

ATIVIDADE PRÁTICA LINGUAGEM DE PROGRAMAÇÃO

GUILHERME GIOVANNI VIEIRA LOURENÇO – 3799879
PROF. WELLINGTON RODRIGO MONTEIRO

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(/)**, **exponenciação(^)** e **módulo(%)**. 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 todas as operações matemáticas da calculadora funcionando!

SOLUÇÃO:

```
class calculadora: #criando a classe
    #definindo funções de cada operação
    def __init__(self, a, b) -> None: #método construtor
        self.a = a
        self.b = b
    def soma(self):
        return self.a + self.b
    def subtracao(self):
        return self.a - self.b
    def divisao(self):
        return self.a / self.b
    def multiplicacao(self):
        return self.a * self.b
    def exponenciacao(self):
        return self.a ** self.b
    def modulo(self):
        return self.a % self.b
    #fim da definição das funções
    #início do programa principal
print("Calculadora. Aluno: Guilherme Lourenço, RU: 3799879")
#identificador pessoal
while True: #loop o funcionamento do menu
    try:
        opcao = int(input("Digite a opção desejada:\n1-Somar\n2-Subtrair\n3-Dividir\n4-Multiplicar\n5-Exponenciação\n6-Módulo\n7-Sair"))
        if opcao == 1:
            a = int(input("Digite o primeiro valor"))
            b = int(input("Digite o segundo valor"))
            operacao = calculadora(a,b)
            print(operacao.soma())
        elif opcao == 2:
            a = int(input("Digite o primeiro valor"))
            b = int(input("Digite o segundo valor"))
            operacao = calculadora(a, b)
```

```
        print(operacao.subtracao())
    elif opcao == 3:
        a = int(input("Digite o primeiro valor"))
        b = int(input("Digite o segundo valor"))
        operacao = calculadora(a,b)
        print(operacao.divisao())
    elif opcao == 4:
        a = int(input("Digite o primeiro valor"))
        b = int(input("Digite o segundo valor"))
        operacao = calculadora(a,b)
        print(operacao.multiplicacao())
    elif opcao == 5:
        a = int(input("Digite o primeiro valor"))
        b = int(input("Digite o segundo valor"))
        operacao = calculadora(a, b)
        print(operacao.exponenciacao())
    elif opcao == 6:
        a = int(input("Digite o primeiro valor"))
        b = int(input("Digite o segundo valor"))
        operacao = calculadora(a, b)
        print(operacao.modulo())
    elif opcao == 7:
        print("Encerrando a calculadora...")
        break #encerra o loop e fecha o programa
    else:
        print("Digite uma opção válida")
except ValueError: #tratamento de erro
    print("Você digitou um valor inválido!\nTente novamente.")
```

CONSOLE:

Calculadora. Aluno: Guilherme Lourenço, RU: 3799879

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair1

Digite o primeiro valor7

Digite o segundo valor9

16

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair2

Digite o primeiro valor7

Digite o segundo valor9

-2

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair3

Digite o primeiro valor7

Digite o segundo valor9

0.7777777777777778

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair⁴

Digite o primeiro valor⁷

Digite o segundo valor⁹

63

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair⁵

Digite o primeiro valor⁷

Digite o segundo valor⁹

40353607

Digite a opção desejada:

1-Somar

2-Subtrair

3-Dividir

4-Multiplicar

5-Exponenciação

6-Módulo

7-Sair⁶

Digite o primeiro valor⁷

Digite o segundo valor⁹

7

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 de **x1 = 5; x2 = 7 e x3 = 9**. Para o plot você precisará utilizar a **biblioteca matplotlib** apresentada na Aula Prática 1; **colocar legenda no gráfico, alterar a cor da linha(s) de seu gráfico; nomear o eixo x, nomear o eixo y**.

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

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

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

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

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

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

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

SOLUÇÃO:

```
#importando bibliotecas necessárias
import plotly.express as px
import numpy as np

class equacao: #criando a classe
    def __init__(self, a, b, c) -> None: #método construtor
        self.a = a
        self.b = b
        self.c = c
    def operacaox1(self):
        x = 5
        y = ((self.a * x) + (self.b * x)) - self.c
        return y
    def operacaox2(self):
        x = 7
        y = ((self.a * x) + (self.b * x)) - self.c
        return y
    def operacaox3(self):
        x = 9
        y = ((self.a * x) + (self.b * x)) - self.c
        return y

#programa principal
operacao = equacao(8,7,9) #criando variável a partir da classe "equacao"
tendo como parâmetro os três últimos dígitos do meu RU
#criando vetor de cada resultado das operações usando NumPy
vetor_x1 = np.array([operacao.operacaox1()])
vetor_x2 = np.array([operacao.operacaox2()])
vetor_x3 = np.array([operacao.operacaox3()])
#criando arquivo de texto com os vetores usando NumPy
np.savetxt('x1_resultado.txt', vetor_x1, fmt='%f', delimiter=';')
np.savetxt('x2_resultado.txt', vetor_x2, fmt='%f', delimiter=';')
np.savetxt('x3_resultado.txt', vetor_x3, fmt='%f', delimiter=';')
#carregando arquivo de texto com os vetores usando NumPy
```

```
array_x1 = np.loadtxt('x1_resultado.txt', dtype = np.float64, delimi-
ter=';')
array_x2 = np.loadtxt('x2_resultado.txt', dtype = np.float64, delimi-
ter=';')
array_x3 = np.loadtxt('x3_resultado.txt', dtype = np.float64, delimi-
ter=';')

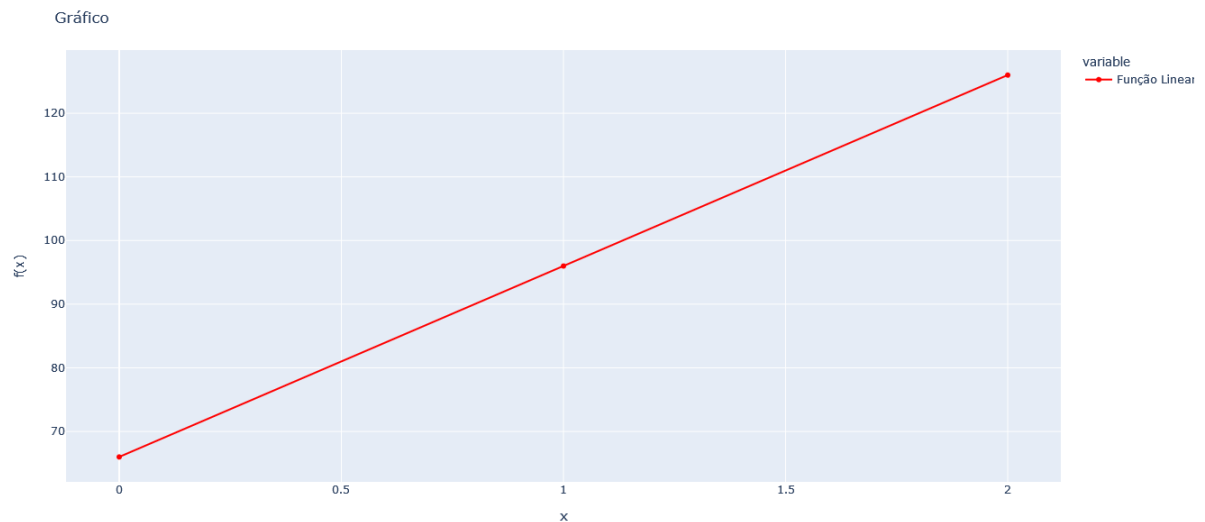
array_x123 = np.vstack([array_x1, array_x2, array_x3]) #juntando os valores
dos outros três arrays em uma única variável para plotar o gráfico
print(array_x123)
fig = px.line(array_x123) #plotando gráfico do tipo linha(line)
fig.update_layout(title = 'Gráfico', xaxis_title = 'x', yaxis_title =
'f(x)') #renomeando o gráfico, eixo x e eixo y
fig.update_traces(name = 'Função Linear', line = dict(color = "red"), mode
= 'lines+markers') #legendando gráfico, alterando a cor e adicionando mar-
cadores
fig.show()
```

CONSOLE:

```
"C:\Users\lauri\Desktop\Faculdade\Linguagem de Programação\Trabalho\venv\Scripts\python.exe" "C:\Users\lauri\Desktop\Faculdade\Linguagem de Programação\Trabalho\EXERCICIO_2.py"
[[ 66.]
 [ 96.]
 [126.]]

Process finished with exit code 0
```

GRÁFICO:



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 seguinte 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.*

SOLUÇÃO:

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

stores = pd.read_csv('Stores.csv', sep = ',') #lendo arquivo csv com pandas
stores_df = pd.DataFrame(stores) #criando um DataFrame
stores_df.rename(columns = {'Store ID' : 'ID_Loja', 'Store_Area' : 'Área',
'Items_Available' : 'Itens_disponíveis', 'Daily_Customer_Count' : 'Visitantes',
'Store_Sales' : 'Vendas(US$)'}, inplace = True) #renomeando colunas
#inicio das análises das colunas
print("Análise Itens disponíveis:")
print("Máximo Itens_disponíveis: {}".format(stores_df['Itens_disponíveis'].max()))
print("Mínimo Itens_disponíveis: {}".format(stores_df['Itens_disponíveis'].min()))
print("Média Itens_disponíveis: {:.2f}".format(stores_df['Itens_disponíveis'].mean()))
print("Desvio padrão Itens_disponíveis: {:.2f}\n".format(stores_df['Itens_disponíveis'].std()))

print("Análise Visitantes:")
print("Máximo Visitantes: {}".format(stores_df['Visitantes'].max()))
print("Mínimo Visitantes: {}".format(stores_df['Visitantes'].min()))
print("Média Visitantes: {:.2f}".format(stores_df['Visitantes'].mean()))
print("Desvio padrão Visitantes: {:.2f}\n".format(stores_df['Visitantes'].std()))

print("Análise Vendas:")
print("Máximo Vendas: {}".format(stores_df['Vendas(US$)'].max()))
print("Mínimo Vendas: {}".format(stores_df['Vendas(US$)'].min()))
print("Média Vendas: {:.2f}".format(stores_df['Vendas(US$)'].mean()))
print("Desvio padrão Vendas: {:.2f}\n".format(stores_df['Vendas(US$)'].std()))
#fim das análises das colunas
```

CONSOLE:


```
Análise Itens disponíveis:  
Máximo Itens_disponíveis: 2667  
Mínimo Itens_disponíveis: 932  
Média Itens_disponíveis: 1782.04  
Desvio padrão Itens_disponíveis: 299.87
```

```
Análise Visitantes:  
Máximo Visitantes: 1560  
Mínimo Visitantes: 10  
Média Visitantes: 786.35  
Desvio padrão Visitantes: 265.39
```

```
Análise Vendas:  
Máximo Vendas: 116320  
Mínimo Vendas: 14920  
Média Vendas: 59351.31  
Desvio padrão Vendas: 17190.74
```

```
Process finished with exit code 0
```