

# Trabalho Prático - DGT2817

**Disciplina: Lógica, Algoritmos e Programação de Computadores**

**Aluno: Bruno Chagas Da Silva**

## Introdução

Este documento apresenta a documentação do Trabalho Prático da disciplina DGTE317 - Lógica, Algoritmos e Programação de Computadores. O objetivo é demonstrar o aprendizado em estruturas condicionais (if, else, elif), estruturas de repetição (for, while), funções e seus argumentos em Python, culminando na refatoração de uma calculadora que utiliza esses conceitos. As microatividades preparatórias e o trabalho prático final foram desenvolvidos utilizando a IDE VS Code e o interpretador Python, conforme orientações fornecidas.

## Microatividade 1 – If e Else

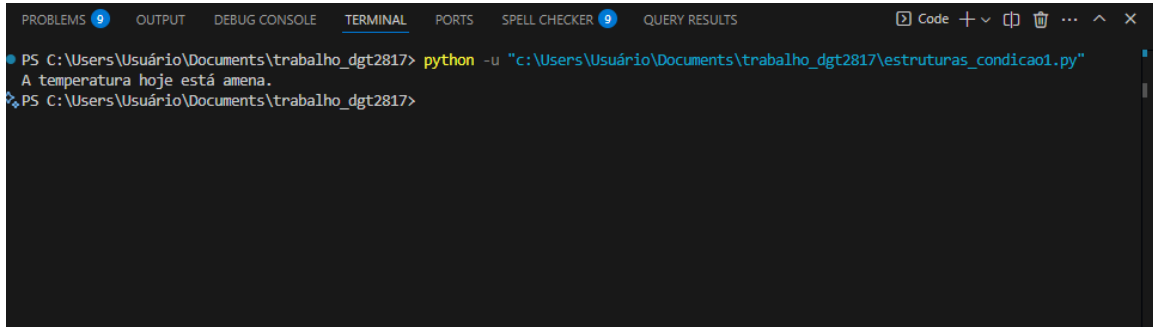
### Objetivo

Compreender e aplicar as estruturas condicionais if e else em Python, criando um programa que verifica a temperatura e exibe uma mensagem apropriada.

```
temperatura = 29

if temperatura < 30:
    print("A temperatura hoje está amena.")
else:
    print("Hoje está fazendo calor.")
```

## Resultados Esperados



The screenshot shows a VS Code terminal window with the following content:

```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 9 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\estruturas_condicao1.py"
A temperatura hoje está amena.
PS C:\Users\Usuário\Documents\trabalho_dgt2817>
```

## Microatividade 2 – Elif.

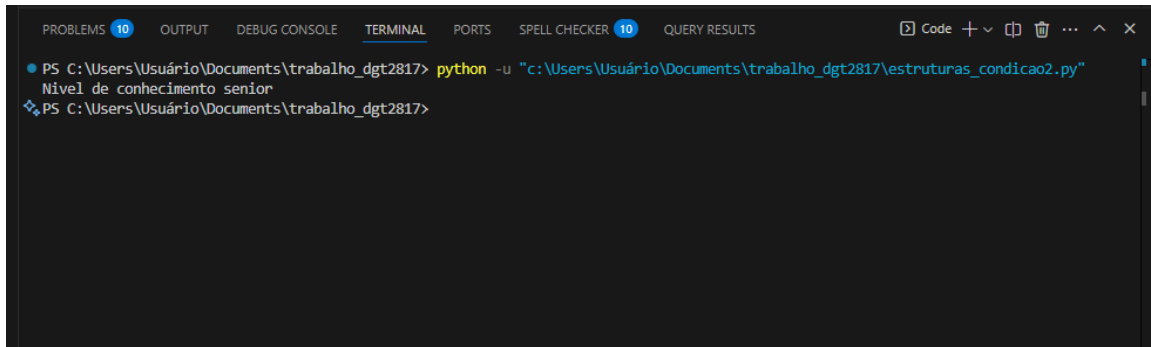
### Objetivo

Aplicar a estrutura elif para classificar o nível de conhecimento com base no tempo de experiência.

```
tempoExperiencia = 5

if tempoExperiencia < 2:
    print("Nível de conhecimento junior")
elif tempoExperiencia < 5:
    print("Nível de conhecimento pleno")
else:
    print("Nível de conhecimento senior")
```

## Resultados Esperados



```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 10 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\estruturas_condicao2.py"
Nível de conhecimento senior
PS C:\Users\Usuário\Documents\trabalho_dgt2817>
```

## Microatividade 3 – Estrutura While

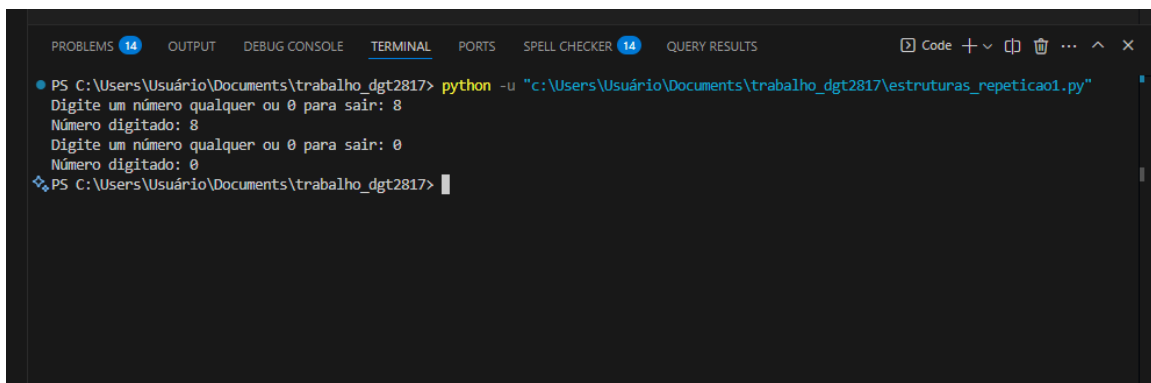
### Objetivo

Utilizar a estrutura while para criar um programa que lê números até que o usuário digite 0.

```
entrada_idade = ''

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

### Resultados Esperados



```
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 14 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\estruturas_repeticao1.py"
Digite um número qualquer ou 0 para sair: 8
Número digitado: 8
Digite um número qualquer ou 0 para sair: 0
Número digitado: 0
PS C:\Users\Usuário\Documents\trabalho_dgt2817>
```

## Microatividade 4 – Estrutura For

### Objetivo

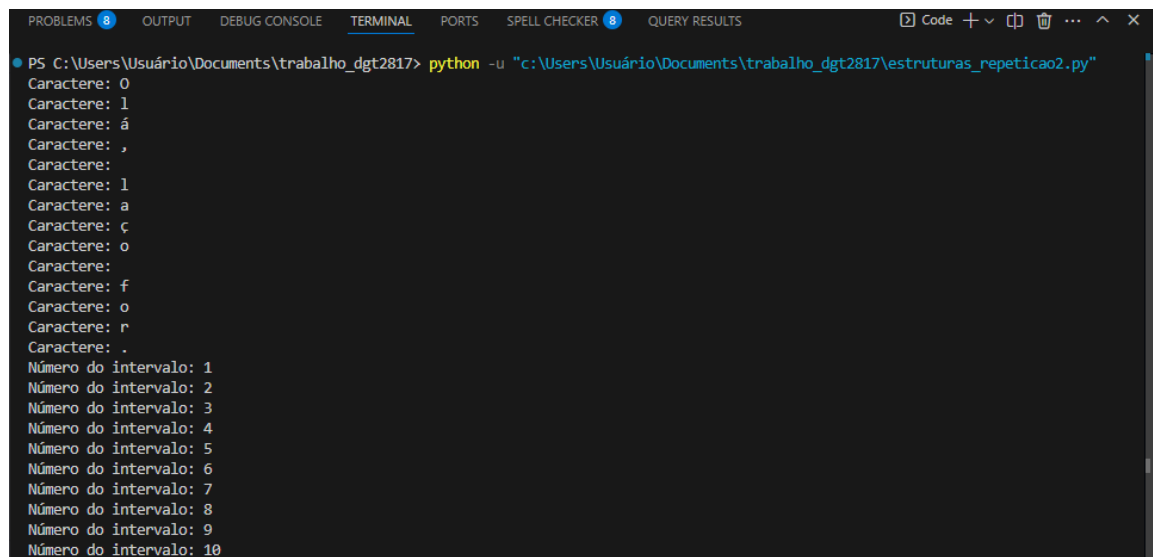
Aplicar a estrutura for para iterar sobre uma string e um intervalo numérico.

```
texto = 'Olá, laço for.'

for item in texto:
    print('Caractere: ' + item)

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

### Resultados Esperados



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 8 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\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
```

## Microatividade 5 – Funções sem Parâmetros

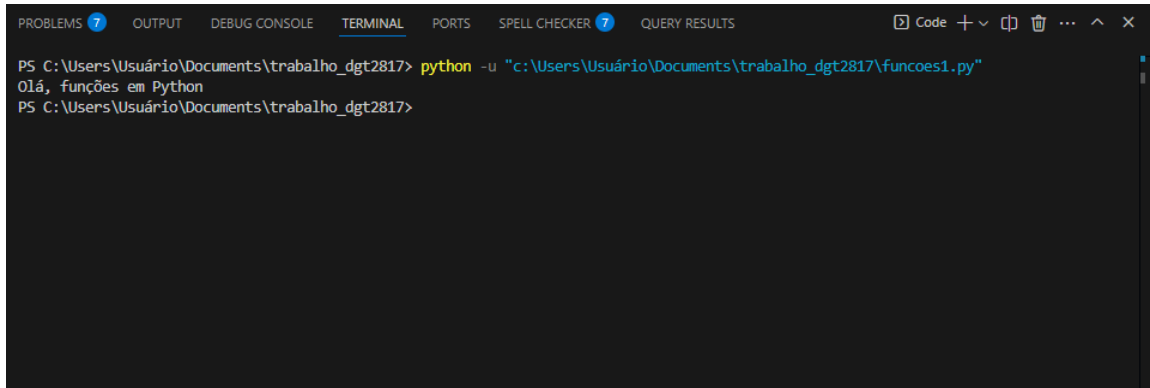
### Objetivo

Criar e chamar uma função sem parâmetros que imprime uma mensagem fixa.

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

imprimir_variavel()
```

## Resultados Esperados



```
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 7 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\funcoes1.py"
Olá, funções em Python
PS C:\Users\Usuário\Documents\trabalho_dgt2817>
```

## Microatividade 6 – Funções com Parâmetros

### Objetivo

Criar uma função com parâmetro que verifica se o perfil é "admin" e exibe uma mensagem apropriada.

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

loginUsuario('Admin')
loginUsuario('admin')
loginUsuario('User')
loginUsuario('usuário')
```

## Resultados Esperados

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 12 QUERY RESULTS
PS C:\Users\Usuário\Documents\trabalho_dgt2817> python -u "c:\Users\Usuário\Documents\trabalho_dgt2817\funcoes2.py"
Bem-vindo, Administrador
Bem-vindo, Usuário
Bem-vindo, Usuário
PS C:\Users\Usuário\Documents\trabalho_dgt2817>
```

## Trabalho Prático Final – Calculadora Refatorada

### Objetivo

Refatorar uma calculadora utilizando funções, estruturas condicionais e um laço while para interação contínua com o usuário.

```
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:
        return 'Não foi possível realizar a divisão por 0'
    return a / b

def calculadora(num1, num2, operacao):
    if operacao == '+' or operacao.lower() == 'soma':
        resultado = adicao(num1, num2)
    elif operacao == '-' or operacao.lower() == 'subtracao':
        resultado = subtracao(num1, num2)
    elif operacao == '*' or operacao.lower() == 'multiplicacao':
        resultado = multiplicacao(num1, num2)
    elif operacao == '/' or operacao.lower() == 'divisao':
```

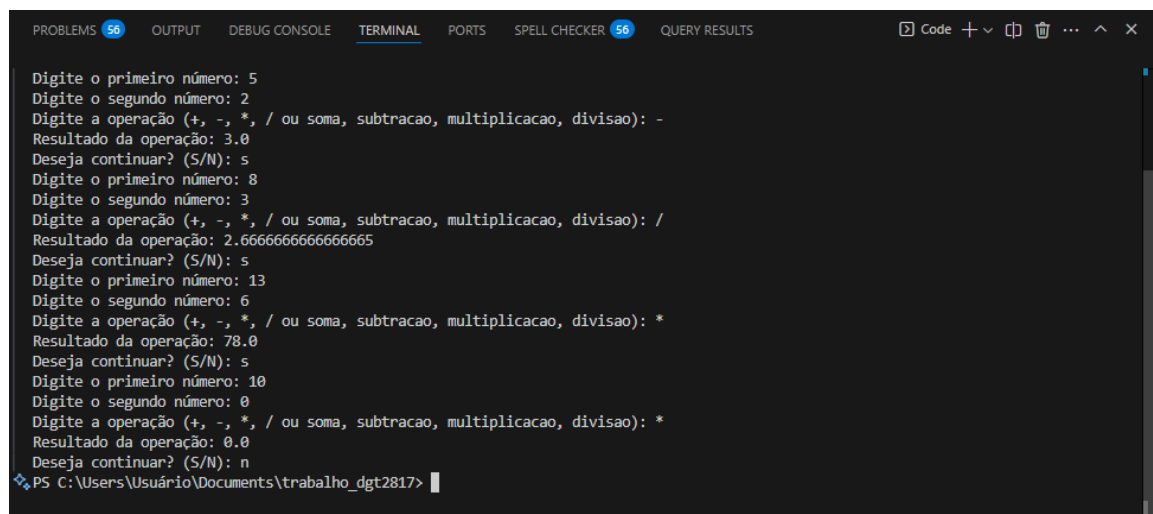
```

        resultado = divisao(num1, num2)
    else:
        resultado = 'Operação inválida'
    return resultado

saida = ''
while saida.lower() != 'n':
    try:
        num1 = float(input('Digite o primeiro número: '))
        num2 = float(input('Digite o segundo número: '))
        operacao = input('Digite a operação (+, -, *, / ou soma,
subtracao, multiplicacao, divisao): ')
        resultado = calculadora(num1, num2, operacao)
        print('Resultado da operação: ' + str(resultado))
    except ValueError:
        print('Entrada inválida. Por favor, digite números válidos.')
    saida = input('Deseja continuar? (S/N): ')

```

## Resultados Esperados



```

PROBLEMS 56 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 56 QUERY RESULTS
Digite o primeiro número: 5
Digite o segundo número: 2
Digite a operação (+, -, *, / ou soma, subtracao, multiplicacao, divisao): -
Resultado da operação: 3.0
Deseja continuar? (S/N): s
Digite o primeiro número: 8
Digite o segundo número: 3
Digite a operação (+, -, *, / ou soma, subtracao, multiplicacao, divisao): /
Resultado da operação: 2.6666666666666665
Deseja continuar? (S/N): s
Digite o primeiro número: 13
Digite o segundo número: 6
Digite a operação (+, -, *, / ou soma, subtracao, multiplicacao, divisao): *
Resultado da operação: 78.0
Deseja continuar? (S/N): s
Digite o primeiro número: 10
Digite o segundo número: 0
Digite a operação (+, -, *, / ou soma, subtracao, multiplicacao, divisao): *
Resultado da operação: 0.0
Deseja continuar? (S/N): n
PS C:\Users\Usuário\Documents\trabalho_dgt2817>

```

## Link do Repositório GitHub

[https://github.com/Dev-Shihin/trabalho\\_dgt2817](https://github.com/Dev-Shihin/trabalho_dgt2817)