# QUESTÃO 1 de 4 - Conteúdo até Aula 03

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

O valor da parcela é calculado da seguinte maneira:

𝐯𝐚𝐥𝐨𝐫𝐃𝐚𝐏𝐚𝐫𝐜𝐞𝐥𝐚 =

O valor total parcelado é calculado da seguinte maneira:

𝐯𝐚𝐥𝐨𝐫𝐃𝐨𝐏𝐞𝐝𝐢𝐝𝐨 ∗ (𝟏 + 𝐣𝐮𝐫𝐨𝐬)

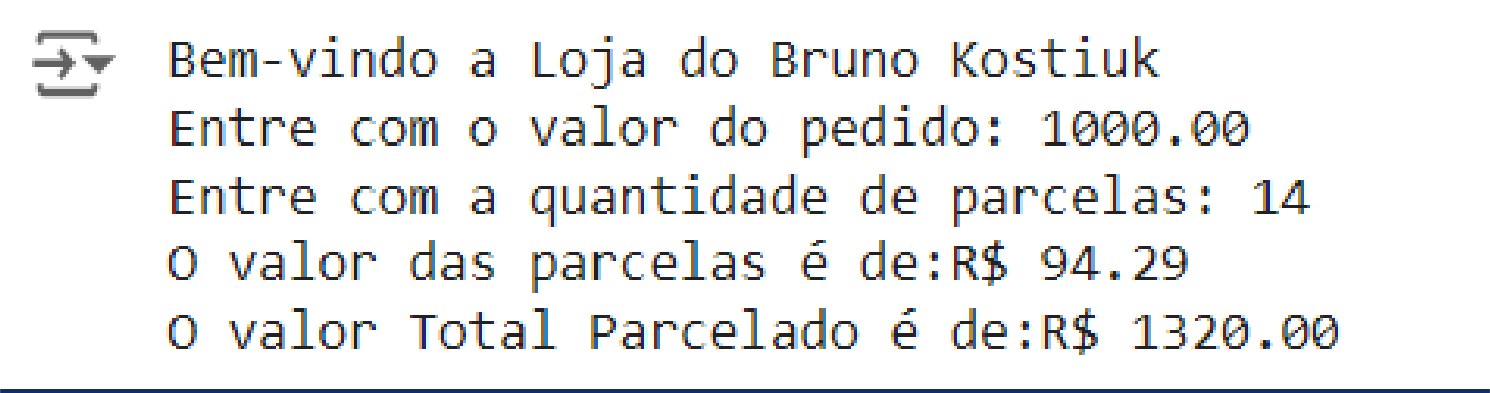
𝐪𝐮𝐚𝐧𝐭𝐢𝐝𝐚𝐝𝐞𝐏𝐚𝐫𝐜𝐞𝐥𝐚𝐬

𝐯𝐚𝐥𝐨𝐫𝐓𝐨𝐭𝐚𝐥𝐏𝐚𝐫𝐜𝐞𝐥𝐚𝐝𝐨 = 𝐯𝐚𝐥𝐨𝐫𝐃𝐚𝐏𝐚𝐫𝐜𝐞𝐥𝐚 ∗ 𝐪𝐮𝐚𝐧𝐭𝐢𝐝𝐚𝐝𝐞𝐏𝐚𝐫𝐜𝐞𝐥𝐚𝐬

Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui). Por exemplo: **print(“Bem-vindos a loja do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 6];
2. Deve-se implementar o input do **valorDoPedido** e da **quantidadeParcelas** [EXIGÊNCIA DE CÓDIGO 2 de 6];
3. 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];
4. Deve-se implementar o **valorDaParcela** e **valorTotalParcelado** [EXIGÊNCIA DE CÓDIGO 4 de 6];
5. Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
6. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
7. Deve-se apresentar na saída de console uma mensagem com seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];
8. 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:



**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("Bem vindo a loja do Matheus de Abreu de França")

valorDoPedido = float(input("Entre com o valor do pedido: "))

quantidadeParcelas = int(input("Entre com a quantidade de parcelas: "))

if quantidadeParcelas < 4:

juros = 0

elif 4 <= quantidadeParcelas < 6:

juros = 0.04

elif 6 <= quantidadeParcelas < 9:

juros = 0.08

elif 9 <= quantidadeParcelas < 13:

juros = 0.16

else:

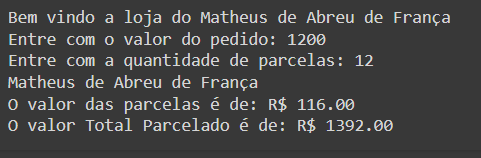
juros = 0.32

valorDaParcela = (valorDoPedido \* (1 + juros)) / quantidadeParcelas

valorTotalParcelado = valorDaParcela \* quantidadeParcelas

print(f"Matheus de Abreu de França")

*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:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem vindos a loja de Marmitas do Bruno Kostiuk”)**

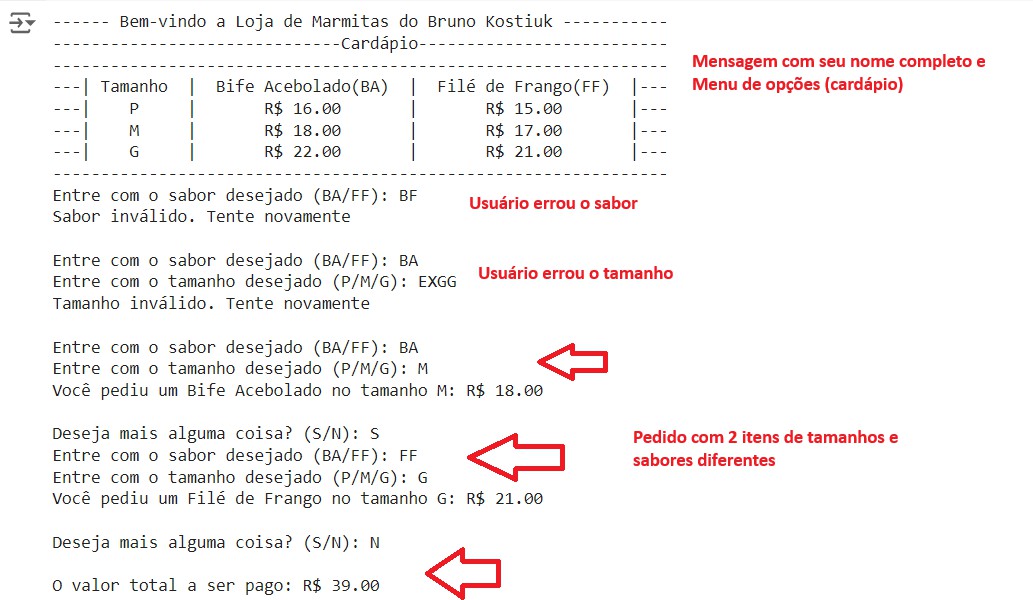
Além do seu nome completo, deve-se implementar um **print com um Menu** para o cliente. [EXIGÊNCIA DE CÓDIGO 1 de 8];

1. 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];
2. 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];
3. 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];

1. Deve-se implementar um **acumulador** para somar os valores dos pedidos [EXIGÊNCIA DE CÓDIGO 5 de 8];
2. 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];
3. Deve-se implementar as estruturas de **while, break, continue (todas elas)** [EXIGÊNCIA DE CÓDIGO 7 de 8];
4. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
5. 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];
6. Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
7. Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
8. 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:



**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("------ Bem-vindo à Loja de Marmitas do Matheus de Abreu de França ")

print(" Cardápio ") print(" ")

print("---| Tamanho | Bife Acebolado(BA) | Filé de Frango(FF) | ")

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| print("---| | P | | | R$ 16.00 | | | R$ 15.00 | |---") |
| print("---| | M | | | R$ 18.00 | | | R$ 17.00 | |---") |
| print("---| print(" | G | | | R$ 22.00 | | | R$ 21.00 | |---")  ") |
| total = 0 |  |  |  |  |  |  |
| while True: |  |  |  |  |  |  |

sabor = input("Entre com o sabor desejado (BA/FF): ").strip().upper() if sabor not in ["BA", "FF"]:

print("Sabor inválido. Tente novamente") continue

# Input para o tamanho

tamanho = input("Entre com o tamanho desejado (P/M/G): ").strip().upper() if tamanho not in ["P", "M", "G"]:

print("Tamanho inválido. Tente novamente") continue

if sabor == "BA":

if tamanho == "P": preco = 16.00

elif tamanho == "M": preco = 18.00

elif tamanho == "G": preco = 22.00

elif sabor == "FF":

if tamanho == "P": preco = 15.00

elif tamanho == "M": preco = 17.00

elif tamanho == "G": preco = 21.00

# Adicionar o preço ao acumulador total += preco

# Exibir o pedido atual

print(f"Você pediu um {sabor} no tamanho {tamanho}: R$ {preco:.2f}")

# Perguntar se o usuário deseja pedir mais alguma coisa

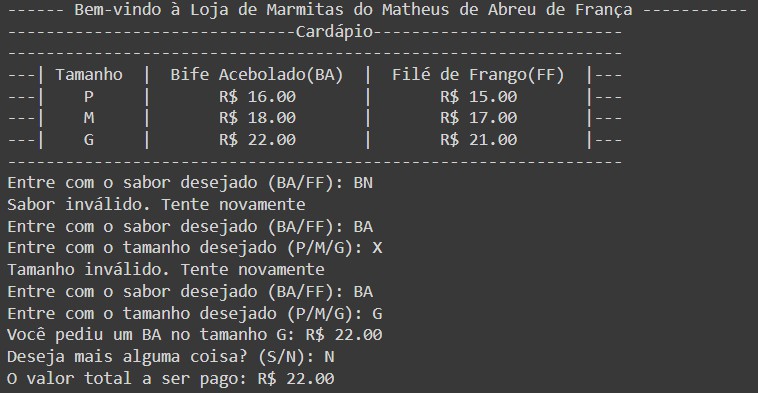
mais\_alguma\_coisa = input("Deseja mais alguma coisa? (S/N): ").strip().upper() if mais\_alguma\_coisa == "N":

break

# Exibir o valor total acumulado

print(f"O valor total a ser pago: R$ {total:.2f}")

*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:

total = (**modelo \* num\_camisetas) + frete**

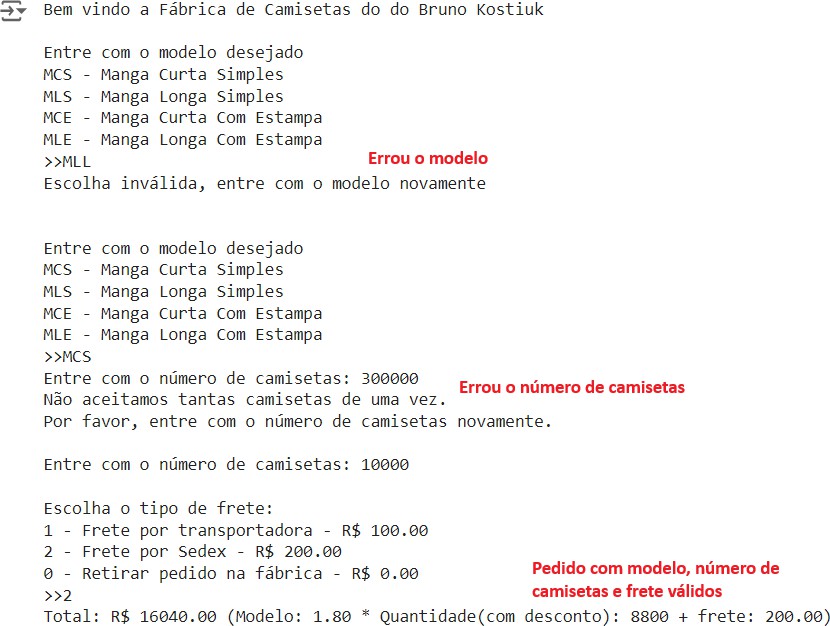
Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem vindos a Fábrica de Camisetas do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 7];

1. Deve-se implementar a função **escolha\_modelo()** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Pergunta o **modelo** desejado;
   2. **Retorna o valor do modelo** com base na escolha do usuário (use **return**);
   3. Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
2. Deve-se implementar a função **num\_camisetas()** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Pergunta o **número de camisetas**;
   2. **Retorna** (use **return**) o **número de camisetas** com desconto seguindo a regra do enunciado (desconto calculado em cima do número de camisetas);
   3. Repete a pergunta do item **C.a** se digitar um valor acima de 20000 ou valor não numérico (use try/except para não numérico)
3. Deve-se implementar a função **frete()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Pergunta pelo serviço **adicional de frete**;
   2. **Retorna** (use **return**) o **valor** de apenas uma das **opções** de **frete**
   3. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/0;
4. 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];
5. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
6. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
7. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
8. 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];
9. 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];
10. 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:



**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 vindos a Fábrica de Camisetas do Matheus de Abreu de França")*

*def escolha\_modelo(): while True:*

*print("\nEntre com o modelo desejado") print("MCS - Manga Curta Simples") print("MLS - Manga Longa Simples") print("MCE - Manga Curta Com Estampa") print("MLE - Manga Longa Com Estampa") modelo = input(">> ").strip().upper()*

*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("Escolha inválida, entre com o modelo novamente")*

*def num\_camisetas(): while True:*

*try:*

*quantidade = int(input("Entre com o número de camisetas: ").strip()) if quantidade > 20000:*

*print("Não aceitamos tantas camisetas de uma vez.") continue*

*if quantidade < 20: return quantidade*

*elif 20 <= quantidade < 200: return quantidade \* 0.95*

*elif 200 <= quantidade < 2000: return quantidade \* 0.93*

*elif 2000 <= quantidade <= 20000: return quantidade \* 0.88*

*except ValueError:*

*print("Valor inválido. Por favor, entre com um número válido de camisetas.")*

*def frete(): while True:*

*print("\nEscolha o tipo de frete:")*

*print("1 - Frete por transportadora - R$ 100.00") print("2 - Frete por Sedex - R$ 200.00")*

*print("0 - Retirar pedido na fábrica - R$ 0.00") opcao\_frete = input(">> ").strip()*

*if opcao\_frete == "1": return 100*

*elif opcao\_frete == "2": return 200*

*elif opcao\_frete == "0": return 0*

*else:*

*print("Opção de frete inválida. Tente novamente.")*

*try:*

*valor\_modelo = escolha\_modelo() quantidade\_com\_desconto = num\_camisetas() valor\_frete = frete()*

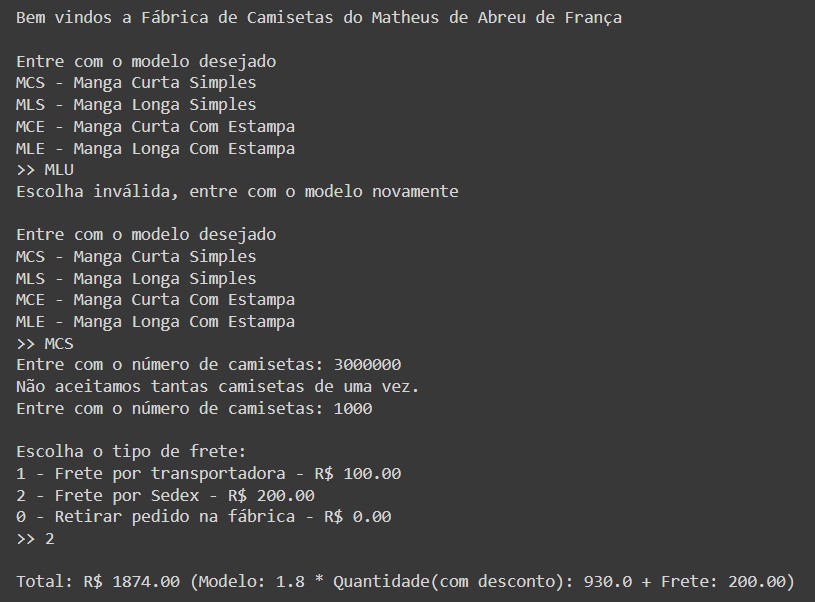
*total = (valor\_modelo \* quantidade\_com\_desconto) + valor\_frete*

*# Exibe o valor total a ser pago*

*print(f"\nTotal: R$ {total:.2f} (Modelo: {valor\_modelo} \* Quantidade(com desconto): {quantidade\_com\_desconto} + Frete: {valor\_frete:.2f})") except Exception as e:*

*print(f"Um erro ocorreu: {e}")*

*Apresentação de Saída do Console da Questão 3:*



# 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

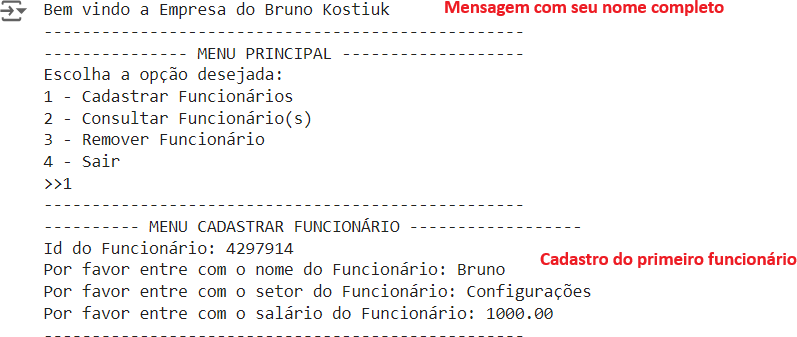
Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

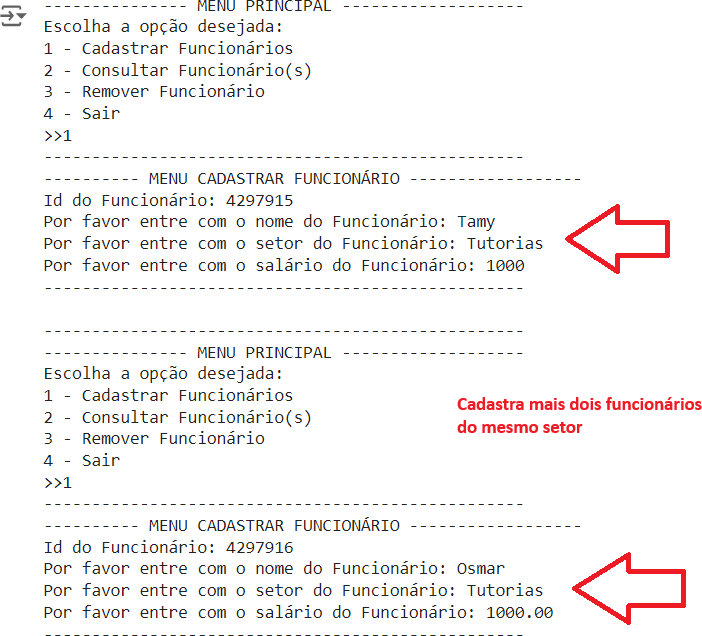
Por exemplo: **print(“Bem vindos a empresa do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 8];

1. 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];
2. Deve-se implementar uma função chamada **cadastrar\_funcionario(id)** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 8];
   1. Pergunta **nome**, **setor**, **salario** do funcionário;
   2. Armazena o **id** (este é fornecido via parâmetro da função), **nome**, **setor**, **salario** dentro de um dicionário;
   3. **Copiar** o dicionário para dentro da **lista\_funcionarios** (utilizar o **copy**);
3. Deve-se implementar uma função chamada **consultar\_funcionarios()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 8];
   1. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
      1. Se Consultar Todos, apresentar todos os funcionários com todos os seus dados cadastrados;
      2. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o funcionário específico com todos os seus dados cadastrados;
      3. 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;
      4. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **D.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu consultar funcionários deve se repetir.
4. Deve-se implementar uma função chamada **remover\_funcionario()** em que**:** [EXIGÊNCIA DE CÓDIGO 5 de 8];
   1. Deve-se pergunta pelo **id** do funcionário a ser removido;
   2. Remover o funcionário da **lista\_funcionarios**;
   3. Se o id fornecido não for de um funcionário da lista, printar “Id inválido” e repetir a pergunta **E.a**.
5. 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];
   1. Deve-se pergunta qual opção deseja (1. Cadastrar Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
      1. Se Cadastrar Funcionário, **incrementar** em um **id\_ global** e chamar a função **cadastrar\_funcionario(id\_ global)**;
      2. Se Consultar Funcionário, chamar função **consultar\_funcionario ()**;
      3. Se Remover Funcionário, chamar função **remover\_funcionario()**;
      4. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **F.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
6. Deve-se implementar uma **lista de dicionários** (uma lista contento dicionários dentro) [EXIGÊNCIA DE CÓDIGO 7 de 8];
7. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
8. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6];
9. 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];
10. Deve-se apresentar na saída de console uma consulta de todos os funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6];
11. Deve-se apresentar na saída de console uma consulta por código (id) de um dos funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6];
12. 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];
13. 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];

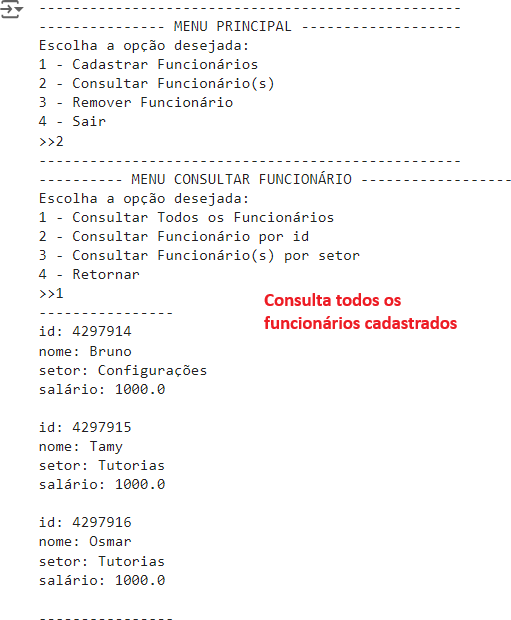
## EXEMPLO DE SAÍDA DE CONSOLE:



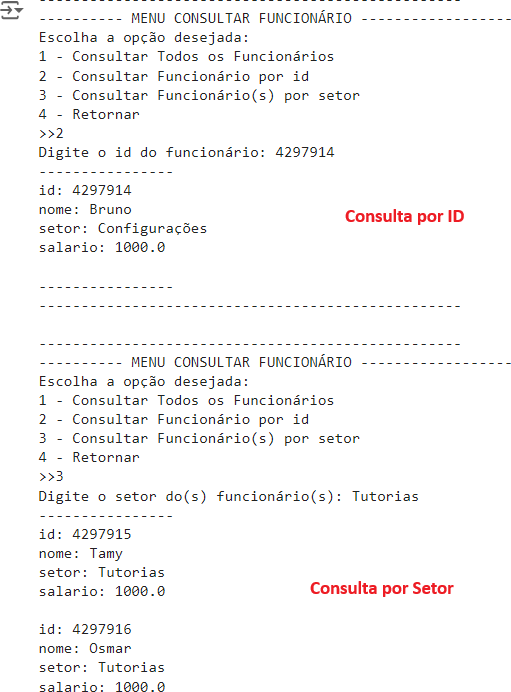
**Figura 4.1: Exemplo de saída de console que o aluno deve fazer. Apresenta o print com seu nome completo e é realizado o cadastro do primeiro funcionário, note que o ID do funcionário inicia em 4297914 pois o id\_global é inicializado em 4297913.**



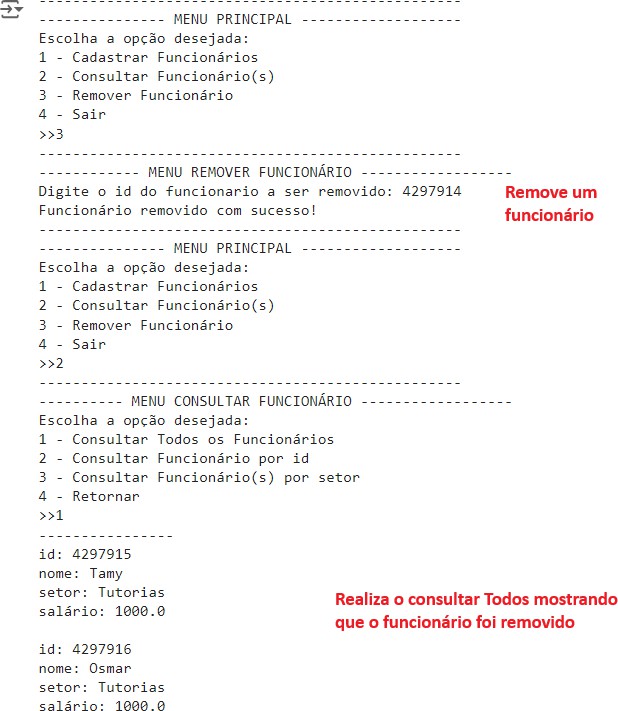
**Figura 4.2: Exemplo de saída de console que o aluno deve fazer. São cadastrados mais dois funcionários com mesmo setor.**



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



**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).**



**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 vindos a empresa do Matheus de Abreu de França")**

**lista\_funcionarios = [] id\_global = 4873462**

**def cadastrar\_funcionario(id): print(f"Id do Funcionário: {id}")**

**nome = input("Por favor entre com o nome do Funcionário: ").strip() setor = input("Por favor entre com o setor do Funcionário: ").strip()**

**salario = float(input("Por favor entre com o salário do Funcionário: ").strip()) funcionario = {**

**'id': id,**

**'nome': nome, 'setor': setor, 'salario': salario**

**}**

**lista\_funcionarios.append(funcionario.copy()) print("Funcionário cadastrado com sucesso!")**

**# Função para consultar funcionários def consultar\_funcionarios():**

**while True:**

**print("\n---------- MENU CONSULTAR FUNCIONÁRIO ")**

**print("1. Consultar Todos") print("2. Consultar por Id") print("3. Consultar por Setor") print("4. Retornar ao menu")**

**opcao = input("Escolha a opção desejada: ").strip()**

**if opcao == "1":**

**print("\n--- Todos os Funcionários ")**

**for func in lista\_funcionarios:**

**print(f"id: {func['id']}, nome: {func['nome']}, setor: {func['setor']}, salário: {func['salario']}")**

**print(" ")**

**elif opcao == "2":**

**id\_consulta = int(input("Digite o id do funcionário: ").strip()) encontrado = False**

**for func in lista\_funcionarios: if func['id'] == id\_consulta:**

**print(f"\nid: {func['id']}, nome: {func['nome']}, setor: {func['setor']}, salário: {func['salario']}") encontrado = True**

**break**

**if not encontrado:**

**print("Funcionário não encontrado.")**

**elif opcao == "3":**

**setor\_consulta = input("Digite o setor do(s) funcionário(s): ").strip() encontrado = False**

**print(f"\n--- Funcionários do setor {setor\_consulta} ")**

**for func in lista\_funcionarios:**

**if func['setor'] == setor\_consulta:**

**print(f"id: {func['id']}, nome: {func['nome']}, setor: {func['setor']}, salário: {func['salario']}") encontrado = True**

**if not encontrado:**

**print("Nenhum funcionário encontrado nesse setor.") print(" ")**

**elif opcao == "4": return**

**else:**

**print("Opção inválida, tente novamente.")**

**# Função para remover funcionário def remover\_funcionario():**

**while True:**

**id\_remocao = int(input("Digite o id do funcionário a ser removido: ").strip()) encontrado = False**

**for func in lista\_funcionarios: if func['id'] == id\_remocao:**

**lista\_funcionarios.remove(func) print("Funcionário removido com sucesso!") encontrado = True**

**return**

**if not encontrado:**

**print("Id inválido, tente novamente.")**

**# Estrutura de menu principal while True:**

**print("\n--------------- MENU PRINCIPAL ")**

**print("1. Cadastrar Funcionário") print("2. Consultar Funcionário") print("3. Remover Funcionário") print("4. Encerrar Programa")**

**opcao\_principal = input("Escolha a opção desejada: ").strip()**

**if opcao\_principal == "1": id\_global += 1 cadastrar\_funcionario(id\_global)**

**elif opcao\_principal == "2": consultar\_funcionarios()**

**elif opcao\_principal == "3": remover\_funcionario()**

**elif opcao\_principal == "4": print("Programa encerrado.") break**

**else:**

**print("Opção inválida, tente novamente.")**

*Apresentação de* ***Saída do Console da Questão 4****:*

