**ATIVIDADE PRÁTICA   
LINGUAGEM DE PROGRAMAÇÃO**

**ROBSON CRUZ DE MELO – 3773638  
Professora Me. Mariane G. Bergamini**

GRAVATAÍ  
2023

# EXEMPLO DE RESOLUÇÃO

*Exercício 01 exemplo:* Realizar uma classe veículo que apresente o tipo do carro, modelo, fabricante e quantidade de passageiros. Além disso, utilizar os metódos GETTER e SETTER para incluir as classses Modelo retornando o modelo e o fabricante do carro e outra classe Quantidade de Passageiro no carro.

**RESPOSTA DO ALUNO**

**COLE AQUI O SEU CÓDIGO FONTE (usar os comandos Ctrl + c / Ctrl + v) :**

class Veiculo:

  def \_\_init\_\_(self, tipo, modelo, fabricante, qtd\_passageiro):

    self.tipo = tipo

    self.modelo = modelo

    self.fabricante = fabricante

    self.qtd\_passageiro = qtd\_passageiro

    # getter: adicionando

  def getModelo(self):

    return f"{self.fabricante} {self.modelo}"

  def getQtdPassageiro(self):

    return f"{self.qtd\_passageiro}"

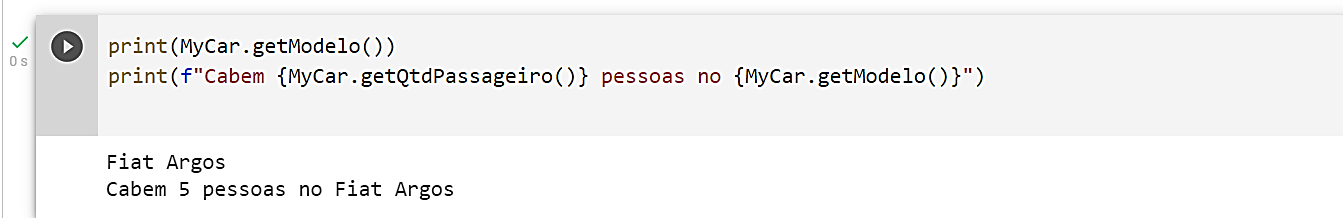
MyCar = Veiculo('carro',modelo = 'Argos', fabricante='Fiat',qtd\_passageiro=5)

MyBus = Veiculo('Ônibus', modelo = 'Bus202', fabricante='Mercedes', qtd\_passageiro=40)

print(MyCar.getModelo())

print(f"Cabem {MyCar.getQtdPassageiro()} pessoas no {MyCar.getModelo()}")

**COLE AQUI IMAGEM(NS)/PRINT(S) DO TERMINAL SENDO EXCECUTADO SEM ERRO**:



# EXERCÍCIOS A SEREM SOLUCIONADOS PELO ALUNO :

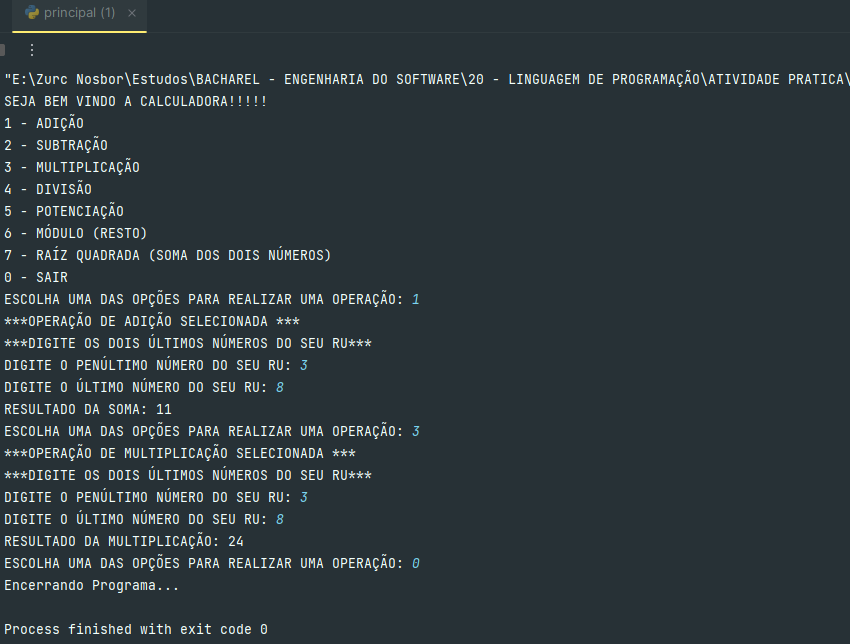
Exercício 1 – Desenvolver uma **classe calculadora** que faça operções matemáticas utilizando dois números inteiros, sendo os **dois últimos números de seu RU**. Os dois números inteiros devem ser solicitados para o usuário digitar. Se o seu **RU** for **zero**, **substituí-lo(s) pelo número 5**. Sendo as possíveis operações matemáticas: **soma(+), subtração(-), multiplicação(\*), divisão(/), expoente (^), resto(%) e raíz quadrada da soma dos dois números ( sqrt(Num1 + Num2) )**. Além destas funcionalidades, o algoritmo deverá ter um **MENU** que possíbilite ao usuário escolher qual o tipo de operação que se deseja realizar e que possibilite ao usuário a digitar os dois números. *Apresentar todas as operações matemáticas da calculadora funcionando!*

**RESPOSTA DO ALUNO:**

*#CLASSE CALCULADORA  
#IMPORT PARA REALIZAR OS CALCULOS MATEMATICOS*import math  
  
*#CLASSE CALCULADORA E SEUS METODOS COM AS OPERAÇOES MATEMATICAS*class Calculadora:  
  
 *#METODO DE INICIALIZAÇÃO DA CLASSE* def *\_\_init\_\_*(*self*):  
 *self*.n1 = 0  
 *self*.n2 = 0  
 *self*.res = 0  
  
 def somar(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 + *self*.n2  
 return *self*.res  
  
 def subtrair(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 - *self*.n2  
 return *self*.res  
  
 def multiplicar(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 \* *self*.n2  
 return *self*.res  
  
 def dividir(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 / *self*.n2  
 return *self*.res  
  
 def expoente(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 \*\* *self*.n2  
 return *self*.res  
  
 def resto(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = *self*.n1 % *self*.n2  
 return *self*.res  
  
 def raiz(*self*, *n1*, *n2*):  
 *self*.n1 = *n1  
 self*.n2 = *n2  
 self*.res = math.sqrt(*self*.n1 + *self*.n2)  
 return *self*.res

*#CLASSE PRINCIPAL  
#IMPORTAR DA CLASSE CALCULADORA*from calculadora import Calculadora  
  
class Principal:  
  
 *#CONSTRUTOR DA CLASSE PRINCIPAL  
 #INICIALIZA A INSTANCIA DA CLASSE CALCULADORA COMO UM ATRIBUTO DA CLASSE PRINCIPAL* def *\_\_init\_\_*(*self*):  
 *self*.calculadora = Calculadora()  
  
 *#METODO PARA REALIZAR OS CALCULOS* def executar\_calculos(*self*):  
  
 *# MENU INICIAL  
 print*("SEJA BEM VINDO A CALCULADORA!!!!!")  
 *print*("1 - ADIÇÃO")  
 *print*("2 - SUBTRAÇÃO")  
 *print*("3 - MULTIPLICAÇÃO")  
 *print*("4 - DIVISÃO")  
 *print*("5 - POTENCIAÇÃO")  
 *print*("6 - MÓDULO (RESTO)")  
 *print*("7 - RAÍZ QUADRADA (SOMA DOS DOIS NÚMEROS)")  
 *print*("0 - SAIR")  
  
 *# LAÇO DE REPETIÇÃO PARA A ESCOLHA DA OPÇÃO* while True:  
 op = *input*("ESCOLHA UMA DAS OPÇÕES PARA REALIZAR UMA OPERAÇÃO: ")  
  
 *#CONDICIONAL COM AS OPÇOES ESCOLHIDAS E SUAS DEVIDAS OPREÇÃOES* if op == '1':  
  
 *print*("\*\*\*OPERAÇÃO DE ADIÇÃO SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.somar(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA SOMA:", resultado)  
  
 if op == '2':  
  
 *print*("\*\*\*OPERAÇÃO DE SUBTRAÇÃO SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.subtrair(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA SUBTRAÇÃO:", resultado)  
  
 if op == '3':  
  
 *print*("\*\*\*OPERAÇÃO DE MULTIPLICAÇÃO SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.multiplicar(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA MULTIPLICAÇÃO:", resultado)  
  
 if op == '4':  
  
 *print*("\*\*\*OPERAÇÃO DE DIVISÃO SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.dividir(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA DIVISÃO:", resultado)  
  
 if op == '5':  
  
 *print*("\*\*\*OPERAÇÃO DE EXPONENCIAÇÃO SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.expoente(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA POTENCIAÇÃO:", resultado)  
  
 if op == '6':  
  
 *print*("\*\*\*OPERAÇÃO DE MÓDULO(RESTO) SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.resto(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DO MÓDULO(RESTO):", resultado)  
  
 if op == '7':  
  
 *print*("\*\*\*OPERAÇÃO DE RAIZ QUADRADA SELECIONADA \*\*\*")  
 *print*("\*\*\*DIGITE OS DOIS ÚLTIMOS NÚMEROS DO SEU RU\*\*\*")  
  
 *# LAÇO DE REPETIÇÃO PARA NÚMERO VÁLIDO* while True:  
 num1 = *input*("DIGITE O PENÚLTIMO NÚMERO DO SEU RU: ")  
 if num1.isdigit():  
 num1 = *int*(num1)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 while True:  
 num2 = *input*("DIGITE O ÚLTIMO NÚMERO DO SEU RU: ")  
 if num2.isdigit():  
 num2 = *int*(num2)  
 break  
 else:  
 *print*("Erro: DIGITE UM NÚMERO VÁLIDO!")  
  
 resultado = *self*.calculadora.raiz(*n1*=num1, *n2*=num2)  
 *print*("RESULTADO DA RAIZ(SOMA DOS DOIS NÚMEROS DIGITADOS):", resultado)  
  
 *# OPÇÃO 0: ENCERRA O PROGRAMA* elif op == '0':  
 *print*('Encerrando Programa...')  
 return  
  
 *# OPÇÃO INVÁLIDA* elif not op.isdigit() or *int*(op) not in *range*(8):  
 *print*('Erro: Digite uma opção válida!!')  
  
*# INSTANCIA DA CLASSE PRINCIPAL*p = Principal()  
  
*# EXECUÇÃO DOS CALCULOS*p.executar\_calculos()

CÓDIGO EXERCÍCIO – 01

TERMINAL EXERCÍCIO - 01

Exercício 2 – Dada uma determinada equação linear **y = ax + bx – c**, sendo que os valores para **a, b e c** serão os **três primeiros números de seu RU (a = NUM1, b = NUM2, c = NUM3)**. Caso, **algum número do RU seja igual a zero**, subistituí-lo(s) pelo **número 5**. Além disso, será preciso criar um vetor aleatório de tamanho 10, onde cada posição do vetor conterá os valores de **x** **para a equação linear**. Feito isto, fazer um gráfico para mostrar os pontos obtidos pela equação linear dentro do plano cartesiano. Por fim, nomear os eixos Y e X do gráfico, colocar cores diferentes para os pontos e colocar legenda. ***Dica****: você vai ter no total, 10 pontos no seu plano cartesiano, ou seja, 10 pontos serão ilustrados no gráfico.*

**RESPOSTA DO ALUNO:**

*# IMPORT PARA A GERAÇÃO DE GRAFICOS*import matplotlib.pyplot as plt  
  
*# IMPORT DO RANDOM PARA GERAR NUMEROS ALEATORIOS*import random  
  
*# MEU RU - COMO PEDE NO ENUNCIADO, UTILIZEI OS 3 PRIMEIROS DIGITOS*RU = 3773638  
a = 3  
b = 7  
c = 7  
  
*# GERA VETOR COM 10 POSIÇÕES COM NÚMEROS ALEÁTORIOS DO NÚMERO 1 AO 25 COMO EXEMPLO*vetorX = random.sample(*range*(1, 25), 10)  
  
*# CALCULA OS VALORES DE Y COM A EQUAÇÃO PARA VALOR DE X NO VETOR*y = [a \* vetorXi + b \* vetorXi - c for vetorXi in vetorX]  
  
*# ORDENA OS PONTOS EM ORDEM CRESCENTE COM BASE NO VETOR*pontos\_ordenados = *sorted*(*zip*(vetorX, y))  
  
vetorX\_ordenado, y\_ordenado = *zip*(\*pontos\_ordenados)  
  
*# PLOTAR OS PONTOS COM CORES*for i in *range*(*len*(vetorX\_ordenado)):  
 plt.plot(vetorX\_ordenado[i], y\_ordenado[i], *marker*="o", *markersize*=12)  
  
*# NOMEAR EIXO X*plt.xlabel('EIXO X')  
  
*# NOMEAR EIXO Y*plt.ylabel('EIXO Y')  
  
*# LEGENDA DO GRÁFICO*legend\_labels = [f'POSIÇÃO X={vetorX\_ordenado[i]} POSIÇÃO Y={y\_ordenado[i]}' for i in *range*(*len*(vetorX\_ordenado))]  
plt.legend(legend\_labels)  
  
*# GRADE NO GRÁFICO*plt.grid()  
  
*# TÍTULO DO GRÁFICO*plt.title('Exercício 02 - EQUAÇÃO LINEAR')  
  
*# MOSTRAR GRÁFICO*plt.show()

CÓDIGO EXERCÍCIO – 02

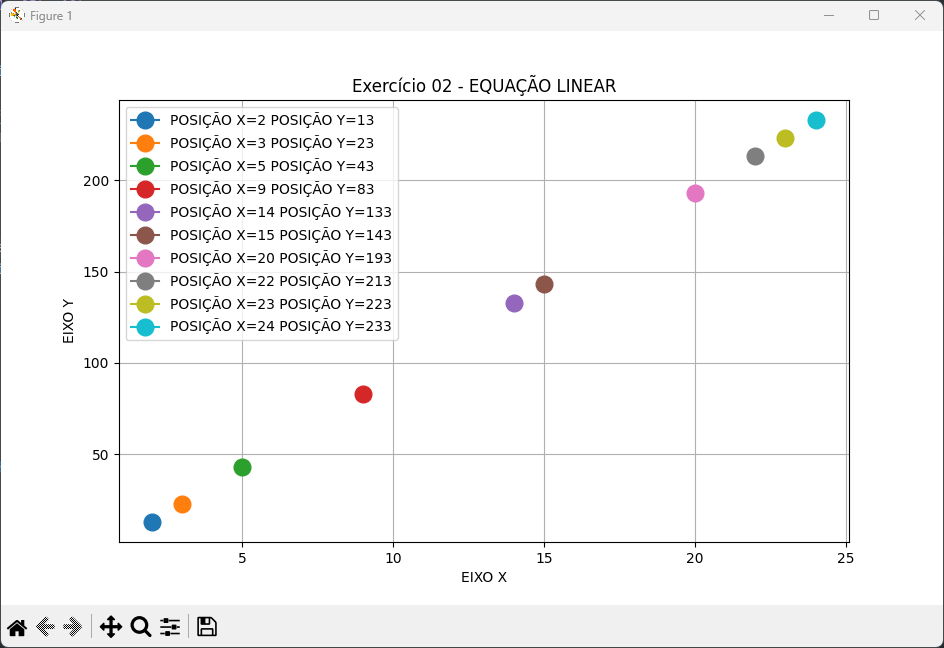
****

GRÁFICO EXERCÍCIO - 02

Exercício 3 – 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 seguinte colunas: **"Items\_Available"; "Daily\_Customer\_Count"; e "Store\_Sales".** Posto isto, realizar três gráficos com as seguintes informações: Items\_Available, Daily\_Customer\_Count e Store\_Sales. Não se esqueça de colocar: nomes para os eixos Y e X do gráfico, colocar cores diferentes para os pontos e colocar legenda.

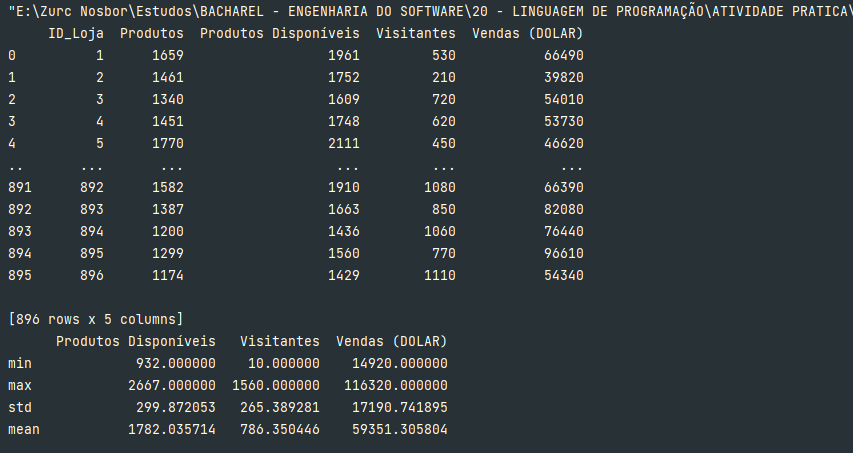
Algumas informações extras sobre a tabela do arquivo **STORES.csv**:

* *ID da loja: (Índice) ID da loja específica.*
* *Store da loja: Área Física da loja em pátio.*
* *Itens Avaliados: Número de itens diferentes disponíveis na loja correspondente.*
* *Contagem diária de clientes: Número de clientes que visitaram as lojas em média ao longo do mês.*
* *Histórico de vendas: Vendas em (US$) que as lojas realizaram.*

**RESPOSTA DO ALUNO:**

*# IMPORT DA BIBLIOTECA PANDAS PARA MANIPULAÇÃO E ANÁLISE DE DADOS*import pandas as pd  
  
*# IMPORT PARA A GERAÇÃO DE GRAFICOS*import matplotlib.pyplot as plt  
  
*# LER O ARQUIVO "Stores.csv" NA PASTA*df = pd.read\_csv("Stores.csv", *sep*=',', *encoding*='ISO 8859-1')  
  
*# RENOMEAR TODAS AS COLUNAS DO ARQUIVO*df = df.rename(*columns*={  
 df.columns[0]: "ID\_Loja",  
 df.columns[1]: "Produtos",  
 df.columns[2]: "Produtos Disponíveis",  
 df.columns[3]: "Visitantes",  
 df.columns[4]: "Vendas (DOLAR)",  
})  
  
*#VER O ARQUIVO NO TERMINAL  
print*(df)  
  
*# SELECIONAR AS COLUNAS PARA BUSCAR OS DADOS: PRODUTOS DISPONÍVEIS, VISITANTES E VENDAS DA LOJA*col\_interesse = ["Produtos Disponíveis", "Visitantes", "Vendas (DOLAR)"]  
df\_col\_interesse = df[col\_interesse]  
  
*#BUSCAR OS DADOS MÍNIMO, MÁXIMO E DESVIO PADRÃO*dados = df\_col\_interesse.agg(["min", "max", "std", "mean"])  
  
*#VER O ARQUIVO NO TERMINAL  
print*(dados)  
  
*# AMOSTRA DE 30% DOS DADOS DA COLUNA "PRODUTOS DISPONIVEIS" DE ACORDO COM A MENSAGEM DA TUTORIA*amostra = df["Produtos Disponíveis"].sample(*frac*=0.3)  
  
*#GRÁFICO DE PRODUTOS DISPONÍVEIS, ACABEI NÃO COLOCANDO CORES ALEATÓRIAS PELA FATO QUE GEROU TRAVAMENTOS NO PLANO CARTESIANO*plt.scatter(amostra.index, amostra, *color*='blue', *label*='Produtos Disponíveis')  
  
plt.xlabel('Índice')  
plt.ylabel('Produtos Disponíveis')  
plt.title('Análise de Produtos Disponíveis')  
plt.legend()  
plt.grid()  
plt.show()  
  
*# AMOSTRA DE 30% DOS DADOS DA COLUNA "VISITANTES" DE ACORDO COM A MENSAGEM DA TUTORIA*amostra\_2 = df["Visitantes"].sample(*frac*=0.3)  
  
*#GRÁFICO DE PRODUTOS DISPONÍVEIS, ACABEI NÃO COLOCANDO CORES ALEATÓRIAS PELA FATO QUE GEROU TRAVAMENTOS NO PLANO CARTESIANO*plt.scatter(amostra\_2.index, amostra\_2, *color*='green', *label*='Visitantes')  
  
plt.xlabel('Índice')  
plt.ylabel('Visitantes')  
plt.title('Análise de Visitantes')  
plt.legend()  
plt.grid()  
plt.show()  
  
*# AMOSTRA DE 30% DOS DADOS DA COLUNA "Vendas (DOLAR)" DE ACORDO COM A MENSAGEM DA TUTORIA*amostra\_3 = df["Vendas (DOLAR)"].sample(*frac*=0.3)  
  
*#GRÁFICO DE PRODUTOS DISPONÍVEIS, ACABEI NÃO COLOCANDO CORES ALEATÓRIAS PELA FATO QUE GEROU TRAVAMENTOS NO PLANO CARTESIANO*plt.scatter(amostra\_3.index, amostra\_3, *color*='red', *label*='Vendas da Loja (DOLAR)')  
  
plt.xlabel('Índice')  
plt.ylabel('Vendas da Loja (DOLAR)')  
plt.title('Análise de Vendas da Loja (DOLAR)')  
plt.legend()  
plt.grid()  
plt.show()

CÓDIGO EXERCÍCIO – 03



TERMINAL EXERCÍCIO - 03

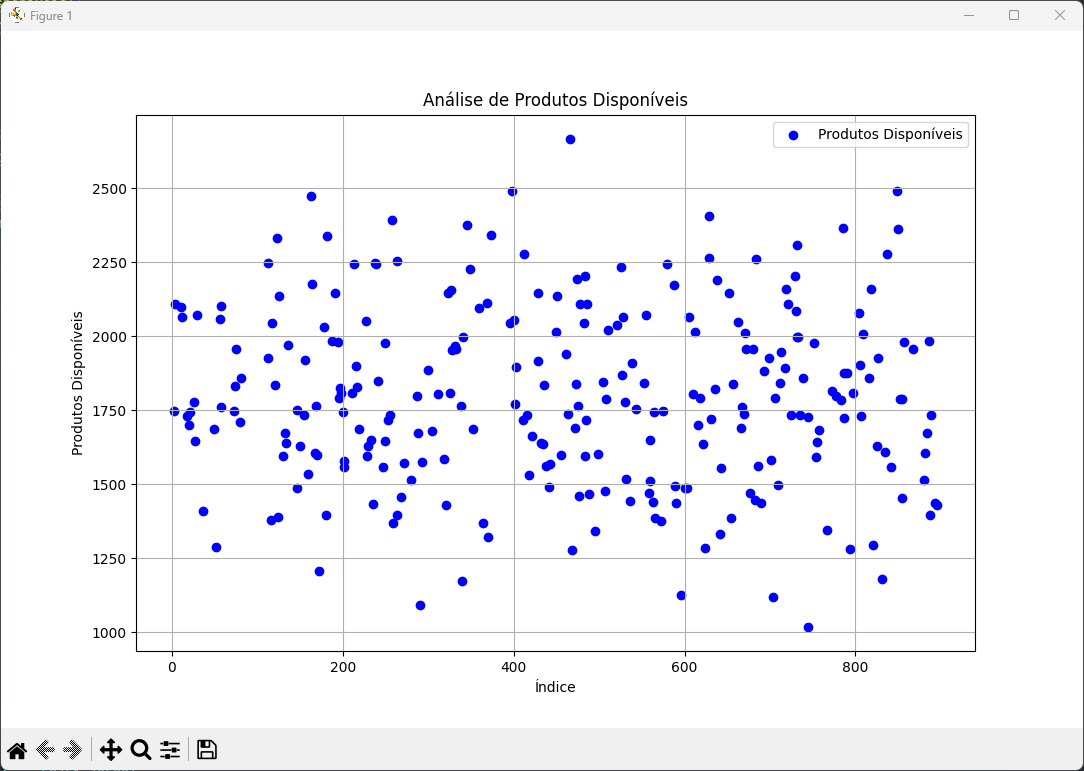


GRÁFICO 01 EXERCÍCIO - 03

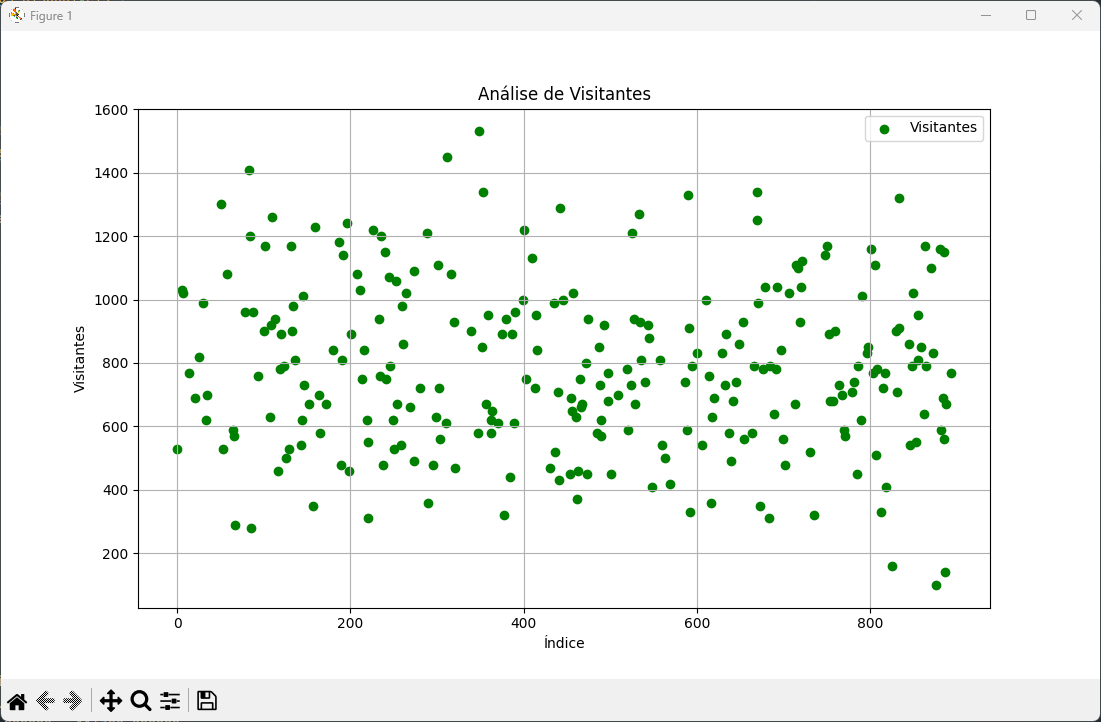
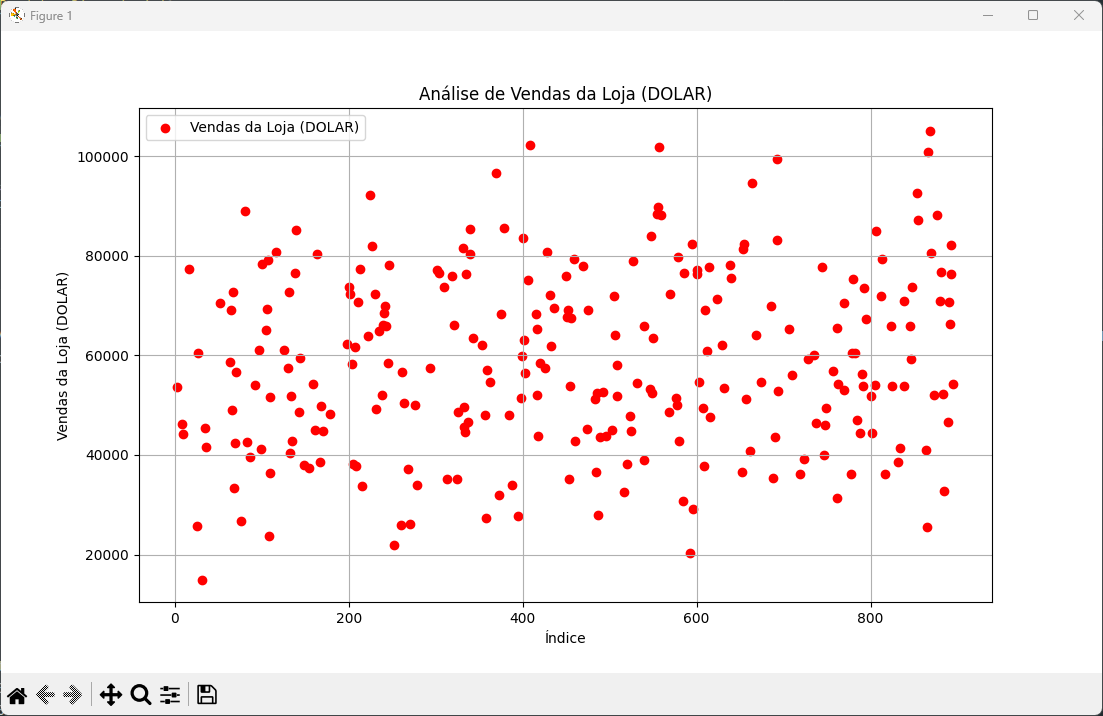


GRÁFICO 02 EXERCÍCIO – 03

GRÁFICO 03 EXERCÍCIO - 03