**ATIVIDADE PRÁTICA   
LINGUAGEM DE PROGRAMAÇÃO**

**MOACIR DOMINGOS DA SILVA JUNIOR - RU: 3539252   
  
Prof.ª ME MARIANE G. B. FERNANDES**

DIADEMA  
2022

# EXERCÍCIOS A SEREM SOLUCIONADOS

I. Desenvolver a classe calculadora que faça qualquer operação matemática utilizando dois números inteiros, sendo os dois últimos número de seu **RU**. Caso o RU termine com zero, substituí-lo pelo número 1. Sendo as possíveis operações matemáticas: **soma, subtração, multiplicação, divisão, exponenciação e módulo**. Para isto, o algoritmo deverá ter um **MENU** que possibilite ao usuário escolher qual o tipo de operação que se deseja realizar. *Apresentar todas as operações matemáticas da calculadora funcionando!*

#RU: 3539252

class Calculadora():

  def \_\_init\_\_(self, a=5, b=2, opcao=None):

      self.a = a

      self.b = b

      self.opcao = opcao

  def start(self):

    print('''

      [1] Somar

      [2] Subtrair

      [3] Multiplicar

      [4] Dividir

      [5] Exponenciação

      [6] Módulo''')

    self.opcao = int(input())

    self.resultado()

  def somar(self):

    return self.a+self.b

  def subtrair(self):

    return self.a-self.b

  def multiplicacao(self):

    return self.a\*self.b

  def dividir(self):

    return self.a/self.b

 def exponenciacao(self):

    return self.a\*\*self.b

  def modulo(self):

    return self.a%self.b

  def resultado(self):

    if self.opcao == 1:

      print(f"Soma = {self.somar()}")

    elif self.opcao == 2:

      print(f"Subtração = {self.subtrair()}")

    elif self.opcao == 3:

      print(f"Multiplicação = {self.multiplicacao()}")

    elif self.opcao == 4:

      print(f"Divisão = {self.dividir()}")

    elif self.opcao == 5:

      print(f"Exponenciação = {self.exponenciacao()}")

    elif self.opcao == 6:

      print(f"Módulo = {self.modulo()}")

    else:

      print("Opção inválida")

  def todos\_resultado(self):

    print(f"Soma = {self.somar()}")

    print(f"Subtração = {self.subtrair()}")

    print(f"Multiplicação = {self.multiplicacao()}")

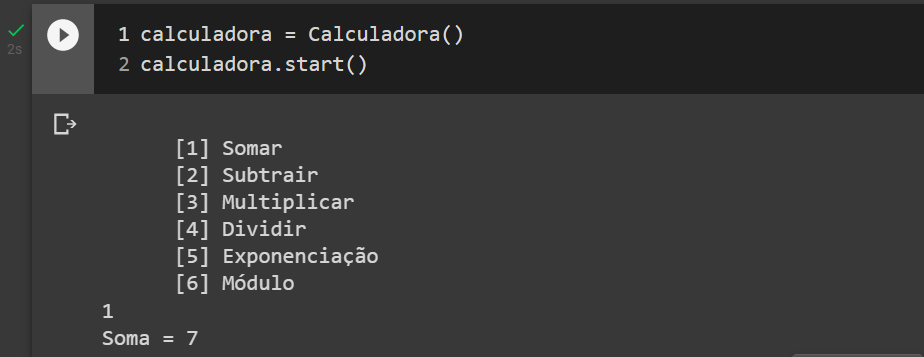
    print(f"Divisão = {self.dividir()}")

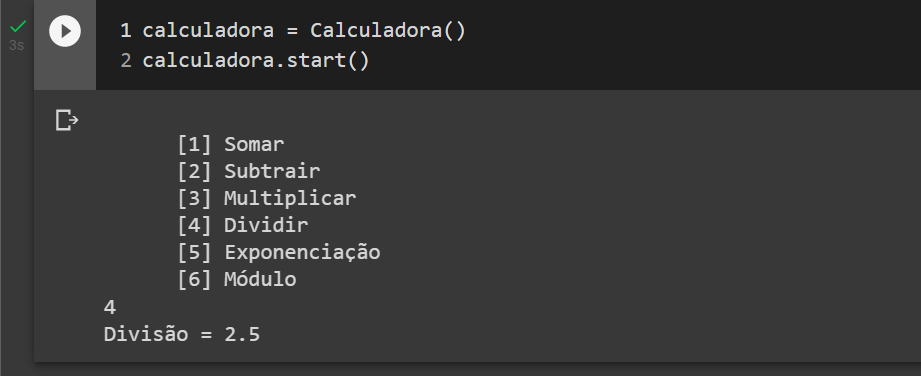
    print(f"Exponenciação = {self.exponenciacao()}")

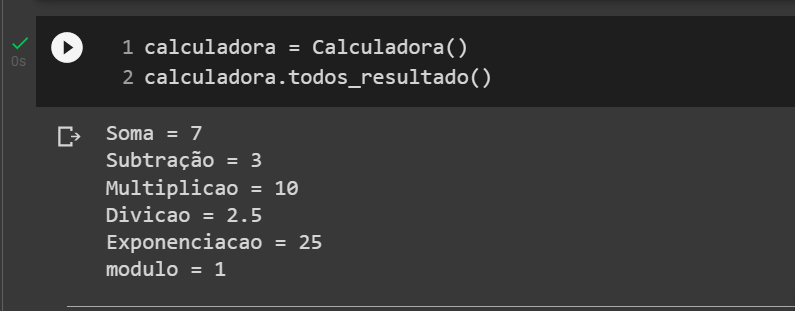
    print(f"Módulo = {self.modulo()}")

**IMAGEM DO EXERCÍCIO:**









II. Encontre os valores para a variável y, onde y = ax + xb – 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, subistituí-lo pelo número 3**. Realizar o plot dos resultados onde será **x = 5; x = 7 e x = 9**. Para o plot você precisará utilizar a **biblioteca matplotlib** apresentada em aula, colocar **legenda** no gráfico, **alterar a cor** dos gráficos (*linhas ou pontos*), **nomear o eixo x**, **nomear o eixo y**:

import numpy as np

# Meu RU: 3539252

a=2; b=5; c=2

x = 5

y1 = a\*x+b\*x-c

x = 7

y2 = a\*x+b\*x-c

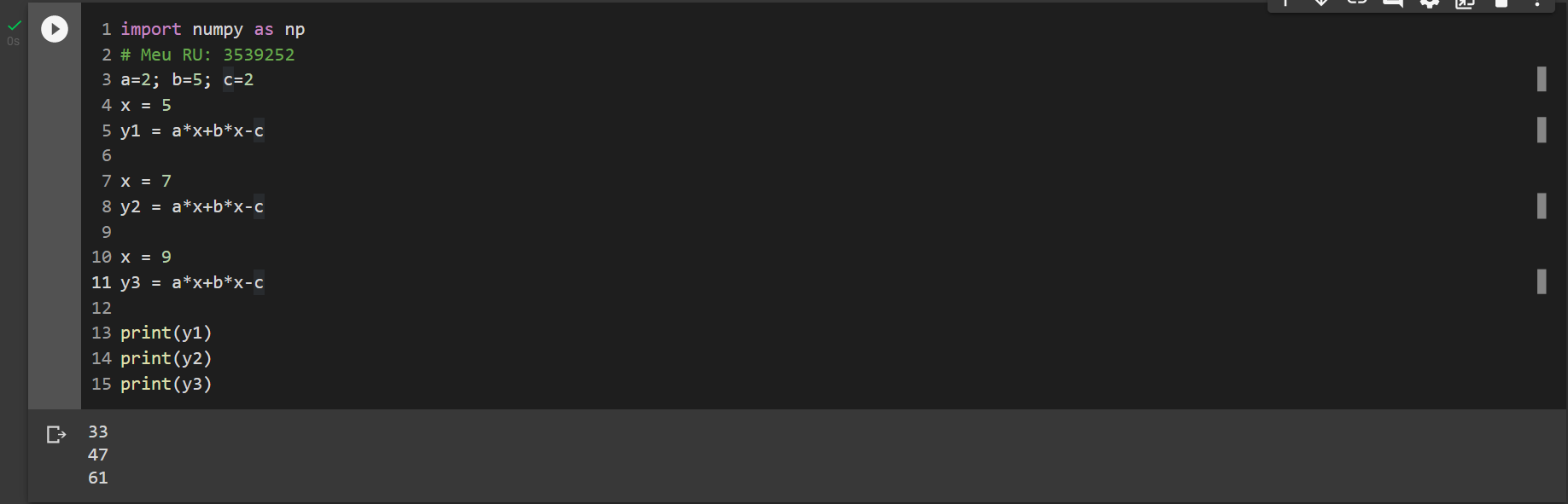
x = 9

y3 = a\*x+b\*x-c

print(y1)

print(y2)

print(y3)



Eixo\_X = [5, 7, 9]

Eixo\_Y = [y1, y2, y3]

import matplotlib.pyplot as plt

import seaborn as sns

plt.style.use('ggplot')

plt.figure(figsize=(7,5))

sns.set\_style('darkgrid')

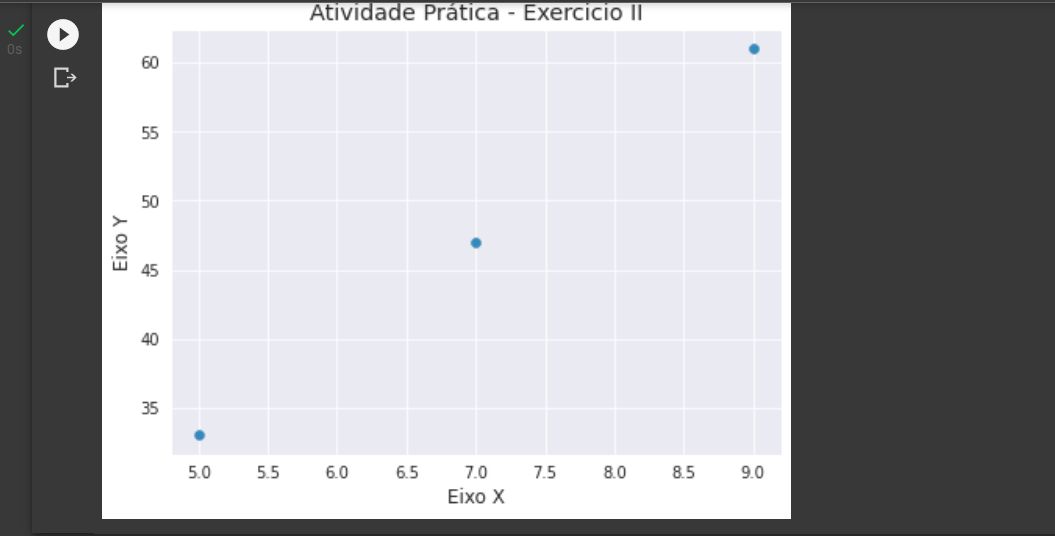
sns.scatterplot(x=Eixo\_X, y=Eixo\_Y)

plt.scatter(x=Eixo\_X, y=Eixo\_Y)

plt.title('Atividade Prática - Exercicio II')

plt.xlabel('Eixo X')

plt.ylabel('Eixo Y')

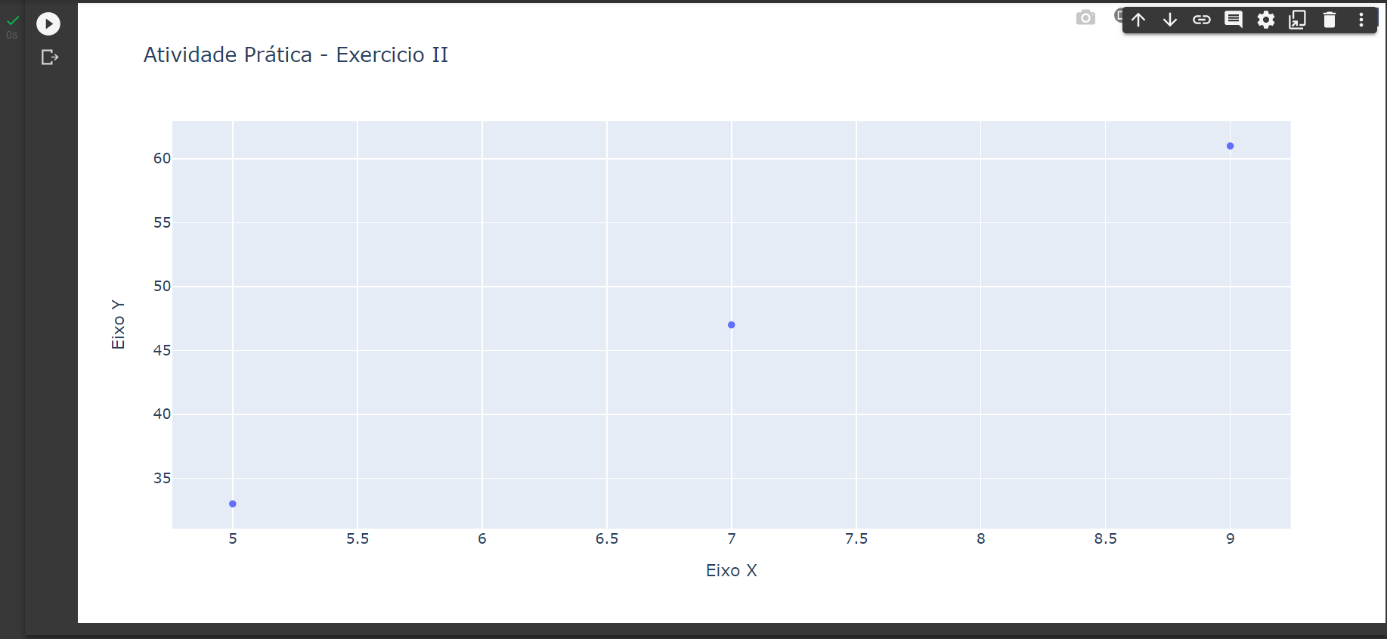


import plotly.express as px

fig = px.scatter(x=Eixo\_X, y=Eixo\_Y,

 title='Atividade Prática - Exercicio II',

 labels={"x": "Eixo X", "y": "Eixo Y"})

fig.show()

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".**

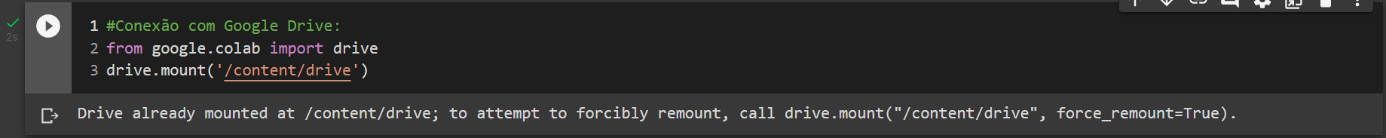
Algumas informações extras sobre a tabela do arquivo **STORES.csv**:

* *ID da loja: (Índice) ID da loja específica.*
* *Store ID: Área Física da loja em pátio.*
* *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.*

#Conexão com Google Drive:

from google.colab import drive

drive.mount('/content/drive')

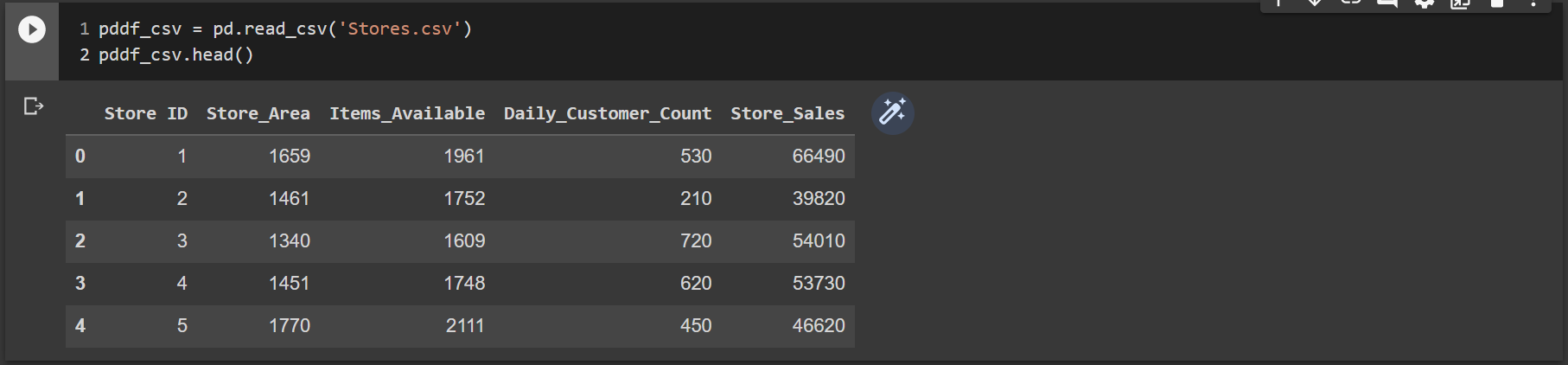
!cp /content/drive/MyDrive/Atividade\_Pratica\_LP/\*.\* /content

arquivo = open('Stores.csv')

conteudo = arquivo.read()

import pandas as pd

pddf\_csv = pd.read\_csv('Stores.csv')

pddf\_csv.head()

#Editando o título das colunas

pddf\_csv.rename({'Store ID ':'Índice Loja','Store\_Area':'Área da loja',

                 'Items\_Available':'Itens disponíveis',

                 'Daily\_Customer\_Count':'Visitantes','Store\_Sales':'Vendas'},

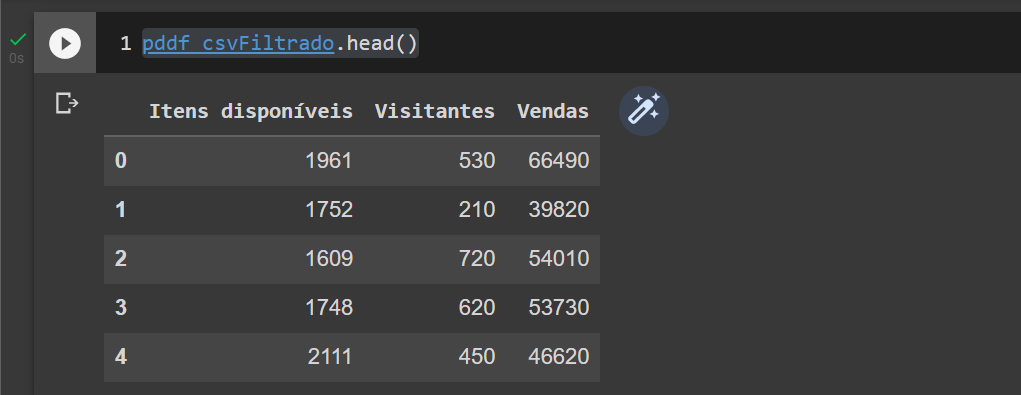
                 axis=1, inplace=True)



pddf\_csvColunas = [ 'Itens disponíveis', 'Visitantes','Vendas']

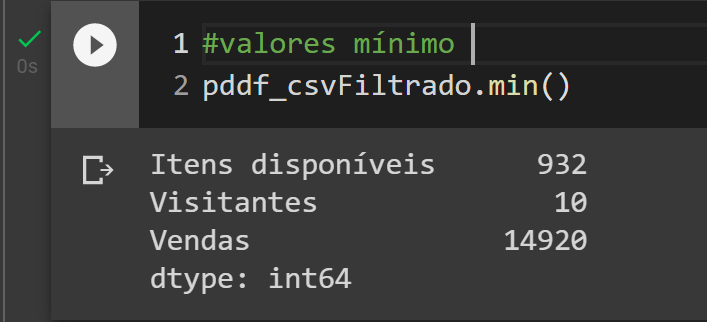
pddf\_csvFiltrado = pddf\_csv.filter(items=pddf\_csvColunas)

pddf\_csvFiltrado.head()



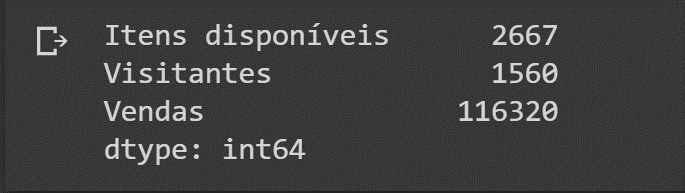
#valores mínimo

pddf\_csvFiltrado.min()



#valores  máximo

pddf\_csvFiltrado.max()



#valores  médio

pddf\_csvFiltrado.mean()



#desvio padrão

pddf\_csvFiltrado.std()

