

INSTITUTO FEDERAL DE SANTA CATARINA - IFSC  
CURSO DE ENGENHARIA ELÉTRICA

ALUNO  
ELIAS ANZINI JUNIOR

PROFESSOR  
STEFANO ROMEU ZEPLIN

**PROCESSAMENTO DIGITAL DE IMAGENS**

JOINVILLE, 2022

## SUMÁRIO

1.	ATIVIDADE 1.....	3
1.1	PROGRAMA.....	3
1.2	RESULTADO OBTIDO.....	4
2.	ATIVIDADE 2.....	6
2.1	PROGRAMA.....	6
2.2	RESULTADO OBTIDO.....	7
3.	ATIVIDADE 3.....	8
3.1	PROGRAMA.....	8
3.2	RESULTADO OBTIDO.....	11
3.2.1	TESTE1 .....	11
3.2.2	TESTE2b .....	11
3.2.3	TESTE3 .....	12
3.2.4	TESTE4 .....	13
4.	ATIVIDADE 4.....	14
4.1	PROGRAMA.....	14
4.2	RESULTADO OBTIDO.....	15

## 1. ATIVIDADE 1

### 1.1 PROGRAMA

```
import matplotlib.pyplot as plt

## LISTA INICIAL QUE ATRAVÉS DELA SERÁ GERADA ListaA e ListaB
lista = [50, 20, 30, 20, 50, 60, 30, 80, 10, 100]
listaA = []
listaB = []
posicao = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
x = 0
pos = 0
maior = 0
menor = 0
cont = 0
contlista = 10

## LOOP DO PROGRAMA INFINITO
while True:

    print('''
    Operações

    [ 1 ] Visualizar Gráfico
    [ 2 ] Acrescentar Dados
    [ 3 ] Finalizar o programa
    ''')

    opção = int(input('Qual você deseja? '))

    ## SE O USUÁRIO APERTAR 1 ELE VIZUALIZA O GRÁFICO NO ESTADO INICIAL
    ## O VALOR DA Lista É ELEVADO AO QUADRADO E ADICIONADO AO FINAL DA ListaA
    ## A ListaB É ORGANIZADA EM ORDEM ATRAVÉS DO COMANDO SORTED

    if opção == 1:
        while x < contlista:
            a = lista[x]
            listaA.append(a**2)
            x += 1
        listaB = lista.copy()
        listaB = sorted(listaB)

    ## COMANDOS DE PLOT DO GRÁFICO
    plt.subplot(2, 2, 1)
    plt.plot(posicao, lista, label='Lista Principal')
    plt.legend()
    plt.grid(True)

    plt.subplot(2, 2, 2)
    plt.plot(posicao, listaA, label='ListaA')
    plt.legend()
    plt.grid(True)

    plt.subplot(2, 2, 3)
    plt.plot(posicao, listaB, label='ListaB')
    plt.legend()
    plt.grid(True)
    plt.show()
```

```

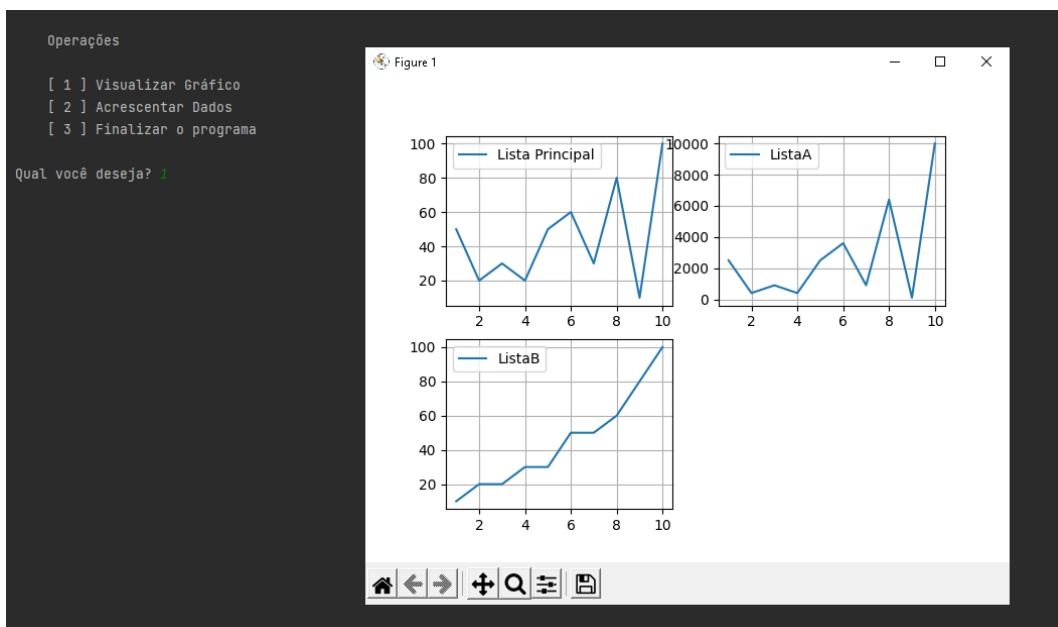
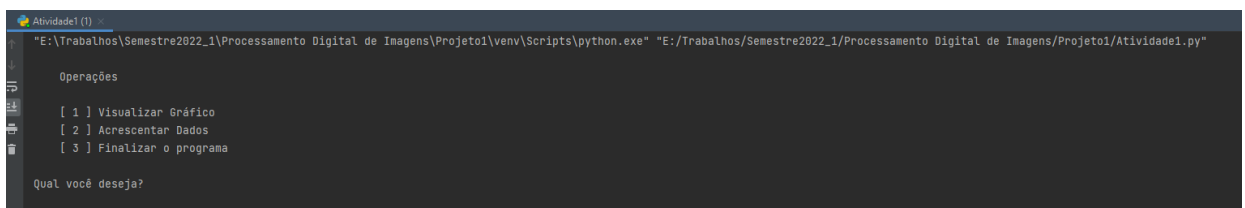
## SE O USUÁRIO APERTAR 2 ELE PODE ACRESCENTAR DADOS
## SÃO UTILIZADAS CONDICIONAIS PARA ELE RETORNAR A PERGUNTA CASO PREENCHA
COM ALGO FORA DA CONDIÇÃO

if opção == 2:
    perg = int(input('Quantos numeros deseja adicionar? '))
    while cont < perg:
        n = int(input(f'Digite o {cont+1}º numero: '))
        while n > 100 or n < 0:
            n = int(input(f'Número Inválido. Digite o {cont + 1}º numero:
'))
        lista.append(n)
        posicao.append(contlista)
        contlista+=1
        cont+=1
    print('Números Adicionados!')

if opção == 3:
    print('\n Programa Finalizado!')
    break

```

## 1.2 RESULTADO OBTIDO



```
Operações

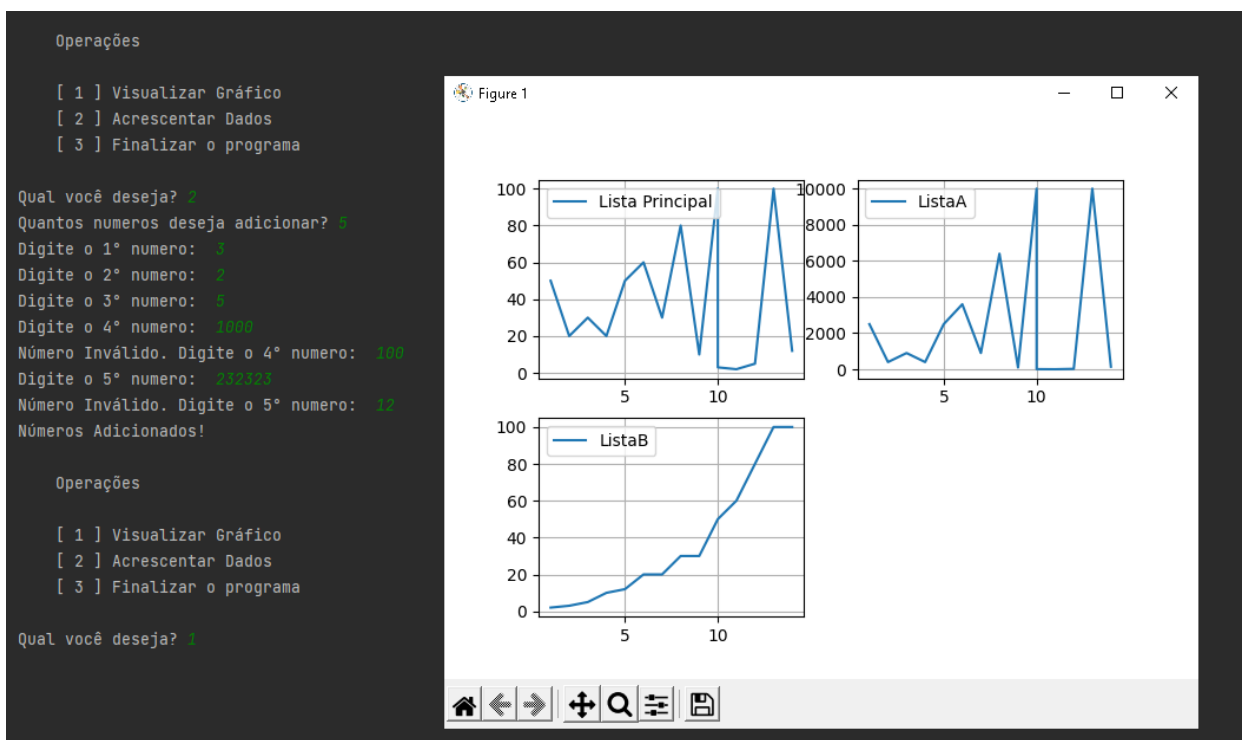
[ 1 ] Visualizar Gráfico
[ 2 ] Acrescentar Dados
[ 3 ] Finalizar o programa

Qual você deseja? 2
Quantos numeros deseja adicionar? 5
Digite o 1° numero: 3
Digite o 2° numero: 2
Digite o 3° numero: 5
Digite o 4° numero: 1000
Número Inválido. Digite o 4° numero: 100
Digite o 5° numero: 232323
Número Inválido. Digite o 5° numero: 12
Números Adicionados!

Operações

[ 1 ] Visualizar Gráfico
[ 2 ] Acrescentar Dados
[ 3 ] Finalizar o programa

Qual você deseja? |
```



## 2. ATIVIDADE 2

### 2.1 PROGRAMA

```
from random import randint
import matplotlib.pyplot as plt
from matplotlib import animation

## LISTA COM OS VALORES DE MEDIÇÃO SORTEADOS

valores = []

## POSIÇÃO DAS MEDIÇÕES

posicao = []
i = 0

## ENTRADA DO VALOR DA TEMPERATURA NOMINAL
## EXISTE O LOOP QUE RETORNA A PERGUNTA CASO AS CONDIÇÕES NÃO SEJAM CUMPRIDAS

setup = input('Qual valor nominal: ')
while (setup.isnumeric() == False) or (int(setup) < 400) or (int(setup) > 900):
    setup = input('ENTRADA INCORRETA: DIGITE NOVAMENTE O VALOR NOMINAL ')

## DEFINIÇÃO DA FIGURA E ADIÇÃO DELA A VARIÁVEL PARA PLOTAGEM
setup = int(setup)
fig = plt.figure()
ax = fig.add_subplot(1,1,1)

## FUNÇÃO NECESSÁRIA PARA USAR A FUNÇÃO DE ANIMAÇÃO
## NA FUNÇÃO É UTILIZADO O COMANDO RANDINT PARA SORTEIO
## EXISTEM AS CONDICIONAIS PARA SEREM EXIBIDAS DEPENDENDO DA TEMPERATURA
def Monitoramento(i):

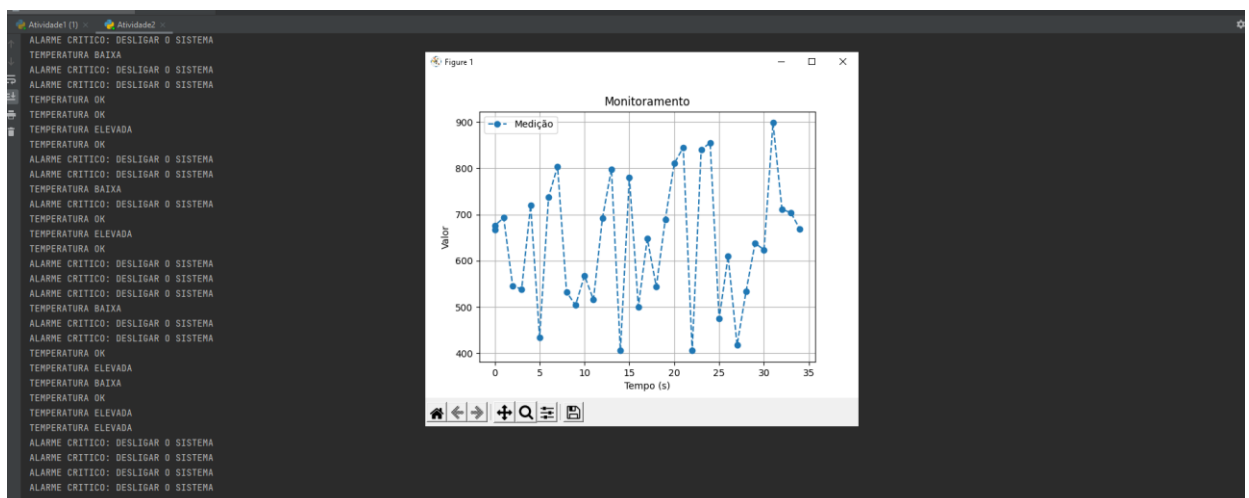
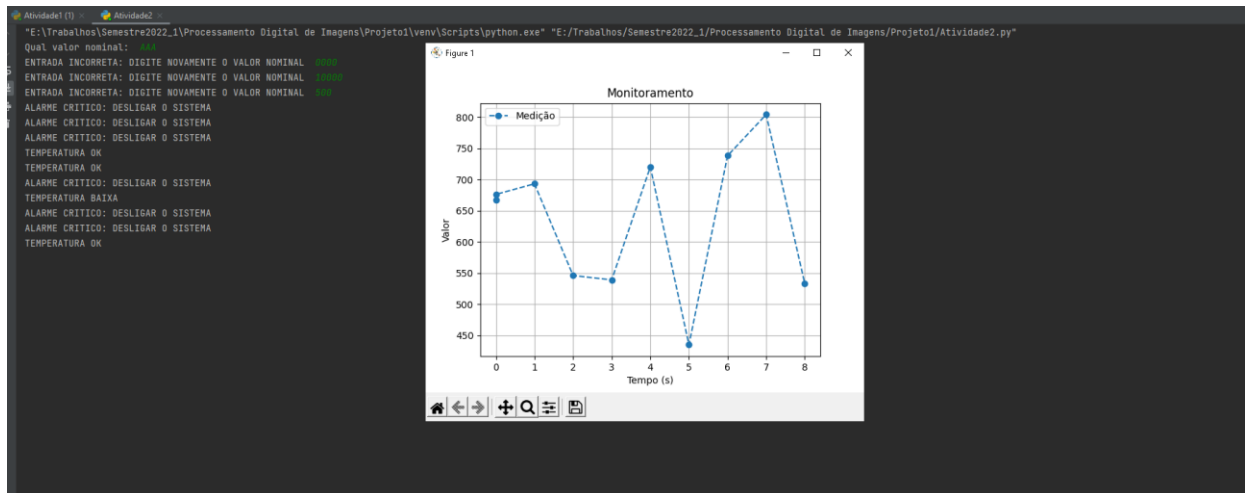
    x = randint(400, 900)
    if x > (setup*1.3):
        print('ALARME CRITICO: DESLIGAR O SISTEMA')
    elif x <= (setup*1.3) and x > (setup*1.1):
        print('TEMPERATURA ELEVADA')
    elif x <= (setup*1.1) and x > (setup*0.9):
        print('TEMPERATURA OK')
    elif x <= (setup*0.9) and x > (setup*0.7):
        print('TEMPERATURA BAIXA')
    elif x < (setup*0.7):
        print('TEMPERATURA NEGATIVA CRÍTICA')

## COMANDOS PARA PLOTAGEM DO GRÁFICO
posicao.append(i)
valores.append(x)
ax.clear()
ax.plot(posicao, valores, "--o", label="Medição")
plt.xlabel("Tempo (s)")
plt.ylabel("Valor")
plt.legend()
plt.grid(True)
plt.title("Monitoramento")

## FUNÇÃO DE ANIMAÇÃO PARA DEIXAR O GRÁFICO EM TEMPO REAL
```

```
a = animation.FuncAnimation(fig, Monitoramento, interval=1000)
i += 1
plt.show()
```

## 2.2 RESULTADO OBTIDO



### 3. ATIVIDADE 3

#### 3.1 PROGRAMA

```
matriz1 = []
matriz2 = []
matriz3 = []
linha = []
conta0 = 0
conta1 = 0
bloco1 = 0
bloco2 = 0
bloco3 = 0

##### CRIAÇÃO DAS MATRIZES #####

print('''
PROGRAMA DE INDICAÇÃO DE DIREÇÃO DO ROBÔ
''')

while True:    #UTILIZADO PARA TRATAMENTO DE ERRO
    try:
        opcao = input('DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: ')
        with open(f'{opcao}.txt', 'r') as f:
            for valor in f:
                colunas = int((len(valor) - 1) / 3)    ## VAI DIVIDIR AS
COLUNAS POR 3 E PEGAR A PRIMEIRA PARTE
                x = valor[0:colunas]
                linha = []
                for j in x:
                    linha.append(j)
                matriz1.append(linha)
            break
        except:
            print('NOME INVÁLIDO! ')

    with open(f'{opcao}.txt', 'r') as g:

        # Separação do segundo bloco
        for valor in g:
            colunas = int((len(valor) - 1) / 3)    ## VAI DIVIDIR AS COLUNAS POR 3
E PEGAR A SEGUNDA PARTE
            x = valor[colunas:colunas*2]
            linha = []
            for j in x:
                linha.append(j)
            matriz2.append(linha)

    with open(f'{opcao}.txt', 'r') as h:

        # Separação do terceiro bloco
        for valor in h:
            colunas = int((len(valor) - 1) / 3)    ## VAI DIVIDIR AS COLUNAS POR 3
E PEGAR A TERCEIRA PARTE
            x = valor[colunas*2:colunas*3]
            linha = []
            for j in x:
                linha.append(j)
            matriz3.append(linha)
```



```

##### INTERPRETAÇÃO DAS MATRIZES #####

#### SENSOR 1 - BLOCO 1 ####

print(f'\nREALIZANDO O TESTE DO ARQUIVO {opcao}...\n')

for linha in matriz1:
    for x in linha:
        if x == '0':
            conta0+=1    # PONTOS PRETOS
        if x == '1':
            conta1+=1    # PONTOS BRANCOS

if conta0 > conta1:

    if (conta1/(colunas*6)) < 0.3:    #DIVIDE O NUMERO DE PONTOS BRANCOS PELO
NÚMERO TOTAL DO BLOCO
        print('NÍVEL LÓGICO BLOCO 1: 0')
        bloco1 = 0
    else:
        #SE O VALOR DE SUJEIRA NO BLOCO FOR
        MAIOR OU IGUAL A 30%
        print('NÍVEL LÓGICO BLOCO 1: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco1 = 2

if conta1 > conta0:
    if (conta0/(colunas*6)) < 0.3:    #DIVIDE O NUMERO DE PONTOS PRETOS PELO
NÚMERO TOTAL DO BLOCO
        print('NÍVEL LÓGICO BLOCO 1: 1')
        bloco1 = 1
    else:
        print('NÍVEL LÓGICO BLOCO 1: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco1 = 2

if conta1 == conta0:
    print('NÍVEL LÓGICO BLOCO 1: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
    bloco1 = 2

conta0 = conta1 = 0

#### SENSOR 2 - BLOCO 2 ####

for linha in matriz2:
    for x in linha:
        if x == '0':
            conta0+=1    # PONTOS PRETOS
        if x == '1':
            conta1+=1    # PONTOS BRANCOS

if conta0 > conta1:
    if (conta1/(colunas*6)) < 0.3:
        print('NÍVEL LÓGICO BLOCO 2: 0')
        bloco2 = 0
    else:
        print('NÍVEL LÓGICO BLOCO 2: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco2 = 2

if conta1 > conta0:
    if (conta0/(colunas*6)) < 0.3:
        print('NÍVEL LÓGICO BLOCO 2: 1')
        bloco2 = 1
    else:
        print('NÍVEL LÓGICO BLOCO 2: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco2 = 2

```

```

if conta1 == conta0:
    print('NÍVEL LÓGICO BLOCO 2: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
    bloco2 = 2

conta0 = conta1 = 0

#### SENSOR 3 - BLOCO 3 ####

for linha in matriz3:
    for x in linha:
        if x == '0':
            conta0+=1    # PONTOS PRETOS
        if x == '1':
            conta1+=1    # PONTOS BRANCOS

if conta0 > conta1:
    if (conta1/(colunas*6)) < 0.3:
        print('NÍVEL LÓGICO BLOCO 3: 0')
        bloco3 = 0
    else:
        print('NÍVEL LÓGICO BLOCO 3: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco3 = 2

if conta1 > conta0:
    if (conta0/(colunas*6)) < 0.3:
        print('NÍVEL LÓGICO BLOCO 3: 1')
        bloco3 = 1
    else:
        print('NÍVEL LÓGICO BLOCO 3: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
        bloco3 = 2

if conta1 == conta0:
    print('NÍVEL LÓGICO BLOCO 3: ERRO NA CONVERSÃO. AJUSTE A IMAGEM ')
    bloco3 = 2

##### RESULTADO DA DIREÇÃO #####

print("")

if bloco1 == 2 or bloco2 == 2 or bloco3 == 2:
    print('COMANDO: ERRO')

if bloco1 == 0 and bloco2 == 0 and bloco3 == 0:
    print('COMANDO: PARE')

if bloco1 == 0 and bloco2 == 0 and bloco3 == 1:
    print('COMANDO: DIREITA')

if bloco1 == 0 and bloco2 == 1 and bloco3 == 0:
    print('COMANDO: FRENTE')

if bloco1 == 0 and bloco2 == 1 and bloco3 == 1:
    print('COMANDO: FRENTE')

if bloco1 == 1 and bloco2 == 0 and bloco3 == 0:
    print('COMANDO: ESQUERDA')

if bloco1 == 1 and bloco2 == 0 and bloco3 == 1:
    print('COMANDO: FRENTE')

if bloco1 == 1 and bloco2 == 1 and bloco3 == 0:
    print('COMANDO: FRENTE')

```

```
if bloco1 == 1 and bloco2 == 1 and bloco3 == 1:
    print('COMANDO: PARE')
```

## 3.2 RESULTADO OBTIDO

### 3.2.1 TESTE1

Os arquivos foram divididos em 3 blocos que representam os sensores. Esses blocos recebem os valores nas matrizes 1, 2 e 3 e é realizado a contagem. Cada matriz possui 6 colunas.

```
PROGRAMA DE INDICAÇÃO DE DIREÇÃO DO ROBÔ
DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste1
NOME INVÁLIDO!
DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste1
NOME INVÁLIDO!
DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste1
REALIZANDO O TESTE DO ARQUIVO teste1...

NÍVEL LÓGICO BLOCO 1: 0
NÍVEL LÓGICO BLOCO 2: 1
NÍVEL LÓGICO BLOCO 3: 0

COMANDO: FRENTE
Process finished with exit code 0
```

teste1 - Bloco de Notas

Arquivo	Editar	Formatar	Exibir	Ajuda
000000	111111	000000		
000000	111111	000000		
000000	111111	000000		
000000	111111	000000		
000000	111111	000000		
000000	111111	000000		

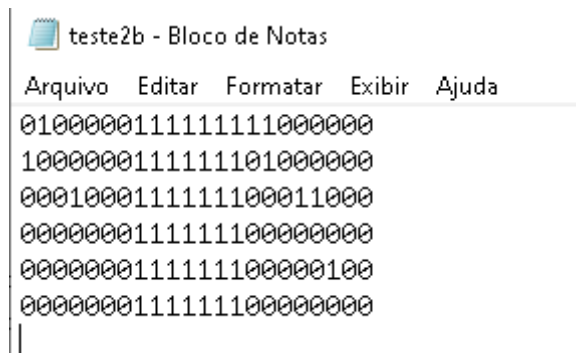
### 3.2.2 TESTE2b

Os arquivos foram divididos em 3 blocos. Desta vez o arquivo possui 23 colunas, o que gerou um pouco de bug no programa. Entretanto ele foi corrigido dividindo as colunas em 3

```
PROGRAMA DE INDICAÇÃO DE DIREÇÃO DO ROBÔ
DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste2b
REALIZANDO O TESTE DO ARQUIVO teste2b...

NÍVEL LÓGICO BLOCO 1: 0
NÍVEL LÓGICO BLOCO 2: 1
NÍVEL LÓGICO BLOCO 3: 0

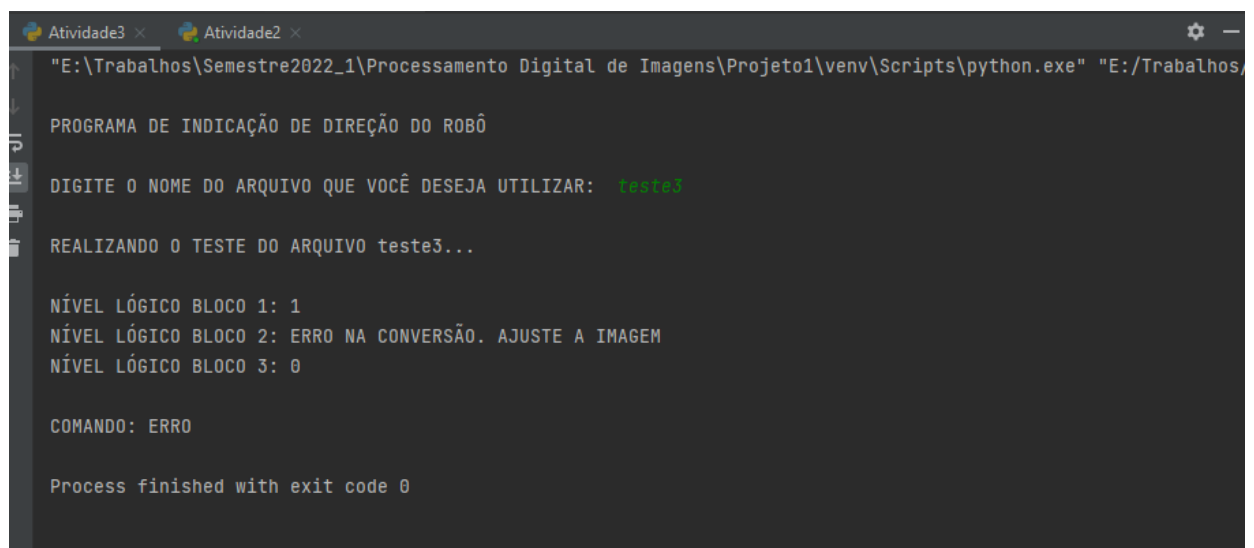
COMANDO: FRENTE
Process finished with exit code 0
```



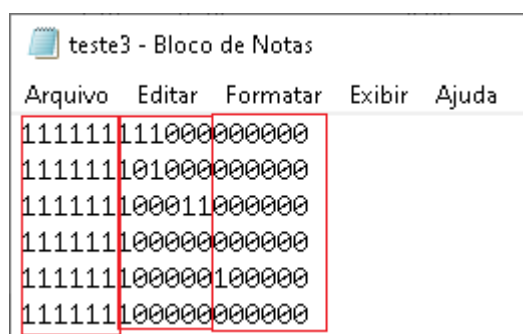
```
teste2b - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
010000001111111111000000
10000000111111101000000
00010001111111100011000
00000001111111100000000
00000001111111100000100
00000001111111100000000
00000001111111100000000
```

### 3.2.3 TESTE3

No segundo bloco possuí mais que 30% de números “1”. Por esse motivo o código entrou na condicional de erro de conversão.



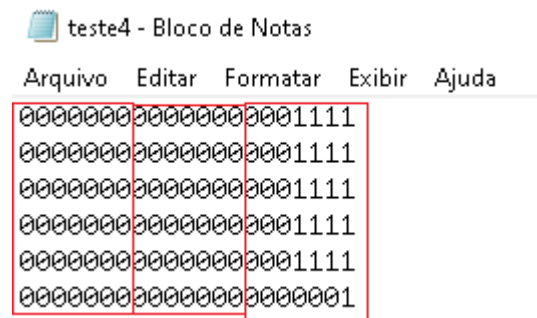
```
Atividade3 x Atividade2 x
"E:\Trabalhos\Semestre2022_1\Processamento Digital de Imagens\Projeto1\venv\Scripts\python.exe" "E:/Trabalhos/
PROGRAMA DE INDICAÇÃO DE DIREÇÃO DO ROBÔ
DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste3
REALIZANDO O TESTE DO ARQUIVO teste3...
NÍVEL LÓGICO BLOCO 1: 1
NÍVEL LÓGICO BLOCO 2: ERRO NA CONVERSÃO. AJUSTE A IMAGEM
NÍVEL LÓGICO BLOCO 3: 0
COMANDO: ERRO
Process finished with exit code 0
```



```
teste3 - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
11111111111000000000000
11111111010000000000000
11111111000111000000000
11111111000000000000000
11111111000000000000000
11111111000000100000000
11111111000000000000000
```

### 3.2.4 TESTE4

O teste4 tem 21 colunas. Deste modo cada bloco foi dividido por  $n^{\circ}\text{colunas}/3 = 7$  colunas. Ocorreu erro no bloco 3 pois existe muita sujeira.



```
Atividade3 x Atividade2 x
"E:\Trabalhos\Semestre2022_1\Processamento Digital de Imagens\Projeto
PROGRAMA DE INDICAÇÃO DE DIREÇÃO DO ROBÔ

DIGITE O NOME DO ARQUIVO QUE VOCÊ DESEJA UTILIZAR: teste4

REALIZANDO O TESTE DO ARQUIVO teste4...

NÍVEL LÓGICO BLOCO 1: 0
NÍVEL LÓGICO BLOCO 2: 0
NÍVEL LÓGICO BLOCO 3: ERRO NA CONVERSÃO. AJUSTE A IMAGEM

COMANDO: ERRO

Process finished with exit code 0
```

## 4. ATIVIDADE 4

### 4.1 PROGRAMA

```
5.
## PEÇAS INICIAIS PARA DESPACHO
pecas = ['Celular', 'Computador', 'Impressora', 'TV', 'Notebook']

## TECNICA INICIAL DE PROCESSAMENTO
tecnica = 'F'

## LOOP INFINITO
while True:

    print('''
        Operações

        [ T ] Selecionar Técnica de Processamento
        [ E ] Entrada de Dados
        [ P ] Processar Peça
        [ S ] Sair do Programa
        ''')

    ## ENTRADA PARA ESCOLHER A OPÇÃO.
    ## CONDICIONAL PARA PODER DIGITAR APENAS AS OPÇÕES EXIBIDAS APLICADA
    opcao = input('Qual você deseja? ')
    while (opcao.isalpha() == False) or opcao not in 'TEPSteps':
        opcao = input('Entrada Incorreta. Digite a opção novamente: ')

    ## SELEÇÃO DA TÉCNICA DE PROCESSAMENTO
    if opcao in 'Tt':

        print('''
            Tecnicas de Processamento:

            [ F ] FIFO
            [ L ] LIFO
            ''')

        tecnica = input('Qual você deseja? ')

        while (tecnica.isalpha() == False) or tecnica not in 'FfLl':
            tecnica = input('Entrada Incorreta. Digite a opção novamente: ')

        if tecnica in 'Ff':
            print('Tecnica FIFO selecionada!')

        if tecnica in 'Ll':
            print('Tecnica LIFO selecionada!')

    ## CÓDIGO PARA ADIÇÃO DE DADOS. DEPENDENDO DA TECNICA SELECIONADA O ITEM É
    ## INSERIDO NO INICIO DA LISTA OU NO FINAL.
    if opcao in 'Ee':
        if tecnica in 'Ff':
            item = input('Qual item deseja incluir?: ')
            pecas.insert(0, item)
            print('FIFO: Item adicionado na primeira posição com
sucesso!')

        if tecnica in 'Ll':
```

```

        item = input('Qual item deseja incluir?: ')
        pecas.append(item)
        print('LIFO: Item adicionado na ultima posição com sucesso!')

## CÓDIGO PARA PROCESSAMENTO DA PEÇA. A PEÇA FINAL OU INICIAL É REMOVIDA
DA LISTA COM O COMANDO POP
    if opcao in 'Pp':

        if len(pecas) != 0:

            if tecnica in 'Ff':
                print(f'Processando {pecas[0]} ... ')
                pecas.pop(0)
                print('Item Despachado Através do Método FIFO!')

            if tecnica in 'Ll':
                print(f'Processando {pecas[len(pecas)-1]} ... ')
                pecas.pop(len(pecas) - 1)
                print('Item Despachado Através do Método LIFO!')

        else:
            print('Nenhuma peça disponível para despacho')

## CÓDIGO PARA FINALIZAÇÃO DO PROGRAMA E EXIBIÇÃO DOS ITENS RESTANTES
    if opcao in 'Ss':
        print('\nFinalizando Programa...')

        print('\nItens restantes para despacho: ')

        for i in range(0, len(pecas), 1):
            print(f'{pecas[i]}')

        print('\nPrograma Finalizado!')
        break

```

## 4.2 RESULTADO OBTIDO

```

## PEÇAS INICIAIS PARA DESPACHO
pecas = ['Celular', 'Computador', 'Impressora', 'TV', 'Notebook']

```

```

"E:\Trabalhos\Semestre2022_1\Processamento Digital de Imagens\Projeto1\venv\Scripts\python.exe" "E:/Trabalhos/Semestre2022_1/Processamento Digital de Imagens/Projeto1/Atividade4.py"

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? █
Processando Celular ...
Item Despachado Através do Método FIFO!

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? █
Processando Computador ...
Item Despachado Através do Método FIFO!

```

```

"E:\Trabalhos\Semestre2022_1\Processamento Digital de Imagens\Projeto1\venv\Scripts\python.exe" "E:/Trab

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? T

Técnicas de Processamento:

[ F ] FIFO
[ L ] LIFO

Qual você deseja? L
Técnica LIFO selecionada!

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? P
Processando Notebook ...
Item Despachado Através do Método LIFO!

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja?

```

```

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? E
Qual item deseja incluir?: PC GAMER
LIFO: Item adicionado na ultima posição com sucesso!

```

```

Operações

[ T ] Selecionar Técnica de Processamento
[ E ] Entrada de Dados
[ P ] Processar Peça
[ S ] Sair do Programa

Qual você deseja? P
Nenhuma peça disponível para despacho

```



### Operações

- [ T ] Selecionar Técnica de Processamento
- [ E ] Entrada de Dados
- [ P ] Processar Peça
- [ S ] Sair do Programa

Qual você deseja? **S**

Finalizando Programa...

Itens restantes para despacho:

Celular

Computador

Impressora

TV

Notebook

Programa Finalizado!