

Trabalho Prático

Disciplina: DGT2817 Lógica, Algoritmos e Programação de Computadores

Aluno: Robson do Nascimento Ferreira

Matricula: 2025 0726 8899

Índice

- 1. Definição do Projeto
- 2. Detalhes da Solução
- 3. Gestão e Planejamento
- 4. Comunicação e Qualidade
- 5. Exercícios e Execuções
 - 5.1 Exercício 1 - IF/ELSE
 - 5.2 Exercício 2 - IF/ELIF/ELSE
 - 5.3 Exercício 3 - WHILE
 - 5.4 Exercício 4 - FOR
 - 5.5 Exercício 5 - Função Simples
 - 5.6 Exercício 6 - Função com Parâmetro
 - 5.7 Trabalho Prático – Calculadora_v2

1. Definição do Projeto

Problema e Justificativa

O problema identificado é a dificuldade dos estudantes em compreender os conceitos básicos de programação em Python.

Objetivos

Auxiliar no aprendizado de lógica de programação com Python através de exemplos práticos.

Visão

Fornecer material didático estruturado que permita ao estudante aprender e aplicar conceitos de programação em Python.

2. Detalhes da Solução

Solução Proposta

Implementação de atividades práticas que abordam variáveis, estruturas condicionais, laços de repetição e funções. Ao final, criação de uma calculadora interativa para consolidar os conhecimentos.

Metodologia

Aprendizagem prática com exemplos resolvidos, evoluindo a dificuldade até um projeto final integrador.

Entregáveis

Conjunto de exercícios resolvidos, documentação formal em PDF e programa de calculadora em Python.

3. Gestão e Planeamento

Cronograma

O projeto será desenvolvido em quatro semanas, com cada semana focando em um conceito específico: estruturas condicionais, laços, funções e projeto final.

Equipa e Responsabilidades

Estudante responsável pelo desenvolvimento das atividades; orientador responsável pela revisão e validação.

Orçamento

Recursos computacionais básicos (computador, Python e editor de código).

Riscos

Dificuldade de compreensão e problemas técnicos, mitigados com explicações detalhadas e uso de ferramentas acessíveis.

4. Comunicação e Qualidade

Plano de Comunicação

Reuniões periódicas e registro de atividades no PDF.

Garantia de Qualidade

Revisão de código, testes práticos e validação pelo orientador.

Métricas de Sucesso (KPIs)

Conclusão de todas as atividades, funcionamento correto da calculadora, clareza do documento e avaliação positiva do orientador.

5. Exercícios e Execuções

Exercício 1 - IF/ELSE

```
temperatura = 29
if temperatura < 30:
    print("A temperatura hoje está amena")
else:
    print("Hoje está fazendo calor")
# Execução:
# A temperatura hoje está amena
```

Explicação rápida:

- `if temperatura < 30:` → verifica se a temperatura é menor que 30.
- Se for **verdadeiro**, executa o `print("A temperatura hoje está amena")`.
- Caso contrário, o programa entra no `else` e executa `print("Hoje está fazendo calor")`.

Exercício 2 - IF/ELIF/ELSE

```
tempoExperiencia = 5
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.")
# Execução:
# Nível de conhecimento sênior.
```

Explicação rápida:

- **if tempoExperiencia < 2:** → verifica se o tempo é menor que 2.
- **elif tempoExperiencia > 2 and tempoExperiencia < 5:** → checa se está entre 2 e 5 (sem incluir).
- **else:** → se não se encaixar em nenhuma condição anterior, cai aqui, no caso "sênior".

Exercício 3 - WHILE

```
entrada_idade = ""
while str(entrada_idade) != "0":
    entrada_idade = input("Digite um número qualquer ou 0 para sair: ")
    print("Número digitado:", entrada_idade)
# Execução simulada:
# Digite 5
# Número digitado: 5
# Digite 0
# Número digitado: 0
```

Explicação rápida:

- `entrada_idade = ""` → começa como string vazia.
- `while str(entrada_idade) != "0":` → enquanto o valor **não for "0"**, **continua repetindo**.
- `input()` → pede ao usuário digitar um número.
- `print("Número digitado:", entrada_idade)` → mostra o número informado.

Quando o usuário digitar 0, a condição não será mais verdadeira e o programa **encerra o laço**.

Exercício 4 - FOR

```
texto = "Olá, laço for."
for item in texto:
    print("Caractere:", item)
for numero in range(1, 11):
    print("Número do intervalo:", numero)
# Execução parcial:
# Caractere: O
# Caractere: l
# Caractere: á
# ...
# Número do intervalo: 10
# Número do intervalo: 1
# Número do intervalo: 2
# Número do intervalo: 3
# ...
```

Explicação rápida:

- `for item in texto:` → percorre cada caractere da string "Olá, laço for."
- `print("Caractere: " + item)` → imprime caractere por caractere.
- `for numero in range(1, 11):` → gera os números de **1 até 10** (o 11 não entra).
- `str(numero)` → converte o número inteiro para string, permitindo a concatenação no `print()`.

Exercício 5 - Função Simples

```
def imprimir_variavel():
    texto = "Olá, funções em Python"
    print(texto)
    imprimir_variavel()
# Execução:
# Olá, funções em Python
```

Explicação rápida:

- `for item in texto:` → percorre cada caractere da string "Olá, laço for."
- `print("Caractere: " + item)` → imprime caractere por caractere.
- `for numero in range(1, 11):` → gera os números de **1 até 10** (o 11 não entra).
- `str(numero)` → converte o número inteiro para string, permitindo a concatenação no `print()`.

Exercício 6 - Função com Parâmetro

```
def loginUsuario(perfil):  
    if perfil.lower() == "admin":  
        print("Bem-vindo, Administrador")  
    else:  
        print("Bem-vindo, Usuário")  
loginUsuario("Admin")  
loginUsuario("admin")  
loginUsuario("User")  
loginUsuario("usuário")  
# Execução:  
# Bem-vindo, Administrador  
# Bem-vindo, Administrador  
# Bem-vindo, Usuário  
# Bem-vindo, Usuário
```

Explicação:

- `def loginUsuario(perfil):` → a função recebe um parâmetro chamado `perfil`.
- `perfil.lower()` → transforma todo o texto digitado em **minúsculas**, evitando erro se o usuário digitar `Admin`, `ADMIN`, etc.
- Se for igual a `"admin"`, imprime a mensagem de administrador.
- Caso contrário, imprime como usuário comum.
- No final, chamamos a função várias vezes com valores diferentes para testar.

Trabalho Prático – Calculadora_v2

```
saida = ''

def adicao(n1, n2): return n1 + n2
def subtracao(n1, n2): return n1 - n2
def multiplicacao(n1, n2): return n1 * n2
def divisao(n1, n2):
    if n2 == 0: return 'Erro: divisão por zero!'
    else: return n1 / n2

def calculadora(n1, n2, op):
    op = op.lower()
    if op == '+' or op == 'adicao':
        return adicao(n1, n2)
    if op == '-' or op == 'subtracao':
        return subtracao(n1, n2)
    if op == '*' or op == 'multiplicacao':
        return multiplicacao(n1, n2)
    if op == '/' or op == 'divisao':
        return divisao(n1, n2)
    else:
        return 'Operação invalida.'

saida = ''

while saida.lower() != 'n':
    try:
        n1 = float(input('Digite o primeiro número: '))
        n2 = float(input('Digite o segundo número: '))
        operacao = input('Digite a operação desejada (+, -, *, / ou nome):')

        resultado = calculadora(n1, n2, operacao)
        print(f'Resultado: {resultado}')

    except ValueError:
        print('Entrada invalida. Por favor digite apenas números.')
    except Exception as e:
        print('Ocorreu um erro: {e}')

    saida = input('Deseja realizar outra operação s/n: ')
    if saida.lower() != 's':
        saida = 'n'

print('Calculadora Encerrada.')
```

Explicação:

- Criamos **quatro funções básicas** (adicao, subtracao, multiplicacao, divisao).
- A função **divisao** verifica se o divisor é 0 antes de calcular.
- A função **calculadora** recebe dois números e a operação, e decide qual função chamar.
- No **while saida.lower() != "n":**, o programa só para quando o usuário digitar N ou n.
- Dentro do laço: pedimos os dois números e a operação → chamamos calculadora → mostramos o resultado.