**Tarefa dia 08.05.24**

**Tarefa no 1º. tempo da aula:**

f = open('meu\_arquivo.txt', 'w')

f.write("Teste de número 1\n") # '\n' gera uma quebra de linha

f.write("teste de número 2")

f.close()

f = open('novo\_arquivo.txt', 'w')

f.write('Novos arquivos\nnovos textos')

f.close()

f = open('novo\_arquivo.txt', 'r+')

conteudo = f.read() # conteudo = 'Novos arquivos\nnovos textos'

f.seek(0) # é preciso voltar ao início para ler outra vez

print(f.read())

f.write('\nNunca é demais')

f.close()

f = open('novo\_arquivo.txt', 'r')

print(f.read())

f = open("novo\_arquivo.txt", "r")

for linha in f:

print(linha)

f.seek(0)

for linha in f:

print(linha, end='')

**Tarefa no 2º. tempo da aula:**

def gravar\_nome\_em\_arquivo():

nome = input("Por favor, digite seu nome: ")

# Abrindo o arquivo em modo de escrita ('w')

with open("nome\_usuario.txt", "w") as arquivo:

arquivo.write(nome)

print("Seu nome foi gravado no arquivo com sucesso!")

# Chamando a função para gravar o nome do usuário

gravar\_nome\_em\_arquivo()

def imprimir\_conteudo\_arquivo():

# Solicita ao usuário o nome do arquivo de texto

nome\_arquivo = input("Por favor, digite o nome do arquivo de texto: ")

try:

# Abre o arquivo em modo de leitura ('r')

with open(nome\_arquivo, "r") as arquivo:

# Lê e imprime o conteúdo do arquivo

conteudo = arquivo.read()

print("Conteúdo do arquivo:")

print(conteudo)

except FileNotFoundError:

# Se o arquivo não for encontrado, imprime uma mensagem de erro

print("O arquivo especificado não foi encontrado.")

# Chama a função para imprimir o conteúdo do arquivo

imprimir\_conteudo\_arquivo()

def copiar\_conteudo\_arquivo():

# Solicita ao usuário o nome do arquivo original

nome\_arquivo\_original = input("Digite o nome do arquivo de origem: ")

try:

# Abre o arquivo original em modo de leitura ('r')

with open(nome\_arquivo\_original, "r") as arquivo\_original:

# Lê o conteúdo do arquivo original

conteudo = arquivo\_original.read()

# Solicita ao usuário o nome do novo arquivo

nome\_novo\_arquivo = input("Digite o nome do novo arquivo: ")

# Abre o novo arquivo em modo de escrita ('w')

with open(nome\_novo\_arquivo, "w") as novo\_arquivo:

# Escreve o conteúdo do arquivo original no novo arquivo

novo\_arquivo.write(conteudo)

print("Conteúdo do arquivo copiado com sucesso para o novo arquivo.")

except FileNotFoundError:

# Se o arquivo original não for encontrado, imprime uma mensagem de erro

print("O arquivo especificado não foi encontrado.")

# Chama a função para copiar o conteúdo do arquivo

copiar\_conteudo\_arquivo()

def encontrar\_nome\_por\_numero():

# Solicita ao usuário um número

numero = input("Digite um número para encontrar o nome correspondente: ")

try:

# Abre o arquivo em modo de leitura ('r')

with open("nomes.txt", "r") as arquivo:

# Lê todas as linhas do arquivo

linhas = arquivo.readlines()

# Percorre cada linha para procurar o número

encontrado = False

for linha in linhas:

# Separa o número e o nome da linha

partes = linha.strip().split(":")

if partes[0] == numero:

# Se o número for encontrado, imprime o nome correspondente

print("O nome correspondente ao número", numero, "é:", partes[1])

encontrado = True

break

# Se o número não for encontrado, exibe uma mensagem

if not encontrado:

print("Número não encontrado no arquivo.")

except FileNotFoundError:

# Se o arquivo não for encontrado, imprime uma mensagem de erro

print("O arquivo de nomes não foi encontrado.")

# Chama a função para encontrar o nome correspondente ao número

encontrar\_nome\_por\_numero()

def encontrar\_nome\_por\_numero():

try:

# Solicita ao usuário um número

numero = input("Digite um número para encontrar o nome correspondente: ")

# Abre o arquivo em modo de leitura ('r')

with open("exemplo.txt", "r") as arquivo:

# Lê todas as linhas do arquivo

linhas = arquivo.readlines()

# Percorre cada linha para procurar o número

encontrado = False

for linha in linhas:

# Separa o número e o nome da linha

partes = linha.strip().split(":")

if partes[0] == numero:

# Se o número for encontrado, imprime o nome correspondente

print("O nome correspondente ao número", numero, "é:", partes[1])

encontrado = True

break

# Se o número não for encontrado, exibe uma mensagem

if not encontrado:

print("Número não encontrado no arquivo.")

except FileNotFoundError:

# Se o arquivo não for encontrado, imprime uma mensagem de erro

print("O arquivo de exemplo não foi encontrado.")

# Chama a função para encontrar o nome correspondente ao número

encontrar\_nome\_por\_numero()

