

ATIVIDADE PRÁTICA

LINGUAGEM DE PROGRAMAÇÃO

Ivan Felipe De Oliveira Santos Dourado – 3357756

Professora Mariane Bergamini

1 EXERCÍCIOS A SEREM SOLUCIONADOS

I. Desenvolver a **classe calculadora** que faça qualquer operação matemática utilizando dois números inteiros, sendo os **dois últimos números de seu RU**. Caso o RU algum destes números sejam **zero**, **substituí-lo(s) pelo número 1**. Sendo as possíveis operações matemáticas: **soma(+)**, **subtração(-)**, **multiplicação(*)**, **divisão(/)**, **expoente(^)** e **resto da divisão (%)**. Além destas funcionalidades, o algoritmo deverá ter um **MENU** que possibilite ao usuário escolher qual o tipo de operação que se deseja realizar e que possibilite ao usuário a digitar os dois números. [Apresentar o terminal das operações matemáticas individualmente funcionando!](#)

===== INÍCIO DO CÓDIGO=====

```
class calculadora: # Crio class calculadora
def __init__(self): # dentro da class, crio função __init__ que recebe como parametro o self
self.num1 = 0 # inicializo as variaveis de self atribuido-lhes valor zero
self.num2 = 0 # inicializo as variaveis de self atribuido-lhes valor zero
self.resultado=0 # inicializo as variaveis de self atribuido-lhes valor zero
def somar(self, num1, num2): # dentro da class, crio função somar que recebe como parametro o self,
num1, num2
self.num1 = num1 # self.num1 recebe num1
self.num2=num2 # self.num1 recebe num1
self.resultado=self.num1+self.num2 # self.resultado recebe a operação matemática da função com os
dois numeros
return self.resultado # retorna novo valor de self.resultado
def subtrair(self, num1, num2):
self.num1 = num1 # self.num1 recebe num1
self.num2=num2 # self.num1 recebe num1
self.resultado=self.num1-self.num2 # self.resultado recebe a operação matemática da função com os
dois numeros
return self.resultado # retorna novo valor de self.resultado

def multiplicar(self, num1, num2):
self.num1 = num1 # self.num1 recebe num1
self.num2=num2 # self.num1 recebe num1
self.resultado=self.num1*self.num2 # self.resultado recebe a operação matemática da função com os
dois numeros
return self.resultado # retorna novo valor de self.resultado
def dividir(self, num1, num2):
self.num1 = num1 # self.num1 recebe num1
self.num2=num2 # self.num1 recebe num1
```

```
self.resultado=self.num1/self.num2 # self.resultado recebe a operação matemática da função com os dois numeros
```

```
return self.resultado # retorna novo valor de self.resultado
```

```
def expoente(self, num1, num2):
```

```
self.num1=num1 # self.num1 recebe num1
```

```
self.num2=num2 # self.num1 recebe num1
```

```
self.resultado=self.num1**self.num2 # self.resultado recebe a operação matemática da função com os dois numeros
```

```
return self.resultado # retorna novo valor de self.resultado
```

```
def resto(self, num1, num2):
```

```
self.num1 = num1 # self.num1 recebe num1
```

```
self.num2=num2 # self.num1 recebe num1
```

```
self.resultado=self.num1 % self.num2 # self.resultado recebe a operação matemática da função com os dois numeros
```

```
return self.resultado # retorna novo valor de self.resultado
```

```
def continuar(entrada): # Crio a função continuar que recebe como parametro entrada
```

```
if entrada: # Se entrada retornar True (não vazio)
```

```
return True #Retorna true
```

```
else: # senão
```

```
return False # retorna False
```

```
def menu(): # Crio função menu
```

```
# Abaixo criado o dicionário opc com as operações matematicas
```

```
opc = {1: 'Adição',
```

```
2: 'Subtração',
```

```
3: 'Multiplicação',
```

```
4: 'Divisão',
```

```
5: 'Exponenciação',
```

```
6: 'Módulo (resto)'}  
calc= calculadora() #variavel calc chama a classe calculadora como função
```

```
print("""
```

```
Qual é a operação matemática desejada?
```

```
1 -> Adição
```

```
2 -> Subtração
```

```
3 -> Multiplicação
```

```
4 -> Divisão
```

```
5 -> Exponenciação
```

```
6 -> Módulo (resto)
```

```
Digite o número desejado e aperte ENTER""") # Print do Menu calculadora
calcular = True # inicializa variavel calcular como true
while calcular: # enquanto calcular retorna true
    opcao = input("\nEscolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): ") # input para o usuario
    escolher operação
    if not(opcao in '1 2 3 4 5 6'): # Caso seja um valor fora das opções informadas
        print('Opção escolhida é inválida!') # imprime mensagem de erro no console
        continue # reinicia laço while
    else: # senão
        opcao=int(opcao) # o valor é convertido para int
        print(f'A operação matemática escolhida {opc[opcao]} é válida') # informa qque a opção escolhida é
        válida
        print('Apenas números inteiros serão trabalhados!! ') # informa que se trabalhará somente com intei-
        ros
        if opcao == 1: # se a opção for 1
            # pede para o usuario inserir os ultimos dois numeros do RU
            num1 = int(input("Digite o penúltimo número de seu RU: "))
            num2 = int(input("Digite o último número de seu RU: "))
            # variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
            resultado = calc.somar(num1,num2)
            print(f'O valor da operação de adição é {resultado}.') # printa resultado
            # pergunta se deseja continuar, re responder vaziao, a variavel calcular receberá false, interrompendo
            o laço
            calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
            dora!'))
        elif opcao == 2: # se a opção for 2
            # pede para o usuario inserir os ultimos dois numeros do RU
            num1 = int(input("Digite o penúltimo número de seu RU: "))
            num2 = int(input("Digite o último número de seu RU: "))
            # variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
            resultado = calc.subtrair(num1,num2)
            print(f'O valor da operação de subtração é {resultado}.') # printa resultado
            calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
            dora!'))
        elif opcao == 3: # se a opção for 3
            # pede para o usuario inserir os ultimos dois numeros do RU
            num1 = int(input("Digite o penúltimo número de seu RU: "))
            num2 = int(input("Digite o último número de seu RU: "))
            # variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
            resultado = calc.multiplicar(num1,num2)
            print(f'O valor da operação de multiplicação é {resultado}.') # printa resultado
            # pergunta se deseja continuar, re responder vaziao, a variavel calcular receberá false, interrompendo
            o laço
            calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
            dora!'))
```

```
elif opcao == 4: # se a opção for 4
# pede para o usuario inserir os ultimos dois numeros do RU
num1 = int(input("Digite o penúltimo número de seu RU: "))
num2 = int(input("Digite o último número de seu RU: "))
# variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
resultado = calc.dividir(num1,num2)
print(f'O valor da operação de divisão é {resultado}.') # printa resultado
# pergunta se deseja continuar, re responder vaziao, a variavel calcular receberá false, interrompendo
o laço
calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
dora!'))
elif opcao == 5: # se a opção for 5
# pede para o usuario inserir os ultimos dois numeros do RU
num1 = int(input("Digite o penúltimo número de seu RU: "))
num2 = int(input("Digite o último número de seu RU: "))
# variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
resultado = calc.expoente(num1,num2)
print(f'O valor da operação de exponenciação é {resultado}.') # printa resultado
# pergunta se deseja continuar, re responder vaziao, a variavel calcular receberá false, interrompendo
o laço
calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
dora!'))
elif opcao == 6: # se a opção for 6
# pede para o usuario inserir os ultimos dois numeros do RU
num1 = int(input("Digite o penúltimo número de seu RU: "))
num2 = int(input("Digite o último número de seu RU: "))
# variavel resultado recebe a função contida na classe que realiza a respectiva operação matemática
resultado = calc.resto(num1,num2)
print(f'O valor da operação de módulo(resto) é {resultado}.') # printa resultado
# pergunta se deseja continuar, re responder vaziao, a variavel calcular receberá false, interrompendo
o laço
calcular = continuar(input('Digite alguma coisa para continuar ou apertar ENTER para sair da calcula-
dora!'))
print('Cálculo encerrado.') # Informa o encerramento a cada cálculo.
```

===== fim do código =====

IMAGEM DO EXERCÍCIO:

Adição:

```
menu()

Qual é a operação matemática desejada?
1 -> Adição
2 -> Subtração
3 -> Multiplicação
4 -> Divisão
5 -> Exponenciação
6 -> Módulo (resto)

Digite o número desejado e aperte ENTER

Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 1
A operação matemática escolhida Adição é válida
Apenas números inteiros serão trabalhados!!
Digite o penúltimo número de seu RU: 5
Digite o último número de seu RU: 6
O valor da operação de adição é 11.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
```

Subtração:

```
O valor da operação de adição é 11.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
Cálculo encerrado.

Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 2
A operação matemática escolhida Subtração é válida
Apenas números inteiros serão trabalhados!!
Digite o penúltimo número de seu RU: 5
Digite o último número de seu RU: 6
O valor da operação de subtração é -1.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
```

Multiplicação:

```
O valor da operação de subtração é -1.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
Cálculo encerrado.

Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 3
A operação matemática escolhida Multiplicação é válida
Apenas números inteiros serão trabalhados!!
Digite o penúltimo número de seu RU: 5
Digite o último número de seu RU: 6
O valor da operação de multiplicação é 30.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
```

Divisão:

```
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
Cálculo encerrado.

Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 4
Digite o penúltimo número de seu RU: 5
A operação matemática escolhida Divisão é válida
Apenas números inteiros serão trabalhados!!
Digite o último número de seu RU: 6
O valor da operação de divisão é 0.8333333333333334.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!>
```

Módulo(resto da divisão):

```
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!u
Cálculo encerrado.
```

```
Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 6
A operação matemática escolhida Módulo (resto) é válida
Apenas números inteiros serão trabalhados!!
Digite o penúltimo número de seu RU: 5
Digite o último número de seu RU: 6
O valor da operação de módulo(resto) é 5.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!
```

Exponenciação:

```
Cálculo encerrado.

Escolha a opção de cálculo desejada (1, 2, 3, 4, 5 ou 6): 5
A operação matemática escolhida Exponenciação é válida
Apenas números inteiros serão trabalhados!!
Digite o penúltimo número de seu RU: 5
Digite o último número de seu RU: 6
O valor da operação de exponenciação é 15625.
Digite alguma coisa para continuar ou apertar ENTER para sair da calculadora!
Cálculo encerrado.
```

II. Dada a equação linear $y = ax + bx - c$. Para os valores de **a**, **b** e **c** serão os **três últimos números de seu RU**. Caso, **algum número do RU seja igual a zero**, substituí-lo(s) pelo **número 3**. Realizar o plot dos resultados, dado os valores para **x1 = 10**; **x2 = 8** e **x3 = 15**. Utilizar os comandos apresentados na Aula Prática 1; **colocar legenda no gráfico**, **alterar a cor da linha(s) ou pontos de seu gráfico**; **nomear o eixo x**, **nomear o eixo y**.

Exemplo de cálculo: Meu RU é 205 103 -> por ter zero fica assim: a=1; b=3; c=3

$$y = 1*5 + 3*5 - 3$$

$$y = 5 + 15 - 3 = 17$$

$$y = 1*7 + 7*3 - 3$$

$$y = 7 + 21 - 3 = 25$$

$$y = 1*9 + 9*3 - 3$$

$$y = 9 + 27 - 3 = 33$$

===== início do código =====

```
import matplotlib.pyplot as plt # importo a biblioteca matplotlib
```

```
# automatizando o cálculo, tornando-o um algoritmo global
class equacao: #crio a classe equação chamando os valores para X
def __init__(self, x): # função init com parametros self e x
self.x = int(x) # Seu valor será inteiro

def valor(self): # retorna os valores
#abaixo é uma docstring
"""RU: 3357756
a,b,c => 3 últimos números do meu RU
y=> retorna o resultado da função do 1º grau"""
a=7 # valores do RU
b=5 # valores do RU
c=6 # valores do RU
y=self.x*a + self.x*b - c # retorna a equação do 1º com respectivo valor de x
return y # retorna valor de y

eixoX = [10,8,15] # pontos dados do eixo X
eixoY=[] # Eixo y/ coordenadas recebe lista vazia

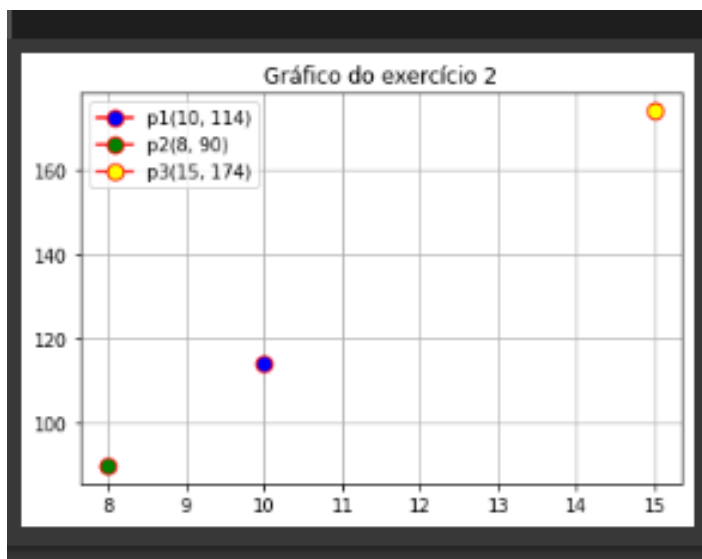
for i in eixoX: # iteração, para cada item da lista
y = equacao(i) # y recebe resultado da classe função equação
eixoY.append(y.valor()) # Será inserido ao fim da lista eixoY cada valor de y

plt.plot(eixoX[0], eixoY[0], color = 'red', marker="o", markersize=9, markerfacecolor='blue') # realizar
grafico com os parametros eixoX e eixoY, e cor
plt.plot(eixoX[1], eixoY[1], color = 'red', marker="o", markersize=9, markerfacecolor='green') # reali-
zar grafico com os parametros eixoX e eixoY, e cor
plt.plot(eixoX[2], eixoY[2], color = 'red', marker="o", markersize=9, markerfacecolor='yellow') # rea-
lizar grafico com os parametros eixoX e eixoY, e cor

plt.legend([f'p1({eixoX[0]}, {eixoY[0]}), f'p2({eixoX[1]}, {eixoY[1]}), f'p3({eixoX[2]}, {eixoY[2]})']) #
Legenda por ponto de acordo com seu índice na lista
plt.title('Gráfico do exercício 2') # título do gráfico
plt.grid() # fundo quadriculado
```

===== Fim do Código =====

IMAGEM DO EXERCÍCIO:



III. **Realizar o upload do arquivo STORES.csv.** Renomear todas as colunas do arquivo STORES.csv, onde os respectivos nomes sejam compactados (Exemplo: Daily Customer Count foi renomeado para Visitantes). Após isto, para se analisar o desempenho das lojas de supermercado/mercado do arquivo STORES.csv encontre os valores **mínimo, máximo, médio e desvio padrão** das seguintes colunas: "Items_Available"; "Daily_Customer_Count"; e "Store_Sales".

Algumas informações extras sobre a tabela do arquivo **STORES.csv**:

- *ID da loja: (Índice) ID da loja específica.*
- *Store_Area: Número de itens diferentes disponíveis na loja correspondente.*
- *DailyCustomerCount: Número de clientes que visitaram as lojas em média ao longo do mês.*
- *Store_Sales: Vendas em (US\$) que as lojas realizaram.*

===== Início do código =====

```
import pandas as pd # importando biblioteca pandas
```

```
from google.colab import drive,files # do google.colab importando drive e file
```

```
drive.mount('/content/drive') # 'monto' o drive, de modo a aceder acesso deste notebook a meus di-  
retórios no google drive
```

```
#importando arquivoStores.csv
```

```
caminho = '/content/drive/MyDrive/ciencia_de_dados_Uninter/AP-linguagem_prog/archive_sto-  
res/Stores.csv' # var caminho recebe caminho do arquivo no directorio do drive
```

```
df = pd.read_csv(caminho, sep=',', encoding='iso-8859-15') # ler arquivo csv read_csv('caminho do ar-  
quivo no drive'); separado por virgula; encoding padrao da europa ocidental  
df.head() # exibir cabeçalho/ primeiras 5 linhas
```

```
df.columns.values # exibe valor das colunas
```

```
#renomeando colunas
```

```
df_colunas = ['>Store ID ', 'Store_Area', 'Items_Available', 'Daily_Customer_Count', 'Store_Sales'] # pe-  
gando as colunas para renomear
```

```
df_filtrado=df.filter(items=df_colunas) # filtrar as colunas
```

```
df_filtrado=df_filtrado.rename(columns={ # renomeando cada coluna 'nome_antigo':'nome_novo'
```

```
'>Store ID ':'ID',
```

```
'Store_Area':'Itens',
```

```
'Items_Available':'Ítems_disponíveis',
```

```
'Daily_Customer_Count':'Visitantes_mensais',
```

```
'Store_Sales':'Vendas_da_loja(US$)'
```

```
}})
```

```
df_filtrado # chama exibindo as colunas renomeadas
```

```
#consultas estatísticas:
```

```
print('Visitantes:')
```

```
dados = df_filtrado['Visitantes_mensais'] # trazendo dados da coluna Visitantes_mensais
```

```
print(f'O valor máximo foi de {dados.values.max()}') # imprime o valor máximo
```

```
print(f'O valor mínimo foi de {dados.values.min()}') # imprime o valor mínimo
```

```
print(f'O valor médio foi de {dados.values.mean()}') # imprime o valor médio
```

```
print(f'O valor do desvio padrão foi de {dados.values.std()}') # imprime o valor do desvio padrão
```

```
print('Vendas em US$:')
```

```
dados = df_filtrado['Vendas_da_loja(US$)'] # trazendo dados da coluna Vendas_da_loja(US$)
```

```
print(f'O valor máximo foi de {dados.values.max()}') # imprime o valor máximo
```

```
print(f'O valor mínimo foi de {dados.values.min()}') # imprime o valor mínimo
```

```
print(f'O valor médio foi de {dados.values.mean()}') # imprime o valor médio
```

```
print(f'O valor do desvio padrão foi de {dados.values.std()}') # imprime o valor do desvio padrão
```

```
print('Ítems disponíveis:')
```

```
dados = df_filtrado['Ítems_disponíveis'] # trazendo dados da coluna Ítems_disponíveis
```

```
print(f'O valor máximo foi de {dados.values.max()}') # imprime o valor máximo
```

```
print(f'O valor mínimo foi de {dados.values.min()}') # imprime o valor mínimo
```

```
print(f'O valor médio foi de {dados.values.mean()}') # imprime o valor médio
```

```
print(f'O valor do desvio padrão foi de {dados.values.std()}') # imprime o valor do desvio padrão
```

===== Fim do Código =====

IMAGENS DO EXERCÍCIO:

Importando bibliotecas e montando o drive:

```
[7] import pandas as pd # importando biblioteca pandas
    from google.colab import drive,files # do google.colab importando drive e file

[8] drive.mount('/content/drive') # 'monto' o drive, de modo a aceder acesso deste notebook a meus diretórios no google drive

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Importando arquivo Stores.csv , lendo-o e exibindo seu cabeçalho;
exibindo o valor das colunas do arquivo;

```
[9] #importando arquivoStores.csv
    caminho = '/content/drive/MyDrive/ciencia de dados Uninter/AP-linguagem prog/archive stores/Stores.csv' # var caminho
    df = pd.read_csv(caminho, sep=',', encoding='iso-8859-15') # ler arquivo csv read_csv('caminho do arquivo no drive');
    df.head() # exibir cabeçalho/ primeiras 5 linhas
```

	i>Store ID	Store_Area	Items_Available	Daily_Customer_Count	Store_Sales
0	1	1659	1961	530	66490
1	2	1461	1752	210	39820
2	3	1340	1609	720	54010
3	4	1451	1748	620	53730
4	5	1770	2111	450	46620

```
[ ] df.columns.values # exibe valor das colunas

array(['i>Store ID ', 'Store_Area', 'Items_Available',
      'Daily_Customer_Count', 'Store_Sales'], dtype=object)
```

Renomeando suas colunas e chamando o arquivo com as devidas edições:

```
[ ] #renomeando colunas
df_colunas = ['i>i>Store ID ', 'Store_Area', 'Items Available','Daily_Customer_Count', 'Store_Sales'] # pegando as colunas pa
df_filtrado=df.filter(items=df_colunas) # filtrar as colunas
df_filtrado=df_filtrado.rename(columns={ # renomeando cada coluna 'nome_antigo':'nome_novo'
    'i>i>Store ID ':'ID',
    'Store_Area':'Itens',
    'Items Available':'itens_disponiveis',
    'Daily_Customer_Count':'Visitantes_mensais',
    'Store_Sales':'Vendas_da_loja(US$)'
})
df_filtrado # chama exibindo as colunas renomeadas
```

	ID	Itens	itens_disponiveis	Visitantes_mensais	Vendas_da_loja(US\$)
0	1	1659	1961	530	66490
1	2	1461	1752	210	39820
2	3	1340	1609	720	54010
3	4	1451	1748	620	53730
4	5	1770	2111	450	46620
...
891	892	1582	1910	1080	66390
892	893	1387	1663	850	82080
893	894	1200	1436	1060	76440
894	895	1299	1560	770	96610
895	896	1174	1429	1110	54340

896 rows x 5 columns

realizando as consultas estatísticas solicitadas (min, max, media, desvio padrão)
para as colunas Visitantes, vendas da loja e itens disponíveis.

```
[ ] #consultas estatísticas:
print('Visitantes:')
dados = df_filtrado['Visitantes_mensais'] # trazendo dados da coluna Visitantes mensais
print(f'0 valor máximo foi de {dados.values.max()}' ) # imprime o valor máximo
print(f'0 valor mínimo foi de {dados.values.min()}' ) # imprime o valor mínimo
print(f'0 valor médio foi de {dados.values.mean()}' ) # imprime o valor médio
print(f'0 valor do desvio padrão foi de {dados.values.std()}' ) # imprime o valor do desvio padrão

Visitantes:
0 valor máximo foi de 1560
0 valor mínimo foi de 10
0 valor médio foi de 786.3504464285714
0 valor do desvio padrão foi de 265.24114272571586

[ ] print('Vendas em US$:')
dados = df_filtrado['Vendas_da_loja(US$)'] # trazendo dados da coluna Vendas da loja(US$)
print(f'0 valor máximo foi de {dados.values.max()}' ) # imprime o valor máximo
print(f'0 valor mínimo foi de {dados.values.min()}' ) # imprime o valor mínimo
print(f'0 valor médio foi de {dados.values.mean()}' ) # imprime o valor médio
print(f'0 valor do desvio padrão foi de {dados.values.std()}' ) # imprime o valor do desvio padrão

Vendas em US$:
0 valor máximo foi de 116320
0 valor mínimo foi de 14920
0 valor médio foi de 59351.30580357143
0 valor do desvio padrão foi de 17181.14616881951

[ ] print('Itens disponíveis:')
dados = df_filtrado['itens_disponiveis'] # trazendo dados da coluna itens disponíveis
print(f'0 valor máximo foi de {dados.values.max()}' ) # imprime o valor máximo
print(f'0 valor mínimo foi de {dados.values.min()}' ) # imprime o valor mínimo
print(f'0 valor médio foi de {dados.values.mean()}' ) # imprime o valor médio
print(f'0 valor do desvio padrão foi de {dados.values.std()}' ) # imprime o valor do desvio padrão

Itens disponíveis:
0 valor máximo foi de 2667
0 valor mínimo foi de 932
0 valor médio foi de 1782.0357142857142
0 valor do desvio padrão foi de 299.7046664924438
```