

**ANO
2025**



UNINTER

ATIVIDADE PRÁTICA

MÓDULO B

LÓGICA DE PROGRAMAÇÃO E ALGORITMOS

BRUNO CESAR LEZCANO VIANA. RU: 4512355

Prof. Me. Bruno Kostiuk

INSTRUÇÕES

SOMENTE INFORMATIVO. PODE APAGAR AO ENTREGAR O TRABALHO.

Esta atividade deve ser desenvolvida individualmente em linguagem Python, não sendo permitido a utilização de Inteligência Artificial para resolução das questões. A entrega deve ser feita na seção Trabalhos do Univirtus. Depois do arquivo enviado, não há possibilidade de reenvio. Verifique o arquivo enviado antes de confirmar a entrega. Só envie seu trabalho quando tiver certeza de que está tudo correto. Preste muita atenção ao prazo! Não haverá prorrogação.

Só serão aceitas submissões em formato doc ou docx. Pois estes formatos mantêm a indentação do Python, facilitando a execução dos códigos de vocês.

Em caso de dificuldade na elaboração ou publicação do trabalho, envie uma tutoria!

Iremos auxiliar na tutoria, desde que não envolva fazer uma correção prévia do trabalho. Na tutoria, nós damos dicas de como solucionar o problema, mas a solução é com vocês alunos.

As quatro questões não poderão ser feitas por qualquer tipo de Inteligência Artificial. Caso seja identificado esse método de solução, o trabalho receberá nota zero automaticamente.

A proposta deste trabalho é que o aluno coloque em prática e demonstre suas competências e habilidades adquiridas ao longo da disciplina. Para isso, pede-se neste trabalho a confecção e entrega das quatro questões a seguir:

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:

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

O valor total parcelado é calculado da seguinte maneira:

$$valorTotalParcelado = valorDaParcela * quantidadeParcelas$$


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];
- 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 Kostiuk
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('Tudo bem Bruno Cesar Lezcano Viana! Calculadora de Juros')      #MENSAGEM DE
BOAS VINDAS
print()
vlr_pedido = float(input('Qual é o valor do pedido? R$_'))
qtd_parcela = int(input('Qual a quantidade de parcelas desejada?  '))

if (qtd_parcela < 4):                                                  #CONDIÇÃO PARA
PARCELA MENOR QUE 3 VEZES
    juros = 0/100
    vlr_parcela = vlr_pedido * (1 + juros) / qtd_parcela
    vlr_total_parcelado = vlr_parcela * qtd_parcela
    print(f'O valor do parcela é de R${vlr_parcela:.2f}')
    print(f'O valor total parcelado é de R${vlr_total_parcelado:.2f}')

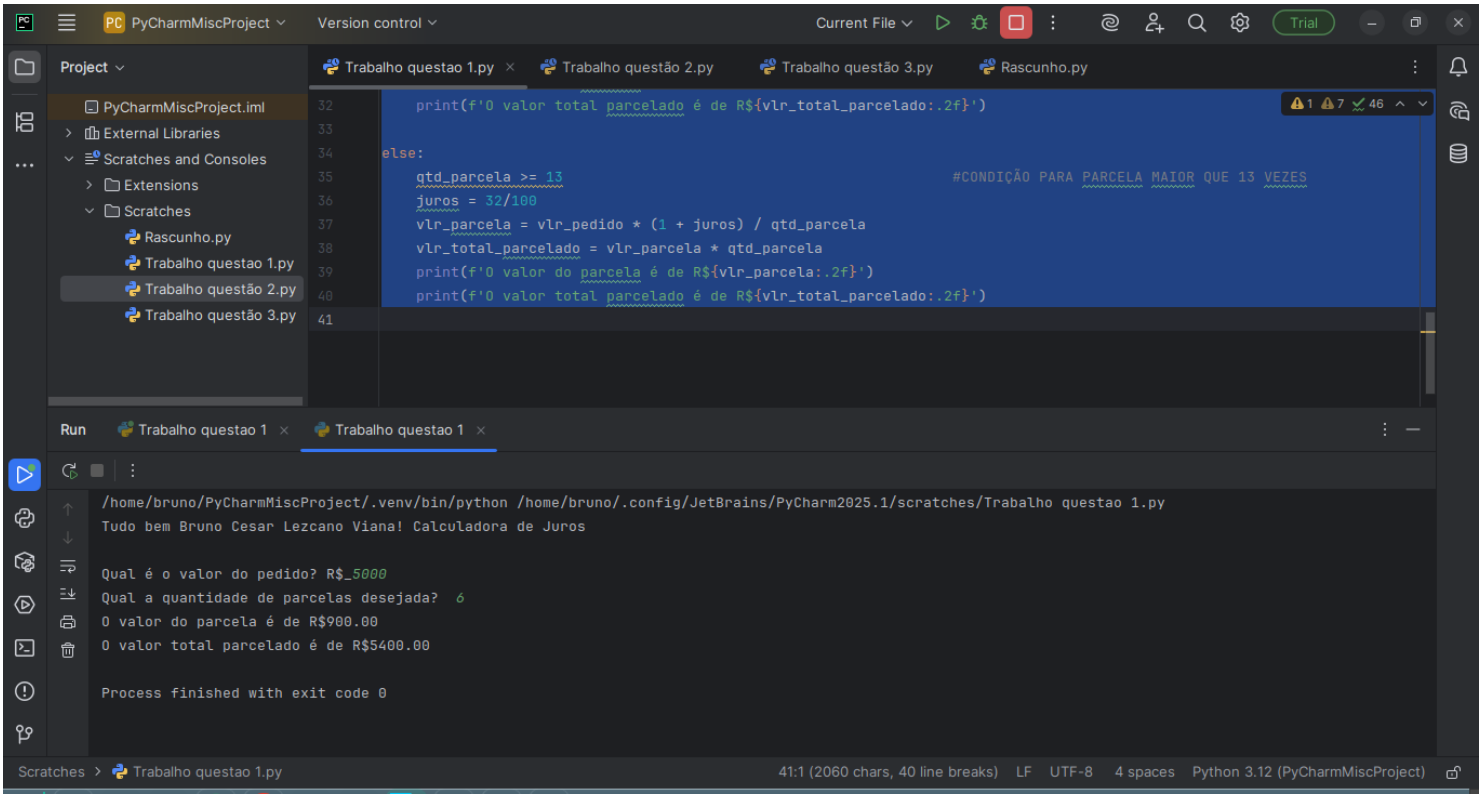
elif (qtd_parcela >= 4 and qtd_parcela < 6):                          #CONDIÇÃO PARA
PARCELA ENTRE 4 A 5 VEZES
    juros = 4/100
    vlr_parcela = vlr_pedido * (1 + juros) / qtd_parcela
    vlr_total_parcelado = vlr_parcela * qtd_parcela
    print(f'O valor do parcela é de R${vlr_parcela:.2f}')
    print(f'O valor total parcelado é de R${vlr_total_parcelado:.2f}')

elif (qtd_parcela >= 6 and qtd_parcela < 9):                          #CONDIÇÃO PARA
PARCELA ENTRE 6 A 8 VEZES
    juros = 8/100
    vlr_parcela = vlr_pedido * (1 + juros) / qtd_parcela
    vlr_total_parcelado = vlr_parcela * qtd_parcela
    print(f'O valor do parcela é de R${vlr_parcela:.2f}')
    print(f'O valor total parcelado é de R${vlr_total_parcelado:.2f}')

elif (qtd_parcela >= 9 and qtd_parcela < 13):                         #CONDIÇÃO PARA
PARCELA ENTRE 9 A 12 VEZES
    juros = 16/100
    vlr_parcela = vlr_pedido * (1 + juros) / qtd_parcela
    vlr_total_parcelado = vlr_parcela * qtd_parcela
    print(f'O valor do parcela é de R${vlr_parcela:.2f}')
    print(f'O valor total parcelado é de R${vlr_total_parcelado:.2f}')

else:
    qtd_parcela >= 13                                                  #CONDIÇÃO PARA
PARCELA MAIOR QUE 13 VEZES
    juros = 32/100
    vlr_parcela = vlr_pedido * (1 + juros) / qtd_parcela
    vlr_total_parcelado = vlr_parcela * qtd_parcela
    print(f'O valor do parcela é de R${vlr_parcela:.2f}')
    print(f'O valor total parcelado é de R${vlr_total_parcelado:.2f}')
```


Apresentação de Saída do Console da Questão 1:



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:

- 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];**
- Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];**
- Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** **[EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];**
- 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:**

```
#INICIO CARDAPIO-----

linha_1 = ("-" * 13 + 'Restaurante Bruno Cesar Lezcano Viana!' + "-" * 13)
print(linha_1)
#print(len(linha_1))
linha_2 = ("-" * 28 + 'Cardápio' + "-" * 28)
print(linha_2)
#print(len(linha_2))
linha_3 = ("-" * 64)
print(linha_3)
#print(len(linha_3))
linha_4 = ('---| Tamanho | Bife Acebolado(BA) | Filé de Frango(FF) |---')
print(linha_4)
#print(len(linha_4))
linha_5 = ('---|      P      |      R$ 16.00      |      R$ 15.00      |---')
print(linha_5)
#print(len(linha_5))
linha_6 = ('---|      M      |      R$ 18.00      |      R$ 17.00      |---')
print(linha_6)
#print(len(linha_6))
linha_7 = ('---|      G      |      R$ 22.00      |      R$ 21.00      |---')
print(linha_7)
#print(len(linha_7))
linha_8 = ("-" * 64)
print(linha_8)
#print(len(linha_8))

#Variaveis-----

pedido_final = 0
sabor_proteina = ''
pergunta_final = ''

sabor = ''
tamanho = ''

#Inicio da programação e do laço-----
while True:
    print()

    #Escolha do sabor
```

```

sabor = input('Entre com o sabor desejado (BA/FF): ')
if (sabor != 'BA' and sabor != 'FF'):
    print('Sabor inválido. Tente novamente')
    print()
    continue # 1º laço

#Escolha do tamanho
tamanho = input('Entre com o tamanho desejado (P/M/G): ')
if (tamanho != 'P' and tamanho != 'M' and tamanho != 'G'):
    print('Tamanho invalido. Tente novamente')
    print()
    continue # 2º laço

#Acumulador e conversão do sabor de sigla para nome
if (sabor == 'BA' and tamanho == 'P'):
    sabor_proteina = 'Bife acebolado'
    pedido_final += 16.00
elif (sabor == 'BA' and tamanho == 'M'):
    sabor_proteina = 'Bife acebolado'
    pedido_final += 18.00
elif (sabor == 'BA' and tamanho == 'G'):
    sabor_proteina = 'Bife acebolado'
    pedido_final += 22.00
elif (sabor == 'FF' and tamanho == 'P'):
    sabor_proteina = 'Filé de frango'
    pedido_final += 15.00
elif (sabor == 'FF' and tamanho == 'M'):
    sabor_proteina = 'Filé de frango'
    pedido_final += 17.00
else:
    (sabor == 'FF' and tamanho == 'G')
    sabor_proteina = 'Filé de frango'
    pedido_final += 21.00

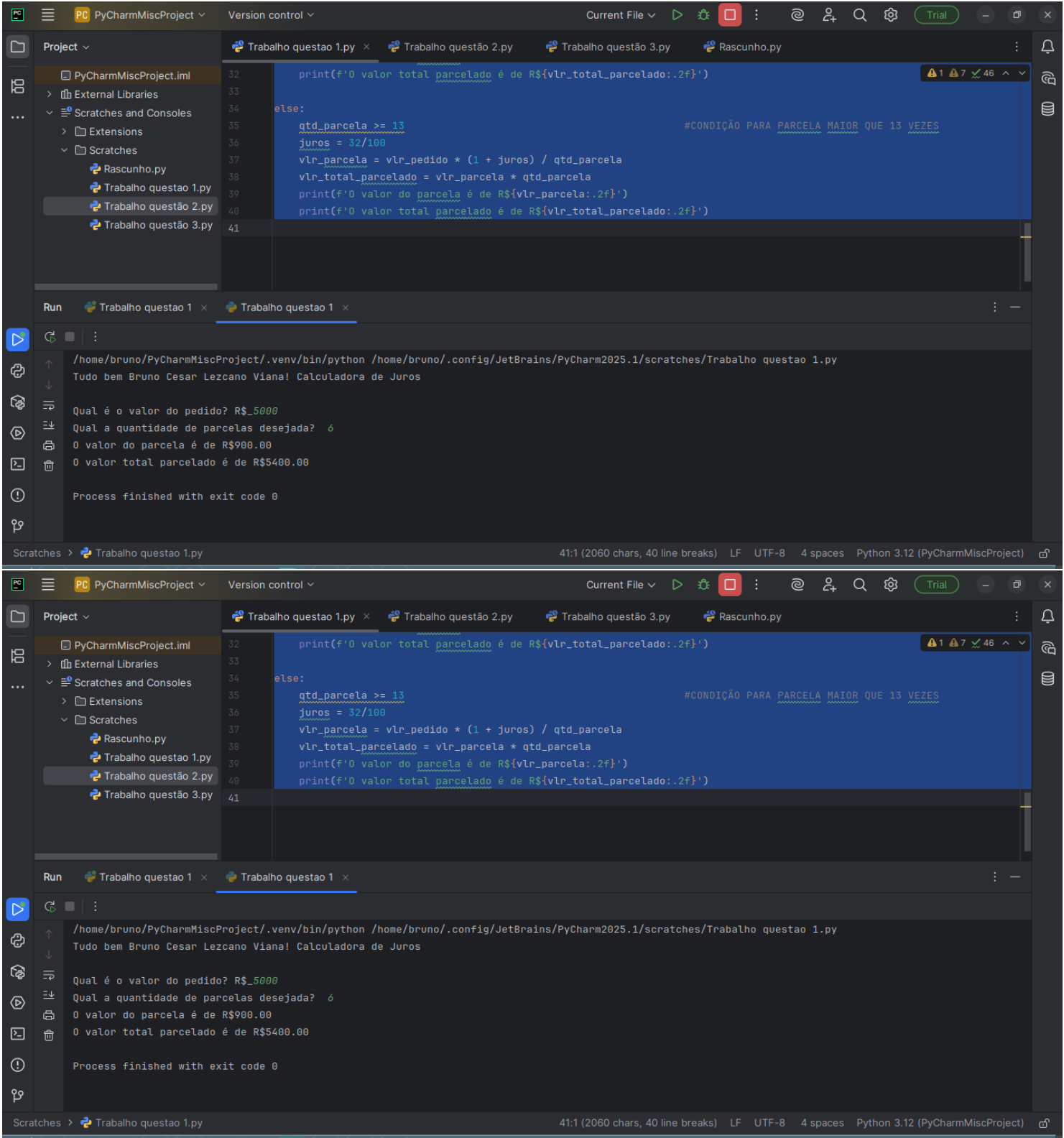
#Pergunta final
pergunta_final = input('Pedido feito. Deseja algo mais? (S/N)')
if (pergunta_final != 'S' and pergunta_final != 'N'):
    print('Escolha invalida. Tente novamente')
    continue # 3º laço
if (pergunta_final == 'S'):
    continue # 4º laço
if (pergunta_final == 'N'):
    break #fim do laço

print()
print(f'O valor total a ser pago: R${pedido_final:.2f}') # mensagem final

```

```
print('fim')
```

Apresentação de Saída do Console da Questão 2:



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:

- 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];
- Deve-se implementar a função **escolha_modelo()** em que: [EXIGÊNCIA DE CÓDIGO 2 de 7];
 - Pergunta o **modelo** desejado;
 - Retorna o valor do modelo** com base na escolha do usuário (use **return**);
 - Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
- Deve-se implementar a função **num_camisetas()** em que: [EXIGÊNCIA DE CÓDIGO 3 de 7];
 - 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
Total: R$ 16040.00 (Modelo: 1.80 * Quantidade(com desconto): 8800 + frete: 200.00) camisetas e frete válidos

```

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:

```
# início do programa -----
print('Estamparia de Bruno Cesar Lezcano Viana!!')
print()

#-----criação da primeira função - escolha da camisa -----

def escolha_modelo():
    while True:                                     # criação do
    laço
        print('Entre com o modelo desejado:')
        print()
        print('Camiseta Manga curta simples - MCS')
        print('Camiseta Manga longa simples - MLS')
        print('Camiseta Manga curta com estampa - MCE')
        print('Camiseta Manga longa com estampa - MLE')
        modelo = input('>>').upper()                # input do modelo
        if (modelo != 'MCS' and modelo != 'MLS' and modelo != 'MCE' and modelo !=
'MLE'):
            print('Escolha inválida, entre como modelo novamente.')
            print()
            print()
            continue
        if (modelo == 'MCS'):                        # associação do
        modelo ao valor
            valor = 1.80
        if (modelo == 'MLS'):                        # associação do
        modelo ao valor
            valor = 2.10
        if (modelo == 'MCE'):                        # associação do
        modelo ao valor
            valor = 2.90
        if (modelo == 'MLE'):                        # associação do
        modelo ao valor
```

```

        valor = 3.20

    break # quebra do laço

print()

return valor, modelo # retorna valor
e modelo

#-----criação da segunda função - escolha da quantidade de camisa
-----

def num_camisetas():

    while True: #criação do laço

        try:

            qtde = int(input('Entre com o numero de camisetas: '))

        except ValueError: # Condição se
for digitado algo diferente de numero

            print('Por favor, entre com um número de camisas novamente.')

            print()

            continue

        if (qtde > 20000): # Condição
para quantidade proibida de camisas

            print('Desculpe, não aceitamos pedidos acima de 20000 unidades')

            continue

        if ( qtde >= 20 and qtde < 200): # Desconto
para a quantidade de camisas

            qtde *= 0.95

        if ( qtde >= 200 and qtde < 2000): # Desconto
para a quantidade de camisas

            qtde *= 0.93

        if ( qtde >= 2000 and qtde < 20000): # Desconto
para a quantidade de camisas

            qtde *= 0.88

        break # quebra
do laço

    print()

    return int(qtde) # retorno
da quantidade de camisas com decontos

```

```

#-----criação da terceira função - escolha do frete -----

def frete():
    while True:
        # criação do laço
        try:
            print('Escolha o tipo de frete:')
            print()
            print('1 - Frete por transportadora - R$ 100.00')
            print('2 - Frete por Sedex - R$ 200.00')
            print('3 - Retirar pedido na fábrica - R$ 0.00')
            escolha = int(input('Entre com o escolha de frete: '))

        except ValueError:
            # Condição se
            for digitado algo diferente de numero
            print('Por favor, entre com um número de frete novamente.')
            print()
            continue

        if (escolha != 1 and escolha != 2 and escolha != 3):
            # Condição se for
            digitado algo diferente do menu
            print('Escolha inválida, entre como modelo novamente.')
            print()
            print()
            continue

        if (escolha == 1):
            # associação do valor do frete
            a modalidade escolhida
            valor_frete = 100.00

        if (escolha == 2):
            # associação do valor do frete
            a modalidade escolhida
            valor_frete = 200.00

        if (escolha == 3):
            # associação do valor do frete
            a modalidade escolhida
            valor_frete = 0.00

        break

```

```
print()

return valor_frete, escolha

# Variáveis para frase final

v_m, m = escolha_modelo() # v_m: valor do modelo e m: modelo escolhido

n = num_camisetas() # n: nº de camisetas com desconto

v_f, e = frete() # v_f: Valor do frete e e: escolha do frete


#print(m)
#print(num_camisetas())
#print(frete())

print(f'Total: R${((v_m + n) + v_f):.2f}. (Modelo:{m} >> {v_m:.2f} * Quantidade  
de camisas (com desconto) >> {n}) + frete: R${v_f:.2f})')
```

Apresentação de Saída do Console da Questão 3:

The screenshot displays the PyCharm IDE interface. At the top, the 'Project' view shows the file structure. The 'Run' toolbar is visible, and the 'Run' configuration dropdown shows 'Trabalho questão 1' and 'Trabalho questão 3'. The main editor window shows the execution of a Python script. The script prompts for a model choice and a quantity. The user enters 'MCS' and '20001', which triggers an error message: 'Desculpe, não aceitamos pedidos acima de 20000 unidades'. The status bar at the bottom shows '98:1 (4473 chars, 102 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject)'.

```
/home/bruno/PyCharmMiscProject/.venv/bin/python /home/bruno/.config/JetBrains/PyCharm2025.1/scratches/Trabalho questão 3.py
Estamparia de Bruno Cesar Lezcano Viana!!

Entre com o modelo desejado:

Camiseta Manga curta simples - MCS
Camiseta Manga longa simples - MLS
Camiseta Manga curta com estampa - MCE
Camiseta Manga longa com estampa - MLE
>>Q
Escolha inválida, entre como modelo novamente.

Entre com o modelo desejado:

Camiseta Manga curta simples - MCS
Camiseta Manga longa simples - MLS
Camiseta Manga curta com estampa - MCE
Camiseta Manga longa com estampa - MLE
>>MCS

Entre com o numero de camisetas: 20001
Desculpe, não aceitamos pedidos acima de 20000 unidades
Entre com o numero de camisetas: 5000
```

The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for Project, Run, and other development tools. The main editor window shows a Python script being executed. The script prompts the user for a shirt model, quantity, and shipping type, then calculates the total price. The output shows a total of R\$4601.80 for 20001 shirts with shipping.

```
Entre com o modelo desejado:
Camiseta Manga curta simples - MCS
Camiseta Manga longa simples - MLS
Camiseta Manga curta com estampa - MCE
Camiseta Manga longa com estampa - MLE
>>MCS

Entre com o numero de camisetas: 20001
Desculpe, não aceitamos pedidos acima de 20000 unidades
Entre com o numero de camisetas: 5000

Escolha o tipo de frete:
1 - Frete por transportadora - R$ 100.00
2 - Frete por Sedex - R$ 200.00
3 - Retirar pedido na fábrica - R$ 0.00
Entre com o escolha de frete: 2

Total: R$4601.80. (Modelo:MCS >> 1.80 * Quantidade de camisas (com desconto) >> 4400) + frete: R$200.00

Process finished with exit code 0
```

The bottom status bar shows the file path: Scratches > Trabalho questão 3.py, and the editor settings: 98:1 (4473 chars, 102 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject).

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];

- a. Deve-se pergunta pelo **id** do funcionário a ser removido;
 - b. Remover o funcionário da **lista_funcionarios**;
 - c. 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]**;
- a. Deve-se pergunta qual opção deseja (1. Cadastrar Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
 - i. Se Cadastrar Funcionário, chamar a função **cadastrar_funcionario(id_global)** e **em seguida, incrementar** em um **id_global** (no menu principal);
 - ii. Se Consultar Funcionário, chamar função **consultar_funcionario()**;
 - iii. Se Remover Funcionário, chamar função **remover_funcionario()**;
 - iv. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
 - v. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar "Opção inválida" e repetir a pergunta **F.a.**
 - vi. 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 Kostiuk      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('Empresas Bruno Cesar Lezcano viana Ltda')
print()

#variaveis

lista_funcionarios = []
cadastro = {}
id_global = 4512355

#Função criada para cadastrar funcionários

def cadastrar_funcionario(id):
    print(' ' * 10 + 'cadastro de funcionário ')
    for i in range(1):
        print(f'ID do funcionário: {id}')
        cadastro['ID'] = id
        cadastro['Nome'] = input('Por favor entre com o nome do funcionário:')
        cadastro['Setor'] = input('Por favor entre com o setor do funcionário:')
        while True:
            try:
                cadastro['Salário'] = float(input('Por favor entre com o salário do funcionário:'))
                break
            except ValueError:
                print('tente novamente')
        lista_funcionarios.append(cadastro.copy())

#FUNÇÃO PARA CONSULTAR FUNCIONÁRIOS
```

```

def consultar_funcionarios():
    while True:
        print(' ' * 10 + 'Consultar Funcionários')
        print("1 - Consultar Todos os Funcionários")
        print("2 - Consultar Funcionários por Id")
        print("3 - Consultar Funcionários por Setor")
        print("4 - Retornar")
        a = input(">> ")

        if a == '1':
            for funcionario in lista_funcionarios:
                print(f"ID: {funcionario['ID']}")
                print(f"Nome: {funcionario['Nome']}")
                print(f"Setor: {funcionario['Setor']}")
                print(f"Salário: {funcionario['Salário']}")
                print()

        elif a == '2':

            id_busca = int(input("Digite o ID do funcionário: "))
            for funcionario in lista_funcionarios:
                if funcionario['ID'] == id_busca:
                    print(f"ID: {funcionario['ID']}")
                    print(f"Nome: {funcionario['Nome']}")
                    print(f"Setor: {funcionario['Setor']}")
                    print(f"Salário: {funcionario['Salário']}")
                    print()

        elif a == '3':
            setor_busca = input("Digite o setor: ")
            for funcionario in lista_funcionarios:
                if funcionario['Setor'] == setor_busca:
                    print(f"ID: {funcionario['ID']}")

```

```

        print(f"Nome: {funcionario['Nome']}")
        print(f"Setor: {funcionario['Setor']}")
        print(f"Salário: {funcionario['Salário']}")
        print()

    elif a == '4':
        return # Volta ao menu principal

    else:
        print("Opção inválida. Tente novamente.")
        print()

#FUNÇÃO PARA REMOVER FUNCIONÁRIOS

def remover_funcionario():
    while True:
        print(' ' * 10 + "Remover Funcionário")
        try:
            id_remover = int(input("Digite o ID do funcionário a ser removido:
"))
        except ValueError:
            print("Digite um número válido.")
            continue

    for funcionario in lista_funcionarios:
        if funcionario['ID'] == id_remover:
            lista_funcionarios.remove(funcionario)
            print("Funcionário removido com sucesso.")
            return

    print("Id inválido. Tente novamente.")

```

```
# MENU PRINCIPAL -----

while True:

    print()
    print(' ' * 20 + 'Menu Principal')
    print("1 - Cadastrar Funcionário")
    print("2 - Consultar Funcionário(s)")
    print("3 - Remover Funcionário")
    print("4 - sair")
    a = int(input(">> "))

    if a == 1:                                     #condição de cadastro funcionário
        cadastrar_funcionario(id_global)
        id_global += 1
        print('Cadastro efetuado com sucesso!')

    elif a == 2:                                   #condição de consulta funcionário
        consultar_funcionarios()

    elif a == 3:                                   #condição de remover funcionário
        remover_funcionario()

    elif a == 4:
        print("Encerrando programa. Até logo!")
        break

    else:
        print("Opção inválida. Tente novamente.")
```


Apresentação de Saída do Console da Questão 4:

```
Project ▾
Trabalho questao 1.py Trabalho questão 2.py Trabalho questão 3.py Trabalho questão 4.py x Rascunho.py
PyCharmMiscProject.iml 128 print("Encerrando programa. Até logo!") 1 2 82
Run Trabalho questao 1 x Rascunho x Trabalho questão 4 x
/home/bruno/PyCharmMiscProject/.venv/bin/python /home/bruno/.config/JetBrains/PyCharm2025.1/scratches/Trabalho questão 4.py
Empresas Bruno Cesar Lezcano viana Ltda

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 1

cadastro de funcionário
ID do funcionário: 4512355
Por favor entre com o nome do funcionário: Bruno Cesar Lezcano Viana
Por favor entre com o setor do funcionário: Administração
Por favor entre com o salário do funcionário: 15000
Cadastro efetuado com sucesso!

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 1
```

```
Project ▾
Trabalho questao 1.py Trabalho questão 2.py Trabalho questão 3.py Trabalho questão 4.py x Rascunho.py
PyCharmMiscProject.iml 128 print("Encerrando programa. Até logo!") 1 2 82
Run Trabalho questao 1 x Rascunho x Trabalho questão 4 x
/home/bruno/PyCharmMiscProject/.venv/bin/python /home/bruno/.config/JetBrains/PyCharm2025.1/scratches/Trabalho questão 4.py
Empresas Bruno Cesar Lezcano viana Ltda

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 1

cadastro de funcionário
ID do funcionário: 4512356
Por favor entre com o nome do funcionário: Rosana Bulhões
Por favor entre com o setor do funcionário: Logística
Por favor entre com o salário do funcionário: 20000
Cadastro efetuado com sucesso!

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 1

cadastro de funcionário
ID do funcionário: 4512357
Por favor entre com o nome do funcionário: Queiroz galvão
Por favor entre com o setor do funcionário: Logística
Por favor entre com o salário do funcionário: 25000
Cadastro efetuado com sucesso!

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 1
```

Scratches > Trabalho questão 4.py 126:1 (3920 chars, 125 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject)

```
Project ▾
Trabalho questao 1.py Trabalho questão 2.py Trabalho questão 3.py Trabalho questão 4.py x Rascunho.py
PyCharmMiscProject.iml 120 print("Encerrando programa. Até logo!")
Run Trabalho questao 1 x Rascunho x Trabalho questão 4 x
>> 2
Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 1
ID: 4512355
Nome: Bruno Cesar Lezcano Viana
Setor: Administração
Salário: 15000.0

ID: 4512356
Nome: Rosana Bulhões
Setor: Logística
Salário: 20000.0

ID: 4512357
Nome: Queiroz galvão
Setor: Logística
Salário: 25000.0

Consultar Funcionários
Scratches > Trabalho questão 4.py 126:1 (3920 chars, 125 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject)
```

```
Project ▾
Trabalho questao 1.py Trabalho questão 2.py Trabalho questão 3.py Trabalho questão 4.py x Rascunho.py
PyCharmMiscProject.iml 120 print("Encerrando programa. Até logo!")
Run Trabalho questao 1 x Rascunho x Trabalho questão 4 x
>> 2
Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 2
Digite o ID do funcionário: 4512355
ID: 4512355
Nome: Bruno Cesar Lezcano Viana
Setor: Administração
Salário: 15000.0
```

```
Salário: 15000.0

Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 3
Digite o setor: Logística
ID: 4512356
Nome: Rosana Bulhões
Setor: Logística
Salário: 20000.0

ID: 4512357
Nome: Queiroz galvão
Setor: Logística
Salário: 25000.0

Consultar Funcionários
```

```
Setor: Logística
Salário: 25000.0

    Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 4

    Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 3

    Remover Funcionário
Digite o ID do funcionário a ser removido: 4512355
Funcionário removido com sucesso.

    Menu Principal
```

```
print("Encerrando programa. Até logo!")
```

```
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 2

    Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 1
ID: 4512356
Nome: Rosana Bulhões
Setor: Logística
Salário: 20000.0

ID: 4512357
Nome: Queiroz galvão
Setor: Logística
Salário: 25000.0

    Consultar Funcionários
1 - Consultar Todos os Funcionários
```

Scratches > Trabalho questão 4.py 126:1 (3920 chars, 125 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject)

```
Project
Trabalho questao 1.py
Trabalho questão 2.py
Trabalho questão 3.py
Trabalho questão 4.py
Rascunho.py

PyCharmMiscProject.iml
120
print("Encerrando programa. Até logo!")

Run
Trabalho questao 1
Rascunho
Trabalho questão 4

ID: 4512357
Nome: Queiroz galvão
Setor: Logística
Salário: 25000.0

Consultar Funcionários
1 - Consultar Todos os Funcionários
2 - Consultar Funcionários por Id
3 - Consultar Funcionários por Setor
4 - Retornar
>> 4

Menu Principal
1 - Cadastrar Funcionário
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - sair
>> 4

Encerrando programa. Até logo!

Process finished with exit code 0

Scratches > Trabalho questão 4.py
126:1 (3920 chars, 125 line breaks) LF UTF-8 4 spaces Python 3.12 (PyCharmMiscProject)
```

CRITÉRIOS DE AVALIAÇÃO

SOMENTE INFORMATIVO. PODE APAGAR AO ENTREGAR O TRABALHO.

A avaliação se dará através de **critérios estritamente objetivos**:

- Dentro de cada questão, metade da nota se refere as exigências de implementação do código (EC), e a outra metade as exigências de saída (ES) de funcionamento do código:

$$NOTA_{QUESTÃO} = \frac{NOTA_{EC} + NOTA_{ES}}{2}$$

- Cada questão vale um quarto da nota da atividade. Assim, calculamos a sua nota final na atividade prática com a fórmula abaixo:

$$NOTA_{AP} = \frac{NOTA_{QUESTÃO_1} + NOTA_{QUESTÃO_2} + NOTA_{QUESTÃO_3} + NOTA_{QUESTÃO_4}}{4}$$

Os critérios e suas respectivas pontuações estão na tabela a seguir:

QUESTÃO 1		
Exigências de Código (EC)		Pts
EC 1	Implementar corretamente a mensagem de Boas-vindas com nome e sobrenome	30
EC 2	Implementar corretamente o input do valor do pedido e quantidade parcelas	10
EC 3	Implementar corretamente a regra de juros conforme o enunciado	15
EC 4	Implementar corretamente o valor da parcela e/ou valor total parcelado	15
EC 5	Implementar corretamente o if, elif e else	20
EC 6	Inserção de comentários no código	10
Total		100
Exigências de Saída (ES)		Pts
ES 1	Apresentar o print de Boas-vindas com nome e sobrenome	30
ES 2	Apresentar um parcelamento com juros	35
ES 2	Apresentar o valor Total Parcelado e/ou valor da Parcela	35
Total		100

<u>QUESTÃO 2</u>		
Exigências de Código (EC)		Pts
EC 1	Implementar corretamente a mensagem de Boas-vindas com nome e sobrenome	15
EC 1	Implementar corretamente no código o Menu para o cliente	15
EC 2	Implementar corretamente o input do sabor e print de sabor invalido	10
EC 3	Implementar corretamente o input do tamanho e print de tamanho invalido	10
EC 4	Implementar corretamente as regras para combinação de sabores e tamanhos de forma aninhada	10
EC 5	Implementar corretamente o acumulador com a soma dos valores	10
EC 6	Implementar corretamente a condição de "Deseja pedir mais alguma coisa?"	10
EC 7	Implementar corretamente o while, continue e break	10
EC 8	Inserção de comentários no código	10
Total		100
Exigências de Saída (ES)		Pts
ES 1	Apresentar o print de Boas-vindas com nome e sobrenome	15
ES 1	Apresentar o print com o menu de opções	15
ES 2	Apresentar um pedido em que o usuário errou ao digitar o sabor	20
ES 3	Apresentar um pedido em que o usuário errou ao digitar o tamanho	20
ES 4	Apresentar um pedido com dois sabores e dois tamanhos diferentes	30
Total		100

<u>QUESTÃO 3</u>		
Exigências de Código (EC)		Pts
EC 1	Implementar corretamente a mensagem de Boas-vindas com nome e sobrenome	30
EC 2	Implementar corretamente a função escolha_modelo()	10
EC 3	Implementar corretamente a função num_camisetas()	20
EC 4	Implementar corretamente a função frete()	10
EC 5	Implementar corretamente o total a pagar conforme a regra do enunciado	10
EC 6	Implementar corretamente a cláusula try/except	10
EC 7	Inserção de comentários no código	10
Total		100
Exigências de Saída (ES)		Pts
ES 1	Apresentar o print de Boas-vindas com nome e sobrenome	30
ES 2	Apresentar um pedido em que o usuário errou a opção de modelo	20
ES 3	Apresentar um pedido em que o usuário ultrapassou o número de camisetas	20
ES 4	Apresentar um pedido válido.	30
Total		100

<u>QUESTÃO 4</u>		
Exigências de Código (EC)		Pts
EC 1	Implementar corretamente a mensagem de Boas-vindas com nome e sobrenome	30
EC 2	Implementar corretamente lista de funcionarios e id_global	5
EC 3	Implementar corretamente a função cadastrar_funcionario (id)	10
EC 4	Implementar corretamente a função consultar_funcionario() opção 1	5
EC 4	Implementar corretamente a função consultar_funcionario() opção 2	5
EC 4	Implementar corretamente a função consultar_funcionario() opção 3	5
EC 4	Implementar corretamente a função consultar_funcionario() opção 4	5
EC 5	Implementar corretamente a função remover_funcionario()	10
EC 6	Implementar corretamente o menu principal conforme enunciado	10
EC 7	Implementar corretamente uma lista contendo dicionários	5
EC 8	Inserção de comentários no código	10
Total		100
Exigências de Saída (ES)		Pts
ES 1	Apresentar o print de Boas-vindas com nome e sobrenome	30
ES 2	Apresentar o cadastro de 3 funcionários (com 2 deles com mesmo setor)	15
ES 3	Apresentar uma consulta de todos os funcionários	15
ES 4	Apresentar uma consulta por código	15
ES 5	Apresentar uma consulta por setor com 2 funcionários do mesmo setor	15
ES 6	Apresentar uma remoção seguida de uma consulta de todos os funcionários	10
Total		100