

**ANO  
2025**



**UNINTER**

**ATIVIDADE PRÁTICA**  
**MÓDULO B**  
**LÓGICA DE PROGRAMAÇÃO**  
**E ALGORITMOS**

**Andrey ricardo da maia. RU: 5244478**

**Prof. Me. Bruno Kostiuk**

## QUESTÃO 1 de 4 – Conteúdos até Aula 3

**Enunciado:** Imagina-se que você é um dos programadores responsáveis pela construção de app de vendas para uma determinada empresa X que aceita cartões de crédito. Uma das estratégias de vendas dessa empresa X é cobrar um Juros maior conforme a quantidade de parcelas que o cliente desejar, conforme a **listagem abaixo**:

- Se a quantidade de parcelas for **menor** que **4**, o Juros será de **0%** (0 / 100);
- Se a quantidade de parcelas for **maior ou igual** que **4** e **menor que 6**, o Juros será de **4%** (4 / 100);
- Se a quantidade de parcelas for **maior ou igual** que **6** e **menor que 9**, o Juros será de **8%** (8 / 100);
- Se a quantidade de parcelas for **maior ou igual** que **9** e **menor que 13**, o Juros será de **16%** (16 / 100);
- Se a quantidade de parcelas for **maior ou igual** que **13**, o Juros será de **32%** (32 / 100);

O valor da parcela é calculado da seguinte maneira:

$$\text{valorDaParcela} = \frac{\text{valorDoPedido} * (1 + \text{juros})}{\text{quantidadeParcelas}}$$

O valor total parcelado é calculado da seguinte maneira:

$$\text{valorTotalParcelado} = \text{valorDaParcela} * \text{quantidadeParcelas}$$

**Elabore um programa em Python que:**


- Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). [EXIGÊNCIA DE CÓDIGO 1 de 6];
- Deve-se implementar o input do **valorDoPedido** e da **quantidadeParcelas** [EXIGÊNCIA DE CÓDIGO 2 de 6];
- Deve-se implementar o Juros **conforme a enunciado acima** (obs.: atente-se as condições de menor, igual e maior) [EXIGÊNCIA DE CÓDIGO 3 de 6];
- Deve-se implementar o **valorDaParcela** e **valorTotalParcelado** [EXIGÊNCIA DE CÓDIGO 4 de 6];
- Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
- Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 6 de 6];

**Teste seu código atendendo as seguintes exigências:**

- Deve-se apresentar na saída de console uma mensagem com seu nome e sobrenome [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];

H. Deve-se apresentar na saída de console um parcelamento com Juros (**quantidadeParcelas** maior ou igual a 4) apresentando o **valor da Parcela e o valor Total Parcelado** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2];

## EXEMPLO DE SAÍDA DE CONSOLE:



```
Bem-vindo a Loja do Bruno Kostiuik
Entre com o valor do pedido: 1000.00
Entre com a quantidade de parcelas: 14
O valor das parcelas é de:R$ 94.29
O valor Total Parcelado é de:R$ 1320.00
```

---

**Figura 1.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o valor do pedido (pode ser qualquer valor) a quantidade de parcelas (maior ou igual a 4 [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2]), e é apresentado o valorDaParcela e o valorTotalParcelado.**

## ***Apresentação de Código da Questão 1:***

```
# Print da mensagem de boas vindas
# print("Bem-vindo a loja do Andrey Ricardo da Maia")
#
# # Input do valor do pedido
# valorDoPedido = float(input("Digite o valor do pedido: R$ "))
#
# # Input da quantidade de parcelas
# quantidadeParcelas = int(input("Digite a quantidade de parcelas desejada: "))
#
# # Logica para o calculo de juros
# if quantidadeParcelas < 4:
#     juros = 0
# elif quantidadeParcelas >= 4 and quantidadeParcelas < 6:
#     juros = 0.04
# elif quantidadeParcelas >= 6 and quantidadeParcelas < 9:
#     juros = 0.08
# elif quantidadeParcelas >= 9 and quantidadeParcelas < 13:
#     juros = 0.16
# else:
#     juros = 0.32
#
# # Calculo do valor da parcela
# valorDaParcela = valorDoPedido * (1 + juros) / quantidadeParcelas
#
# # Calculo do total parcelado
# valorTotalParcelado = valorDaParcela * quantidadeParcelas
#
# # Print do valor de cada parcela
# print("Valor de cada parcela: R$ {:.2f}".format(valorDaParcela))
# Print do valor total com acrescimo de juros
print("Valor total parcelado: R$ {:.2f}".format(valorTotalParcelado))
```

*Apresentação de **Saída do Console da Questão 1:***

```
C:\Users\lucia\AppData\Local\Programs\Python\Python310\python.exe -u C:\Users\lucia\AppData\Local\Programs\Python\Python310\python.exe
 Bem-vindo a loja do Andrey Ricardo da Maia
 Digite o valor do pedido: R$ 1000
 Digite a quantidade de parcelas desejada: 14
 Valor de cada parcela: R$ 94.29
 Valor total parcelado: R$ 1320.00
```

## QUESTÃO 2 de 4 - Conteúdo até aula 04

**Enunciado:** Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma loja que vende Marmitas de Bife Acebolado ou Filé de Frango. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.

A Loja possui seguinte relação:

- Tamanho **P** de Bife Acebolado (**BA**) custa 16 reais e o Filé de Frango (**FF**) custa 15 reais;
- Tamanho **M** de Bife Acebolado (**BA**) custa 18 reais e o Filé de Frango (**FF**) custa 17 reais;
- Tamanho **G** de Bife Acebolado (**BA**) custa 22 reais e o Filé de Frango (**FF**) custa 21 reais;

**Elabore um programa em Python que:**


- Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). Além do seu nome e sobrenome, deve-se implementar um **print com um Menu** para o cliente. **[EXIGÊNCIA DE CÓDIGO 1 de 8];**
- Deve-se implementar o input do **sabor** (BA/FF) e o print "Sabor inválido. Tente novamente" se o usuário entra com valor diferente de BA e FF **[EXIGÊNCIA DE CÓDIGO 2 de 8];**
- Deve-se implementar o input do **tamanho** (P/M/G) e o print "Tamanho inválido. Tente novamente" se o usuário com entra valor diferente de P, M ou G **[EXIGÊNCIA DE CÓDIGO 3 de 8];**
- Deve-se implementar **if, elif e/ou else**, utilizando o modelo **aninhado** (aula 3 – Tema 4) com cada uma das combinações de **sabor** e **tamanho** **[EXIGÊNCIA DE CÓDIGO 4 de 8];**
- Deve-se implementar um **acumulador** para somar os valores dos pedidos **[EXIGÊNCIA DE CÓDIGO 5 de 8];**
- Deve-se implementar o input com a pergunta: "Deseja pedir mais alguma coisa?". Se sim **repetir a partir do item B**, senão encerrar o programa executar o print do **acumulador** **[EXIGÊNCIA DE CÓDIGO 6 de 8];**
- Deve-se implementar as estruturas de **while, break, continue (todas elas)** **[EXIGÊNCIA DE CÓDIGO 7 de 8];**
- Deve-se inserir comentários relevantes no código **[EXIGÊNCIA DE CÓDIGO 8 de 8];**

**Teste seu código atendendo as seguintes exigências:**

- I. Deve-se apresentar na saída de console uma mensagem com o seu nome completo e o menu para o cliente conhecer as opções [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
- J. Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
- K. Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
- L. Deve-se apresentar na saída de console um pedido com duas opções sabores diferentes e com tamanhos diferentes [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];



EXEMPLO DE SAÍDA DE CONSOLE:



```
----- Bem-vindo a Loja de Marmitas do Bruno KostiuK -----
-----Cardápio-----
---| Tamanho | Bife Acebolado(BA) | Filé de Frango(FF) |---
---| P        | R$ 16.00           | R$ 15.00           |---
---| M        | R$ 18.00           | R$ 17.00           |---
---| G        | R$ 22.00           | R$ 21.00           |---
-----
Entre com o sabor desejado (BA/FF): BF
Sabor inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): EXGG
Tamanho inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): M
Você pediu um Bife Acebolado no tamanho M: R$ 18.00

Deseja mais alguma coisa? (S/N): S
Entre com o sabor desejado (BA/FF): FF
Entre com o tamanho desejado (P/M/G): G
Você pediu um Filé de Frango no tamanho G: R$ 21.00

Deseja mais alguma coisa? (S/N): N

O valor total a ser pago: R$ 39.00
```

Mensagem com seu nome completo e Menu de opções (cardápio)

Usuário errou o sabor

Usuário errou o tamanho

Pedido com 2 itens de tamanhos e sabores diferentes

**Figura 2.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o sabor e o tamanho. Há uma tentativa de pedido que se errou o sabor e outra que se errou o tamanho. Há também um pedido com dois itens com sabores e tamanhos diferentes.**

## Apresentação de **Código da Questão 2:**

```
# Print de boas vindas
print("Bem-vindo à Marmitaria do Andrey Ricardo da Maia")

# Print do menu
print("Cardápio:")
print("BA - Bife Acebolado")
print("FF - Filé de Frango")
print("Tamanhos: P (Pequeno), M (Médio), G (Grande)\n")
# Fim do print do menu

# Totalizador de pedidos
total = 0

# Loop infinito para manter o usuario preso no menu
while True:
    sabor = input("Digite o sabor desejado (BA ou FF): ")

    if sabor != "BA" and sabor != "FF":
        print("Sabor inválido. Tente novamente\n")
        continue

    tamanho = input("Digite o tamanho desejado (P, M ou G): ")

    # Menu de seleção de tamanho
    if tamanho != "P" and tamanho != "M" and tamanho != "G":
        print("Tamnho inválido. Tente novamente\n")
        continue
    if sabor == "BA":
        if tamanho == "P":
            preco = 16
        elif tamanho == "M":
            preco = 18
        else:
            preco = 22
    elif sabor == "FF":
        if tamanho == "P":
            preco = 15
        elif tamanho == "M":
            preco = 17
        else:
```

```
        preco = 21
    else:
        print("Errrrrou!")

    # Adiciona no totalizador
    total = total + preco

    # Print do status atual
    print("Pedido adicionado: {} tamanho {} - R$ {:.2f}\n".format(sabor, tamanho,
preco))

# Controle de encerramento de pedido
continuar = input("Deseja pedir mais alguma coisa? (s/n): ")
if continuar == "s":
    continue
elif continuar == "n":
    break
else:
    print("Resposta inválida, mas vamos encerrar mesmo assim.\n")
    break

print("\nValor total do seu pedido: R$", total)
```

*Apresentação de **Saída do Console da Questão 2:***

```
C:\Users\lilid\AppData\Local\Programs\Python\Python313\python.exe "C
Bem-vindo à Marmitaria do Andrey Ricardo da Maia
Cardápio:
BA - Bife Acebolado
FF - Filé de Frango
Tamanhos: P (Pequeno), M (Médio), G (Grande)

Digite o sabor desejado (BA ou FF): BF
Sabor inválido. Tente novamente

Digite o sabor desejado (BA ou FF): BA
Digite o tamanho desejado (P, M ou G): EX66
Tamnho inválido. Tente novamente

Digite o sabor desejado (BA ou FF): BA
Digite o tamanho desejado (P, M ou G): M
Pedido adicionado: BA tamanho M - R$ 18.00

Deseja pedir mais alguma coisa? (s/n): S
Resposta inválida, mas vamos encerrar mesmo assim.

Valor total do seu pedido: R$ 18
```

## QUESTÃO 3 de 4 - Conteúdo até aula 05

**Enunciado:** Você foi contratado para desenvolver um sistema de cobrança de serviços de uma fábrica que vende Camisetas em atacado. Você ficou com a parte de desenvolver a interface com o funcionário.

A Fábrica opera as vendas da seguinte maneira:

- Camiseta Manga Curta Simples (**MCS**), o valor unitário é de um real e oitenta centavos;
  - Camiseta Manga Longa Simples (**MLS**), o valor unitário é de dois reais e dez centavos;
  - Camiseta Manga Curta Com Estampa (**MCE**), o valor unitário é de dois reais e noventa centavos;
  - Camiseta Manga Longa Com Estampa (**MLE**), o valor unitário é de três reais e vinte centavos;
- 
- Se número de camisetas for **menor** que 20 não há desconto na venda;
  - Se número de camisetas for **igual ou maior** que 20 e **menor** que 200, o desconto será de 5%;
  - Se número de camisetas for **igual ou maior** que 200 e **menor** que 2000, o desconto será de 7%;

- Se número de camisetas for **igual ou maior** que 2000 e **menor ou igual** que 20000, o desconto será de 12%;
- Se número de camisetas for **maior** que 20000, não é aceito pedidos nessa quantidade de camisetas;
- ♦ Para o **adicional** de frete por transportadora (1) é cobrado um valor **extra** de 100 reais;
- ♦ Para o **adicional** de frete por Sedex (2) é cobrado um valor **extra** de 200 reais;
- ♦ Para o **adicional** de retirar o pedido na fábrica (0) é cobrado um valor **extra** de 0 reais;

O valor final da conta é calculado da seguinte maneira:

$$\text{total} = (\text{modelo} * \text{num\_camisetas}) + \text{frete}$$

### Elabore um programa em Python que:

- A. Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui). [EXIGÊNCIA DE CÓDIGO 1 de 7];
- B. Deve-se implementar a função **escolha\_modelo()** em que: [EXIGÊNCIA DE CÓDIGO 2 de 7];
  - a. Pergunta o **modelo** desejado;
  - b. **Retorna o valor do modelo** com base na escolha do usuário (use **return**);
  - c. Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
- C. Deve-se implementar a função **num\_camisetas()** em que: [EXIGÊNCIA DE CÓDIGO 3 de 7];
  - a. Pergunta o **número de camisetas**;
  - b. **Retorna** (use **return**) o **número de camisetas** com desconto seguindo a regra do enunciado (desconto calculado em cima do número de camisetas);
  - c. Repete a pergunta do item **C.a** se digitar um valor acima de 20000 ou valor não numérico (use try/except para não numérico)
- D. Deve-se implementar a função **frete()** em que: [EXIGÊNCIA DE CÓDIGO 4 de 7];
  - a. Pergunta pelo serviço **adicional de frete**;
  - b. **Retorna** (use **return**) o **valor** de apenas uma das **opções de frete**
  - c. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/0;
- E. Deve-se implementar o total a pagar no código principal (**main**), ou seja, **não pode estar dentro de função**, conforme o enunciado [EXIGÊNCIA DE CÓDIGO 5 de 7];
- F. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
- G. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];

### Teste seu código atendendo as seguintes exigências:

- H. Deve-se apresentar na saída de console uma mensagem com o seu **nome e sobrenome** [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
- I. Deve-se apresentar na saída de console um pedido no qual o usuário errou a opção de modelo [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];

- J. Deve-se apresentar na saída de console um pedido no qual o usuário digitou ultrapassou no número de camisetas [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
- K. Deve-se apresentar na saída de console um pedido com opção de modelo, número de camisetas e frete válidos [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

### EXEMPLO DE SAÍDA DE CONSOLE:

```
➔ Bem vindo a Fábrica de Camisetas do do Bruno Kostiuk

Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MLL                                     Errou o modelo
Escolha inválida, entre com o modelo novamente

Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MCS
Entre com o número de camisetas: 300000    Errou o número de camisetas
Não aceitamos tantas camisetas de uma vez.
Por favor, entre com o número de camisetas novamente.

Entre com o número de camisetas: 10000

Escolha o tipo de frete:
1 - Frete por transportadora - R$ 100.00
2 - Frete por Sedex - R$ 200.00
0 - Retirar pedido na fábrica - R$ 0.00
>>2                                     Pedido com modelo, número de
                                         camisetas e frete válidos
Total: R$ 16040.00 (Modelo: 1.80 * Quantidade(com desconto): 8800 + frete: 200.00)
```

**Figura 3.1: Exemplo de saída de console que o aluno deve fazer. Em que se pergunta pelo modelo e se erra opção inicialmente, que se passa do número de camisetas acima do aceito. Na sequência, o usuário digitou um modelo, número de camisetas e frete válidos.**

## Apresentação de Código da Questão 3:

```
print("Bem vindo ao sistema da fábrica do Andrey Ricardo da Maia\n")

# Metodo para escolher o modelo
def escolha_modelo():
    while True:
        modelo = input("Escolha o modelo (MCS, MLS, MCE, MLE): ")

        if modelo == "MCS":
            return 1.80
        elif modelo == "MLS":
            return 2.10
```

```
elif modelo == "MCE":  
    return 2.90  
elif modelo == "MLE":  
    return 3.20  
else:  
    print("Errrrrou!\n")
```

# Metodo para quantidade de camisetas com desconto

```
def num_camisetas():  
    while True:  
        try:  
            quantidade = int(input("Digite o número de camisetas: "))  
  
            if quantidade > 20000:  
                print("Nao aceitamos pedidos com mais de 20000 camisetas\n")  
                continue  
  
            if quantidade < 20:  
                desconto = 0  
            elif quantidade >= 20 and quantidade < 200:  
                desconto = 0.05  
            elif quantidade >= 200 and quantidade < 2000:  
                desconto = 0.07  
            elif quantidade >= 2000 and quantidade <= 20000:  
                desconto = 0.12  
            else:  
                desconto = 0  
  
            return quantidade - (desconto * quantidade)  
  
        except:  
            print("Valor invalido. Digite um numero inteiro!\n")
```

# Metodo do frete

```
def frete():  
    while True:  
        print("Opções de frete:")  
        print("0 - Retirar na fábrica (Grátis)")  
        print("1 - Transportadora (R$ 100)")  
        print("2 - Sedex (R$ 200)")
```



```
tipo_frete = input("Digite o tipo de frete (0, 1 ou 2): ")

if tipo_frete == "0":
    return 0
elif tipo_frete == "1":
    return 100
elif tipo_frete == "2":
    return 200
else:
    print("Opção de frete inválido. Tente dnv\n")
```

```
preco_modelo = escolha_modelo()
```

```
quantidade_com_desconto = num_camisetas()
```

```
valor_frete = frete()
```

```
# Totalizando
```

```
total = (preco_modelo * quantidade_com_desconto) + valor_frete
```

```
# Print do resumo do pedido
```

```
print("\nResumo do Pedido")
```

```
print("Preço unitario do modelo escolhido: R$ ", preco_modelo)
```

```
print("Quantidade final com desconto: ", quantidade_com_desconto)
```

```
print("Valor do frete: R$ ", valor_frete)
```

```
print("Total a pagar: R$ ", total)
```

```
# Fim do resumo do pedido
```

*Apresentação de Saída do Console da Questão 3:*

Bem vindo ao sistema da fábrica do Andrey Ricardo da Maia

Escolha o modelo (MCS, MLS, MCE, MLE): *MLL*

Errrrrou!

Escolha o modelo (MCS, MLS, MCE, MLE): *MCS*

Digite o número de camisetas: *3000000*

Nao aceitamos pedidos com mais de 20000 camisetas

Digite o número de camisetas: *10000*

Opções de frete:

0 - Retirar na fábrica (Grátis)

1 - Transportadora (R\$ 100)

2 - Sedex (R\$ 200)

Digite o tipo de frete (0, 1 ou 2): *2*

Resumo do Pedido

Preço unitario do modelo escolhido: R\$ 1.8

Quantidade final com desconto: 8800.0

Valor do frete: R\$ 200

Total a pagar: R\$ 16040.0

## QUESTÃO 4 de 4 - Conteúdo até aula 06

**Enunciado:** Você e sua equipe de programadores foram contratados por pequena empresa para desenvolver o software de gerenciamento de funcionários. Este software deve ter o seguinte menu e opções:

- 1) Cadastrar Funcionário
- 2) Consultar Funcionário
  1. Consultar Todos
  2. Consultar por Id
  3. Consultar por setor
  4. Retornar ao menu
- 3) Remover Funcionário
- 4) Encerrar Programa

**Elabore um programa em Python que:**

- A. Deve-se implementar o print com uma mensagem de boas-vindas que apareça o seu **nome e sobrenome** (somente print, não usar input aqui) [EXIGÊNCIA DE CÓDIGO 1 de 8];
- B. Deve-se implementar uma lista com o nome de **lista\_funcionarios** e a variável **id\_global** com valor inicial **igual ao número de seu RU** [EXIGÊNCIA DE CÓDIGO 2 de 8];
- C. Deve-se implementar uma função chamada **cadastrar\_funcionario(id)** em que: [EXIGÊNCIA DE CÓDIGO 3 de 8];
  - a. Pergunta **nome, setor, salario** do funcionário;
  - b. Armazena o **id** (este é fornecido via parâmetro da função), **nome, setor, salario** dentro de um dicionário;
  - c. **Copiar** o dicionário para dentro da **lista\_funcionarios**, usando **.copy()** ou similar;
- D. Deve-se implementar uma função chamada **consultar\_funcionarios()** em que: [EXIGÊNCIA DE CÓDIGO 4 de 8];
  - a. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
    - i. Se Consultar Todos, apresentar todos os funcionários com todos os seus dados cadastrados;
    - ii. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o funcionário específico com todos os seus dados cadastrados;
    - iii. Se Consultar por Setor, solicitar ao usuário que informe o setor, e apresentar o(s) funcionário(s) do setor com todos os seus dados cadastrados;
    - iv. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
    - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **D.a.**
    - vi. Enquanto o usuário não escolher a opção 4, o menu consultar funcionários deve se repetir.

- E. Deve-se implementar uma função chamada **remover\_funcionario()** em que: **[EXIGÊNCIA DE CÓDIGO 5 de 8];**
- Deve-se pergunta pelo **id** do funcionário a ser removido;
  - Remover o funcionário da **lista\_funcionarios**;
  - Se o id fornecido não for de um funcionário da lista, printar "Id inválido" e repetir a pergunta **E.a.**
- F. Deve-se implementar uma estrutura de menu no código principal (**main**), ou seja, **não pode estar dentro de função**, em que: **[EXIGÊNCIA DE CÓDIGO 6 de 8];**
- Deve-se pergunta qual opção deseja (1. Cadastrar Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
    - Se Cadastrar Funcionário, chamar a função **cadaststrar\_funcionario(id\_global)** e **em seguida, incrementar** em um **id\_global** (no menu principal);
    - Se Consultar Funcionário, chamar função **consultar\_funcionario ()**;
    - Se Remover Funcionário, chamar função **remover\_funcionario()**;
    - Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
    - Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **F.a.**
    - Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
- G. Deve-se implementar uma **lista de dicionários** (uma lista contento dicionários dentro) **[EXIGÊNCIA DE CÓDIGO 7 de 8];**
- H. Deve-se inserir comentários relevantes no código **[EXIGÊNCIA DE CÓDIGO 8 de 8];**

### Teste seu código atendendo as seguintes exigências:

- I. Deve-se apresentar na saída de console uma mensagem com o seu **nome e sobrenome** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6];**
- J. Deve-se apresentar na saída de console um cadastro de 3 funcionários (sendo **2** deles no **mesmo setor**) **[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 6];**
- K. Deve-se apresentar na saída de console uma **consulta de todos os funcionários** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6];**
- L. Deve-se apresentar na saída de console uma **consulta por código (id)** de um dos funcionários **[EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6];**
- M. Deve-se apresentar na saída de console uma **consulta por setor** em que **2** funcionários sejam do mesmo setor **[EXIGÊNCIA DE SAÍDA DE CONSOLE 5 de 6];**
- N. Deve-se apresentar na saída de console **uma remoção** de um dos funcionários seguida de uma **consulta de todos** os funcionários **[EXIGÊNCIA DE SAÍDA DE CONSOLE 6 de 6];**

## EXEMPLO DE SAÍDA DE CONSOLE:

```
➡ Bem vindo a Empresa do Bruno Kostiuik      Mensagem com seu nome completo
-----
----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
-----
----- MENU CADASTRAR FUNCIONÁRIO -----
Id do Funcionário: 4297914
Por favor entre com o nome do Funcionário: Bruno      Cadastro do primeiro funcionário
Por favor entre com o setor do Funcionário: Configurações
Por favor entre com o salário do Funcionário: 1000.00
-----
```



**Figura 4.1: Exemplo de saída de console que o aluno deve fazer. Apresenta o print com seu nome completo e é realizado o cadastro do primeiro funcionário, note que o ID do funcionário não inicia em 1, pois ele deve iniciar com o seu RU (caso o RU informado não seja o seu, irá receber zero em toda questão).**

```
⇄ ----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
-----

----- MENU CADASTRAR FUNCIONÁRIO -----
Id do Funcionário: 4297915
Por favor entre com o nome do Funcionário: Tamy
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000
-----


----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
-----

----- MENU CADASTRAR FUNCIONÁRIO -----
Id do Funcionário: 4297916
Por favor entre com o nome do Funcionário: Osmar
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000.00
-----
```



**Cadastra mais dois funcionários do mesmo setor**

**Figura 4.2: Exemplo de saída de console que o aluno deve fazer. São cadastrados mais dois funcionários com mesmo setor.**



```
-----  
----- MENU PRINCIPAL -----  
Escolha a opção desejada:  
1 - Cadastrar Funcionários  
2 - Consultar Funcionário(s)  
3 - Remover Funcionário  
4 - Sair  
>>2  
-----  
----- MENU CONSULTAR FUNCIONÁRIO -----  
Escolha a opção desejada:  
1 - Consultar Todos os Funcionários  
2 - Consultar Funcionário por id  
3 - Consultar Funcionário(s) por setor  
4 - Retornar  
>>1  
-----  
id: 4297914  
nome: Bruno  
setor: Configurações  
salário: 1000.0  
  
id: 4297915  
nome: Tamy  
setor: Tutorias  
salário: 1000.0  
  
id: 4297916  
nome: Osmar  
setor: Tutorias  
salário: 1000.0  
-----
```

**Consulta todos os  
funcionários cadastrados**

**Figura 4.3: Exemplo de saída de console que o aluno deve fazer. Em que se consulta Todos os funcionários cadastrados.**



```
-----  
----- MENU CONSULTAR FUNCIONÁRIO -----  
Escolha a opção desejada:  
1 - Consultar Todos os Funcionários  
2 - Consultar Funcionário por id  
3 - Consultar Funcionário(s) por setor  
4 - Retornar  
>>2  
Digite o id do funcionário: 4297914  
-----  
id: 4297914  
nome: Bruno  
setor: Configurações  
salario: 1000.0  
  
-----  
  
-----  
----- MENU CONSULTAR FUNCIONÁRIO -----  
Escolha a opção desejada:  
1 - Consultar Todos os Funcionários  
2 - Consultar Funcionário por id  
3 - Consultar Funcionário(s) por setor  
4 - Retornar  
>>3  
Digite o setor do(s) funcionário(s): Tutorias  
-----  
id: 4297915  
nome: Tamy  
setor: Tutorias  
salario: 1000.0  
  
id: 4297916  
nome: Osmar  
setor: Tutorias  
salario: 1000.0
```

**Consulta por ID**

**Consulta por Setor**

**Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o funcionário com id número 4297914 e consulta pelo nome do setor (Tutorias).**





```
----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>3

----- MENU REMOVER FUNCIONÁRIO -----
Digite o id do funcionario a ser removido: 4297914
Funcionário removido com sucesso!

----- MENU PRINCIPAL -----
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>2

----- MENU CONSULTAR FUNCIONÁRIO -----
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>1

id: 4297915
nome: Tamy
setor: Tutorias
salário: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salário: 1000.0
```

**Remove um  
funcionário**

**Realiza o consultar Todos mostrando  
que o funcionário foi removido**

**Figura 4.5: Exemplo de saída de console que o aluno deve fazer. Em que se remove o funcionário de Id número 4297914 e depois se faz uma consulta de todos os funcionários.**

## Apresentação de **Código da Questão 4:**

```
print("Bem vindo ao sistema de cadastro da empresa - Andrey Ricardo da Maia\n")

# Lista de funcionários e id inicial
lista_funcionarios = []
id_global = 5244478

# Metodo para cadastrar
def cadastrar_funcionario(id):
    nome = input("Digite o nome do funcionario: ")
    setor = input("Digite o setor do funcionario: ")
    salario = input("Digite o salario do funcionario: ")

    funcionario = {}
    funcionario["id"] = id
    funcionario["nome"] = nome
    funcionario["setor"] = setor
    funcionario["salario"] = salario

    lista_funcionarios.append(funcionario.copy())
    print("Funcionario cadastrado com sucesso!\n")

# Funcao para consultar
def consultar_funcionarios():
    while True:
        print("Menu de consulta:")
        print("1 - Consultar todos")
        print("2 - Consultar por ID")
        print("3 - Consultar por setor")
        print("4 - Voltar ao menu principal")

        opcao = input("Escolha uma opcao: ")

        if opcao == "1":
            for func in lista_funcionarios:
                print("ID:", func["id"])
                print("Nome:", func["nome"])
                print("Setor:", func["setor"])
                print("Salario:", func["salario"])
                print("-----")
        elif opcao == "2":
            id_buscado = input("Digite o ID do funcionario: ")
```

```

        achou = False
        for func in lista_funcionarios:
            if str(func["id"]) == id_buscado:
                print("ID:", func["id"])
                print("Nome:", func["nome"])
                print("Setor:", func["setor"])
                print("Salario:", func["salario"])
                print("-----")
                achou = True

        if not achou:
            print("Funcionario nao encontrado!\n")
    elif opcao == "3":
        setor_buscado = input("Digite o setor: ")
        for func in lista_funcionarios:
            if func["setor"] == setor_buscado:
                print("ID:", func["id"])
                print("Nome:", func["nome"])
                print("Setor:", func["setor"])
                print("Salario:", func["salario"])
                print("-----")
    elif opcao == "4":
        return

```

# Funcao para remover

```

def remover_funcionario():
    while True:
        id_remover = input("Digite o ID do funcionario que deseja remover: ")
        encontrou = False
        for func in lista_funcionarios:
            if str(func["id"]) == id_remover:
                lista_funcionarios.remove(func)
                print("Funcionario removido com sucesso!\n")
                encontrou = True
                break
        if encontrou:
            break
        else:
            print("ID invalido, tente novamente!\n")

```

# Menu principal

```

while True:
    print("Menu principal:")
    print("1 - Cadastrar Funcionario")

```

```

print("2 - Consultar Funcionario")
print("3 - Remover Funcionario")
print("4 - Encerrar Programa")

escolha = input("Escolha uma opcao: ")

if escolha == "1":
    cadastrar_funcionario(id_global)
    id_global = id_global + 1
elif escolha == "2":
    consultar_funcionarios()
elif escolha == "3":
    remover_funcionario()
elif escolha == "4":
    print("Programa encerrado. Obrigado!")
    break

```

## Apresentação de Saída do Console da Questão 4:

```

C:\Users\lilid\AppData\Local\Programs\Python\Python313\python.exe "C:\Users\lilid\PycharmProjects\trabalho atividade pratica modulo b\main.py"
Bem vindo ao sistema de cadastro da empresa - Andrey Ricardo da Maia

Menu principal:
1 - Cadastrar Funcionario
2 - Consultar Funcionario
3 - Remover Funcionario
4 - Encerrar Programa
Escolha uma opcao: 1
Digite o nome do funcionario: bruno
Digite o setor do funcionario: configuracoes
Digite o salario do funcionario: 1000.00
Funcionario cadastrado com sucesso!

Menu principal:
1 - Cadastrar Funcionario
2 - Consultar Funcionario
3 - Remover Funcionario
4 - Encerrar Programa
Escolha uma opcao: 1
Digite o nome do funcionario: tamy
Digite o setor do funcionario: tutorias
Digite o salario do funcionario: 1000
Funcionario cadastrado com sucesso!

Menu principal:

```

PC

TB trabalho atividade pratica modulo b

Version control

main

main

main.py

Run

main

Menu principal:  
1 - Cadastrar Funcionario  
2 - Consultar Funcionario  
3 - Remover Funcionario  
4 - Encerrar Programa  
Escolha uma opcao: 1  
Digite o nome do funcionario: osmar  
Digite o setor do funcionario: tutorias  
Digite o salario do funcionario: 1000.00  
Funcionario cadastrado com sucesso!  
  
Menu principal:  
1 - Cadastrar Funcionario  
2 - Consultar Funcionario  
3 - Remover Funcionario  
4 - Encerrar Programa  
Escolha uma opcao: 2  
Menu de consulta:  
1 - Consultar todos  
2 - Consultar por ID  
3 - Consultar por setor  
4 - Voltar ao menu principal  
Escolha uma opcao: 1  
ID: 5244478  
Nome: bruno

trabalho atividade pratica modulo b > main.py

7:24 (3362 chars, 99 line breaks) CRLF UTF-8 4 spaces Python 3.13

11°C Limpo

Pesquisar

22:43 05/07/2025

PC

TB trabalho atividade pratica modulo b

Version control

main

main

main.py

Run

main

ID: 5244478  
Nome: bruno  
Setor: configurações  
Salario: 1000.00  
-----  
ID: 5244479  
Nome: tamy  
Setor: tutorias  
Salario: 1000  
-----  
ID: 5244480  
Nome: osmar  
Setor: tutorias  
Salario: 1000.00  
-----  
Menu de consulta:  
1 - Consultar todos  
2 - Consultar por ID  
3 - Consultar por setor  
4 - Voltar ao menu principal  
Escolha uma opcao: 2  
Digite o ID do funcionario: 5244478  
ID: 5244478  
Nome: bruno  
Setor: configurações  
Salario: 1000.00

trabalho atividade pratica modulo b > main.py

7:24 (3362 chars, 99 line breaks) CRLF UTF-8 4 spaces Python 3.13

11°C Limpo

Pesquisar

22:46 05/07/2025

```
Setor: configurações
Salario: 1000.00
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao:
Digite o setor: tutorias
ID: 5244479
Nome: tamy
Setor: tutorias
Salario: 1000
-----
ID: 5244480
Nome: osmar
Setor: tutorias
Salario: 1000.00
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 3
```

```
4 - Voltar ao menu principal
Escolha uma opcao: 3
Digite o setor: tutorias
ID: 5244479
Nome: tamy
Setor: tutorias
Salario: 1000
-----
ID: 5244480
Nome: osmar
Setor: tutorias
Salario: 1000.00
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 2
Digite o ID do funcionario: 5244478
ID: 5244478
Nome: bruno
Setor: configurações
Salario: 1000.00
-----
Menu de consulta:
```

```
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 3
Digite o setor: tutorias
ID: 5244479
Nome: tamy
Setor: tutorias
Salario: 1000
-----
ID: 5244480
Nome: osmar
Setor: tutorias
Salario: 1000.00
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 4
Menu principal:
1 - Cadastrar Funcionario
```

```
Menu principal:
1 - Cadastrar Funcionario
2 - Consultar Funcionario
3 - Remover Funcionario
4 - Encerrar Programa
Escolha uma opcao: 3
Digite o ID do funcionario que deseja remover: 5244478
ID invalido, tente novamente!

Digite o ID do funcionario que deseja remover: 5244478
Funcionario removido com sucesso!

Menu principal:
1 - Cadastrar Funcionario
2 - Consultar Funcionario
3 - Remover Funcionario
4 - Encerrar Programa
Escolha uma opcao: 2
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 1
ID: 5244479
Nome: tamy
```

The image shows a Windows desktop with a VS Code editor window open. The editor has a dark theme and displays a Python file named `main.py`. The file contains a script for a simple employee management system. The script defines two classes, `Funcionario` and `Setor`, and a `Menu` class that handles user input and displays data. The script is executed, and the output is shown in the Run console at the bottom of the editor. The output displays the details of two employees, a menu of options, and the user's selection of option 4 to view all employees. The script ends with a message indicating it has finished execution with exit code 0.

```
class Funcionario:
    def __init__(self, id, nome, salario):
        self.id = id
        self.nome = nome
        self.salario = salario

class Setor:
    def __init__(self, id, nome):
        self.id = id
        self.nome = nome

class Menu:
    def __init__(self):
        self.opcao = 0

    def menu_principal(self):
        print("Menu principal:")
        for i in range(1, 5):
            print(i, "- ", end="")
            if i % 4 == 1:
                print("Cadastrar Funcionario")
            elif i % 4 == 2:
                print("Consultar Funcionario")
            elif i % 4 == 3:
                print("Remover Funcionario")
            elif i % 4 == 4:
                print("Encerrar Programa")
            if i % 4 != 1:
                print(" ")
        print()

    def menu_consulta(self):
        print("Menu de consulta:")
        for i in range(1, 5):
            print(i, "- ", end="")
            if i % 4 == 1:
                print("Consultar todos")
            elif i % 4 == 2:
                print("Consultar por ID")
            elif i % 4 == 3:
                print("Consultar por setor")
            elif i % 4 == 4:
                print("Voltar ao menu principal")
            if i % 4 != 1:
                print(" ")
        print()

    def executar(self):
        while True:
            self.menu_principal()
            self.opcao = int(input("Escolha uma opcao: "))
            if self.opcao == 1:
                # Cadastrar Funcionario
                pass
            elif self.opcao == 2:
                # Consultar Funcionario
                pass
            elif self.opcao == 3:
                # Remover Funcionario
                pass
            elif self.opcao == 4:
                # Encerrar Programa
                break
            else:
                print("Opcao invalida. Tente novamente.")

    def executar_consulta(self):
        while True:
            self.menu_consulta()
            self.opcao = int(input("Escolha uma opcao: "))
            if self.opcao == 1:
                # Consultar todos
                for funcionario in funcionarios:
                    print(f"ID: {funcionario.id}\nNome: {funcionario.nome}\nSetor: {funcionario.setor.nome}\nSalario: {funcionario.salario}\n-----")
            elif self.opcao == 2:
                # Consultar por ID
                id = int(input("Digite o ID do funcionario: "))
                for funcionario in funcionarios:
                    if funcionario.id == id:
                        print(f"ID: {funcionario.id}\nNome: {funcionario.nome}\nSetor: {funcionario.setor.nome}\nSalario: {funcionario.salario}\n-----")
            elif self.opcao == 3:
                # Consultar por setor
                setor_id = int(input("Digite o ID do setor: "))
                for funcionario in funcionarios:
                    if funcionario.setor.id == setor_id:
                        print(f"ID: {funcionario.id}\nNome: {funcionario.nome}\nSetor: {funcionario.setor.nome}\nSalario: {funcionario.salario}\n-----")
            elif self.opcao == 4:
                # Voltar ao menu principal
                break
            else:
                print("Opcao invalida. Tente novamente.")

funcionarios = []
setores = []

setor1 = Setor(1, "Tutorias")
setor2 = Setor(2, "Tutorias")
setores.append(setor1)
setores.append(setor2)

funcionario1 = Funcionario(1, "Tamy", 1000)
funcionario2 = Funcionario(2, "Osmar", 1000)
funcionarios.append(funcionario1)
funcionarios.append(funcionario2)

menu = Menu()
menu.executar()

Process finished with exit code 0
```

The Run console at the bottom of the editor shows the following output:

```
ID: 5244479
Nome: tamy
Setor: tutorias
Salario: 1000
-----
ID: 5244480
Nome: osmar
Setor: tutorias
Salario: 1000.00
-----
Menu de consulta:
1 - Consultar todos
2 - Consultar por ID
3 - Consultar por setor
4 - Voltar ao menu principal
Escolha uma opcao: 4
Menu principal:
1 - Cadastrar Funcionario
2 - Consultar Funcionario
3 - Remover Funcionario
4 - Encerrar Programa
Escolha uma opcao: 4
Programa encerrado. Obrigado!

Process finished with exit code 0
```

The status bar at the bottom of the editor shows the file path `trabalho atividade pratica modulo b > main.py`, the character count `7:24 (3362 chars, 99 line breaks)`, the encoding `CRLF`, the file size `UTF-8`, the indentation `4 spaces`, and the Python version `Python 3.13`.











