

# LOGICA, ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

ALUNO

Luccas E. Costa

TUTOR

Alessandro Calin

CURSO

Desenvolvimento Full Stack

## TAREFAS A FAZER

DESCREVER A UTILIZAÇÃO DAS ESTRUTURAS DE CONDIÇÃO IF E ELSE EM PYTHON;  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE CONDIÇÃO ELSE IF (ELIF) EM PYTHON;  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO WHILE EM PYTHON;  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO FOR EM PYTHON;  
DESCREVER A UTILIZAÇÃO DE FUNÇÕES EM PYTHON;  
DESCREVER A UTILIZAÇÃO ARGUMENTOS DE FUNÇÕES NO PYTHON;  
REFAZER A CALCULADORA UTILIZANDO ESTRUTURA CONDICIONAL E FUNÇÕES;

## TAREFAS CONCLUÍDAS

DESCREVER A UTILIZAÇÃO DAS ESTRUTURAS DE CONDIÇÃO IF E ELSE EM PYTHON; ✓  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE CONDIÇÃO ELSE IF (ELIF) EM PYTHON; ✓  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO WHILE EM PYTHON; ✓  
DESCREVER A UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO FOR EM PYTHON; ✓  
DESCREVER A UTILIZAÇÃO DE FUNÇÕES EM PYTHON; ✓  
DESCREVER A UTILIZAÇÃO ARGUMENTOS DE FUNÇÕES NO PYTHON; ✓  
REFAZER A CALCULADORA UTILIZANDO ESTRUTURA CONDICIONAL E FUNÇÕES; ✓

## UTILIZAÇÃO DAS ESTRUTURAS DE CONDIÇÃO IF E ELSE EM PYTHON

### IF

```
#if: "se" , se a condição for verdadeira, ele executa o que está dentro do bloco.  
temperatura = 29  
if temperatura < 30:  
    print ('A temperatura hoje está amena')  
else:  
    print ('Hoje está fazendo calor')
```

A temperatura hoje está amena

### ELSE

```
#else: "senão" , se a condição não for verdadeira, faz outra coisa.  
temperatura = 31  
if temperatura < 30:  
    print ('A temperatura hoje está amena')  
else:  
    print ('Hoje está fazendo calor')
```

Hoje está fazendo calor

## UTILIZAÇÃO DA ESTRUTURA DE CONDIÇÃO ELSE IF (ELIF) EM PYTHON

```
#Elif: "senao se" ,E usado quando temos mais de 2 possibilidades ,  
#assim ele testa o "if" , se o "if" for falso, ele testa o proximo "elif".  
#se nenhum if ou elif for verdadeiro, ele cai no else.
```

### IF

```
tempoExperiencia = 1  
if tempoExperiencia < 2:  
    print('Nível de conhecimento júnior.')  
elif tempoExperiencia <= 5:  
    print('Nível de conhecimento pleno.')  
else:  
    print('Nível de conhecimento sênior.')
```

Nível de conhecimento júnior.

### ELIF

```
tempoExperiencia = 3  
if tempoExperiencia < 2:  
    print('Nível de conhecimento júnior.')  
elif tempoExperiencia <= 5:  
    print('Nível de conhecimento pleno.')  
else:  
    print('Nível de conhecimento sênior.')
```

Nível de conhecimento pleno.

### ELSE

```
tempoExperiencia = 6  
if tempoExperiencia < 2:  
    print('Nível de conhecimento júnior.')  
elif tempoExperiencia <= 5:  
    print('Nível de conhecimento pleno.')  
else:  
    print('Nível de conhecimento sênior.')
```

Nível de conhecimento sênior.

## UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO WHILE EM PYTHON

```
#O while é usado quando precisa repetir alguma coisa enquanto uma condição for verdadeira  
#nesse exemplo a variável "entrada_idade" é uma str e está vazia  
#o while vai continuar sendo executado enquanto o valor digitado for diferente de "0"
```

### WHILE

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

```
Digite um número qualquer ou 0 para sair: 1  
Número digitado: 1  
Digite um número qualquer ou 0 para sair: 2  
Número digitado: 2  
Digite um número qualquer ou 0 para sair: 0  
Número digitado: 0
```

## UTILIZAÇÃO DA ESTRUTURA DE REPETIÇÃO FOR EM PYTHON

```
#for é usado quando eu preciso repetir algo , percorrer uma sequencia de coisas ,  
#como um itens de uma lista , numeros dentro de um intervalo. o for passa por cada item.
```

### FOR

```
#no primeiro exemplo o for vai passando por cada letra da frase  
#que esta em "texto" uma de cada vez e vai imprimindo o caractere  
texto = 'Olá, laço for.'  
for item in texto:  
    print('Caractere: ' + item)
```

Caractere: 0	Caractere: 1	Caractere:
Caractere: l	Caractere: a	Caractere: f
Caractere: á	Caractere: ç	Caractere: o
Caractere: ,	Caractere: o	Caractere: r
Caractere:	Caractere:	Caractere: .

### FOR

```
#no segundo exemplo o for esta rodando de 1 ate 10 (range1,11),  
#ele vai contando numero por numero e mostrando o resultado  
for numero in range(1, 11):  
    print('Número do intervalo: ' + str(numero))
```

intervalo: 1
intervalo: 2
intervalo: 3
intervalo: 4
intervalo: 5
intervalo: 6
intervalo: 7
intervalo: 8
intervalo: 9
intervalo: 10

## UTILIZAÇÃO DE FUNÇÕES EM PYTHON

```
#funções são blocos de código que podemos criar e usar sempre que quiser.  
#como se fosse uma caixa , atribuimos um nome e uma ação  
#depois podemos chamar essa caixa para que ela execute o que tem dentro  
#nesse exemplo o "def" cria a função chamada "imprimir_variavel"  
#dentro da função nos colocamos o que ela vai fazer , e neste caso ela cria  
#a variável "texto" e imprime a frase "Olá, funções em Python"  
#quando for chamada a função "imprimir_variavel()"
```

### FUNÇÕES

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

```
imprimir_variavel()
```

Olá, funções em Python

## UTILIZAÇÃO ARGUMENTOS DE FUNÇÕES NO PYTHON

```
#Quando criamos a função, as vezes precisa passar algumas informações para ela trabalhar,  
#que no caso são os argumentos ou parâmetros quando estamos definindo uma função  
#nesse exemplo criamos a função "loginUsuario" que recebe o argumento de "perfil".  
#quando chamamos a função e passamos um valor como em "loginUsuario('Admin')"  
#esse valor é usado dentro da função para fazer algo, no caso do exemplo ele verifica  
#se é um administrador ou não  
#a função "perfil.lower()" é para garantir que não importa se a pessoa escreva "admin", "ADMIN", "aDmIn" ,  
#tudo fica em minúsculo antes de comparar  
#argumentos são as informações que você passa para a função funcionar de formas diferentes dependendo  
#de cada situação, para que a função fique mais flexível
```

### ARGUMENTOS DE FUNÇÕES

```
def loginUsuario(perfil):  
    if perfil.lower() == 'admin':  
        print('Bem-vindo, Administrador')  
    else:  
        print('Bem-vindo, Usuário')  
#chamada da mesma passando diferentes valores como parâmetro  
loginUsuario('Admin')  
loginUsuario('admin')  
loginUsuario('User')  
loginUsuario('usuário')
```

```
Bem-vindo, Administrador  
Bem-vindo, Administrador  
Bem-vindo, Usuário  
Bem-vindo, Usuário
```

## CALCULADORA UTILIZANDO ESTRUTURA CONDICIONAL E FUNÇÕES

### ESTRUTURA DA CALCULADORA V2 USANDO O QUE FOI APRENDIDO NAS MICRO-ATIVIDADES

#### CALCULADORA V2

```
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() == 'adicao':
        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

saída = ''
while saída.lower() != 'n':
    try:
        num1 = float(input('Digite o primeiro número: '))
        num2 = float(input('Digite o segundo número: '))
        operacao = input('Digite a operação (+, -, *, /) ou o nome da operação (adicao, subtracao, multiplicacao, divisao): ').strip()

        resultado = calculadora(num1, num2, operacao)
        print('Resultado da operação:', resultado)

    except ValueError:
        print("Por favor, insira números válidos.")
    saída = input('Deseja continuar? (S/N): ').strip()
print('Programa encerrado.')
```

## CALCULADORA UTILIZANDO ESTRUTURA CONDICIONAL E FUNÇÕES

### CALCULADORA V2 FUNCIONANDO

```
Digite o primeiro numero: 10
Digite o segundo numero: 20
Digite a operação (+, -, *, /) ou o nome da operação (adicao, subtracao, multiplicacao, divisao): +
Resultado da operação: 30.0
Deseja continuar? (S/N): s
Digite o primeiro numero: 10
Digite o segundo numero: 20
Digite a operação (+, -, *, /) ou o nome da operação (adicao, subtracao, multiplicacao, divisao): -
Resultado da operação: -10.0
Deseja continuar? (S/N): s
Digite o primeiro numero: 10
Digite o segundo numero: 20
Digite a operação (+, -, *, /) ou o nome da operação (adicao, subtracao, multiplicacao, divisao): *
Resultado da operação: 200.0
Deseja continuar? (S/N): s
Digite o primeiro numero: 10
Digite o segundo numero: 20
Digite a operação (+, -, *, /) ou o nome da operação (adicao, subtracao, multiplicacao, divisao): /
Resultado da operação: 0.5
Deseja continuar? (S/N): n
Programa encerrado.
```