num1, num2)
28     elif operacao in ('*', 'multiplicacao'):
29         resultado = multiplicacao(num1, num2)
30     elif operacao in ('/', 'divisao'):
31         resultado = divisao(num1, num2)
32     else:
33         resultado = "Operação inválida"
34     return resultado
35
36 # Laço principal
37 while saida.lower() != 'n':
38     try:
39         # Solicita os números e a operação ao usuário
40         num1 = float(input("Digite o primeiro número: "))
41         num2 = float(input("Digite o segundo número: "))
42         operacao = input("Digite a operação desejada (+, -, *, /): ")
43
44         # Calcula o resultado
45         resultado = calculadora(num1, num2, operacao)
46
47         # Exibe o resultado
48         print(f"Resultado da operação: {resultado}")
49
50         # Pergunta se o usuário deseja continuar
51         saida = input("Deseja continuar? Digite S para sim ou N para não: ")
52     except ValueError:
53         print("Entrada inválida. Certifique-se de inserir um valor válido.")
54
55 # Problemas de execução:
56 Deseja continuar? Digite S para sim ou N para não: s
57 Digite o primeiro número: 88
58 Digite o segundo número: 77
59 Digite a operação desejada (+, -, *, / ou o nome da operação): -
60 Resultado da operação: 11.0
61 Deseja continuar? Digite S para sim ou N para não: s
62 Digite o primeiro número: 88
63 Digite o segundo número: 77
64 Digite a operação desejada (+, -, *, / ou o nome da operação): *
65 Resultado da operação: 6776.0
66 Deseja continuar? Digite S para sim ou N para não: s
67 Digite o primeiro número: 88
68 Digite o segundo número: 77
69 Digite a operação desejada (+, -, *, / ou o nome da operação): /
70 Resultado da operação: 1.1428571428571428
71 Deseja continuar? Digite S para sim ou N para não: s
72 Digite o primeiro número: 88
73 Digite o segundo número: 77
74 Digite a operação desejada (+, -, *, / ou o nome da operação): n
75 Resultado da operação: Operação inválida
76 Deseja continuar? Digite S para sim ou N para não: n
77 PS C:\Users\Athos\Documents\FACULDADE\Trabalho de Lógica Algoritmos\Trabalho_logicaatv1>
```

Esta documentação apresentou uma série de exemplos de código em Python que demonstram diferentes estruturas de controle e funções. Cada exemplo é acompanhado de uma descrição detalhada, do código correspondente, comentários sobre funcionalidades e estrutura, e imagens que ilustram a execução dos scripts.

Athos de Araujo Gomes de França