POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Trabalho Prático 2 – Transitórios Eletromagnéticos

Marcella Cavalcanti
Tiago da Silva

Leiria
2021

Sumário

Índice de figuras

Introdução

Este trabalho realiza uma análise de qualidade de energia de dados de tensão e corrente fornecidos pelo docente em duas planilhas excel. A primeira, “dados_1ciclo_G4”, contém somente informações relativas a tensão e corrente elétrica em um período definido de tempo. A segunda, “Dados_audit”, possui diversos dados de tensão, frequência, taxas de harmónicos e compõem a análise de um determinado local. Há ainda auma terceira planilha, “cacebalhos – audit”, que serve como cabeçalho para facilitação na manipulação das informações.

Devido à grande quantidade de dados envolvido nelas, o estudo foi feito por meio de scripts de Python, na tentativa de obter os melhores resultados no menor tempo possível de trabalho. Foram importadas diferentes bibliotecas dessa linguagem, nomeadamente Numpy, Pandas, Matplotlib, Scipy e Statitics.

O trabalho se divide em 3 partes. No primeiro enunciado é pedido que se faça as formas de onda de tensão, de corrente e do espectro de frequência de ambas variáveis relativas à planilha “dados_1ciclo_G4”. Depois que se calcule as taxas de distorção harmónica dessas variavéis e comente os resultados de acordo com a norma EN50160. No segundo enunciado, é pedido que se verifique a conformidade dos dados contidos “Dados_audit” com a mesma norma no que tange a tensão, frequência, taxa de distorção harmónica, peso relativo dos harmónicos individuais e desequilíbrio de tensões. Já no terceiro e último enenunciado é pedido que se analise a distorção harmónica da corrente dessa mesma planilha, tal como o desequilíbrio entre fases.

A realização deste trabalho permitiu um melhor entendimento dos fatores impactantes da qualidade de energia em um local e o conhecimento da norma supracitada que determina os limites e os valores dos principais parâmetros de uma rede elétrica no ponto de entrega ao cliente.

1- Análise do ficheiro dados_1ciclo_G4

A análise dos dados de tensão e corrente referentes à planilha “dados_1ciclo_G4” foi realizada por meio de scripts Python pela maior facilidade de construção do espectro de frequência de ambas variáveis. Além disso, a linguagem possibilita reconstruir os mesmos gráficos caso haja uma mudança nos dados da planilha sem que necessite de alteração dos comandos, desde que mantidas as formatações originais das posições das linhas e das colunas.

Os dados foram extraídos por meio dos seguintes comandos:

```
#inserção de dados
dataset=pd.read_excel('dados_1ciclo_G4.xlsx')
dataset=np.array(dataset)
tempo = dataset[0:4096,0]
tensao = dataset[0:4096,1]
corrente = dataset[0:4096,2]
```

1.1- Representação das formas de ondas da corrente e da tensão

Para representação das formas de onda foram utilizados os comandos abaixo que, além de plotar as variavéis em uma função regida pelo tempo, salva as imagens em arquivos no formato PNG. Foram determinados os nomes de títulos dos gráficos, eixos nas coordenadas x e y. O período de tempo dos sinais propostos é de 20 milisegundos, caracterizando uma frequência fundamental de 50 Hz. Ao primeiro olhar é possível obervar na plotagem uma onda de tensão sinusoidal praticamente perfeita, enquanto a onda de corrente possui distorções bastante relevantes.

```
#plotagem dos dados de corrente
fig1, ax = plt.subplots()
ax.plot(tempo, corrente)
ax.set_title('Sinal de corrente')
ax.set_xlabel('Tempo [s]')
ax.set_ylabel('Corrente [A]')
fig1.savefig('corrente.png')

#plotagem dos dados de tensão
fig2, bx = plt.subplots()
bx.plot(tempo, tensao)
bx.set_title('Sinal de tensão')
bx.set_xlabel('Tempo [s]')
bx.set_ylabel('Tensão [V]')
fig2.savefig('tensao.png')
```

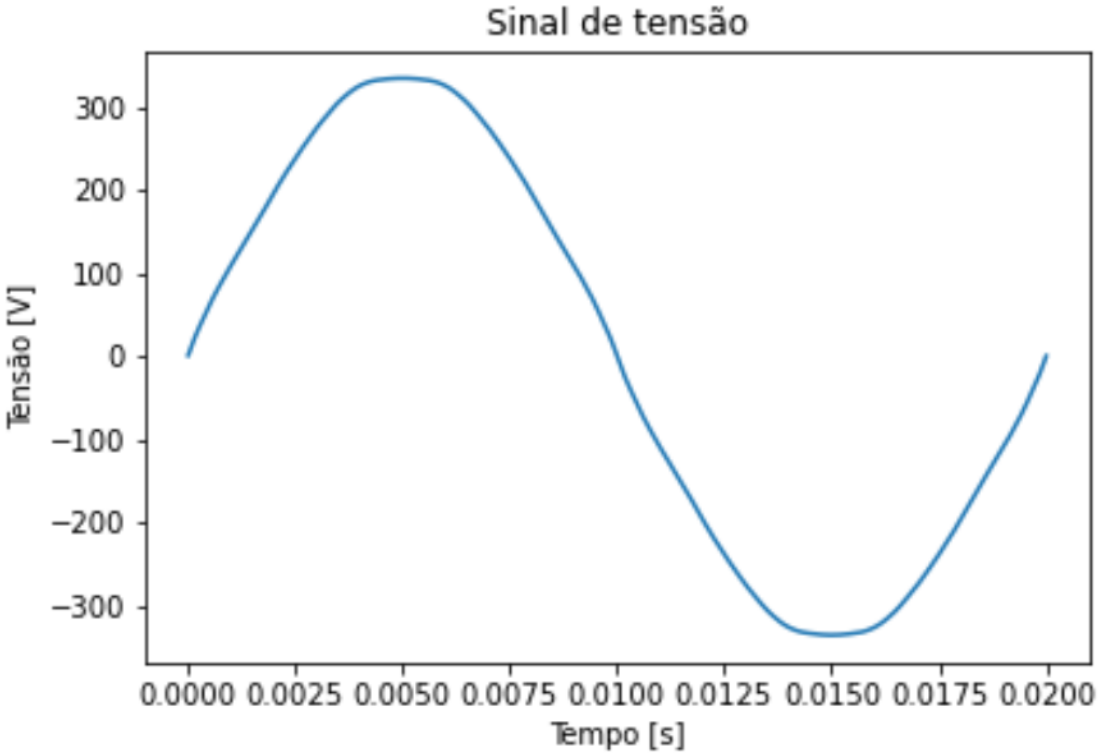

Figura 1: Sinal de tensão

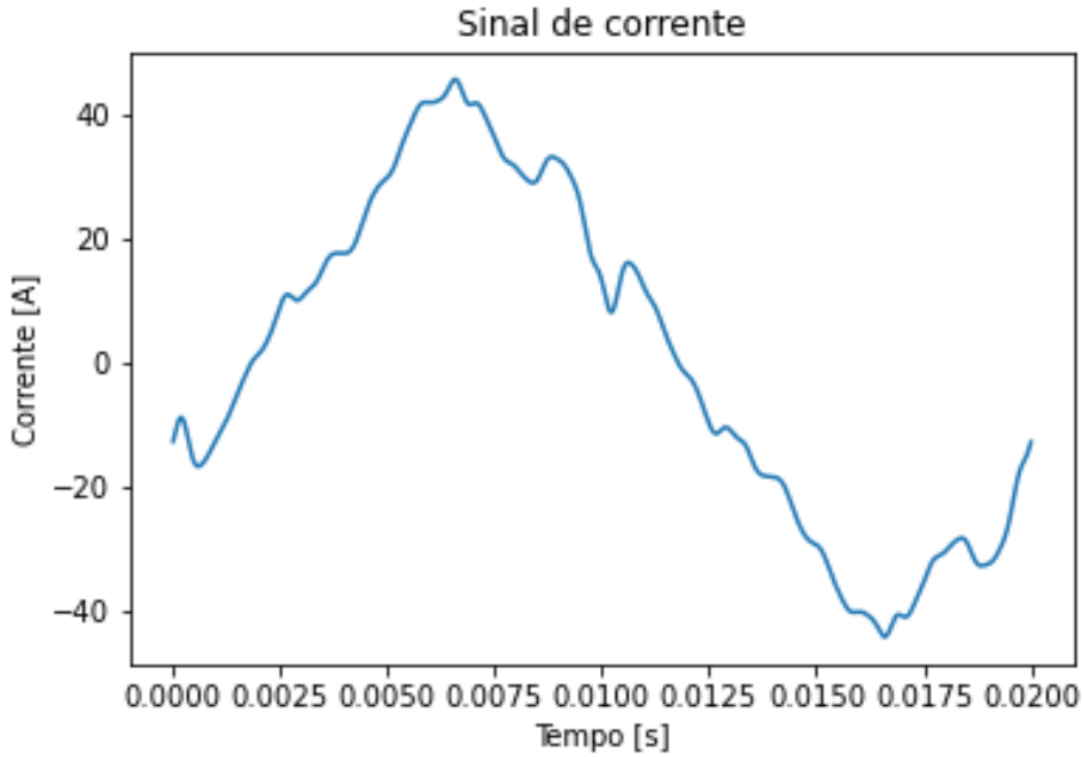

Figura 2: Sinal de corrente

1.2- Espectro de frequência para as grandezas em causa

Para obtenção do espectro de frequência das grandezas de tensão e corrente elétrica foi utilizado o recurso da Tranformada Rápida de Fourier, *Fast Fourier Transform (FFT),* descrita pela fórmula abaixo.

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn}$$

Conforme descrito por Fourier, todo sinal períodico pode ser decomposto em uma série trignómetrica inifinita de senos e cossenos e, por meio do algoritmo FFT é possível converter um sinal descrito no tempo em uma representação no domínio da frequência. É importante notar que, segundo o critério de Nyquist, a mínima frequência observável para o sinal deve ser duas vezes a sua frequência fundamental a fim de se evitar o efeito Aliasing. Dessa forma, como há 4096 dados no período amostral, o algoritmo funciona perfeitamente para as formas de onda em questão. Assim, as linhas de comando para composição dos espectros forma concebidas para plotar ambos os gráficos e salva-los em um formato PNG em uma pasta determinada pelo Framework utilizado no processamento do programa.

```
#obtensão da análise de Fourier
from scipy.fft import fftfreq

N = len(tempo) #número de amostragens
periodo = tempo[-1] #período do sinal
sample_rate = periodo/N #tempo de amostragem
frequencia = fftfreq(N, sample_rate) #extração das frequências harmónicas

#análise de fourier para tensão
fft_calculo1 = np.fft.fft(tensao)
fft_abs1 = 2 * np.abs(fft_calculo1/N) #extração das amplitudes
plt.xticks([0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600])
plt.xlim([0,550])
plt.ylim([0,350])
plt.title('Espectro de frequência do sinal de tensão')
plt.ylabel('Amplitude')
plt.xlabel('Frequência')
plt.bar(frequencia, fft_abs1, width=2)
plt.savefig('Especto de frequência do sinal de tensao.png')
plt.show()

#análise de fourier para corrente
fft_calculo2 = np.fft.fft(corrente)
fft_abs2 = 2 * np.abs(fft_calculo2/N)#extração das amplitudes
plt.xticks([0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600])
plt.yticks([0, 2.5 , 5, 10, 15, 20, 25, 30, 35])
plt.xlim([0,550])
plt.ylim([0,40])
```

```
plt.title('Espectro de frequência do sinal de corrente')
plt.ylabel('Amplitude')
plt.xlabel('Frequência')
plt.bar(frequencia, fft_abs2, width=2)
plt.savefig('Especto de frequência do sinal de corrente.png')
plt.show()
```

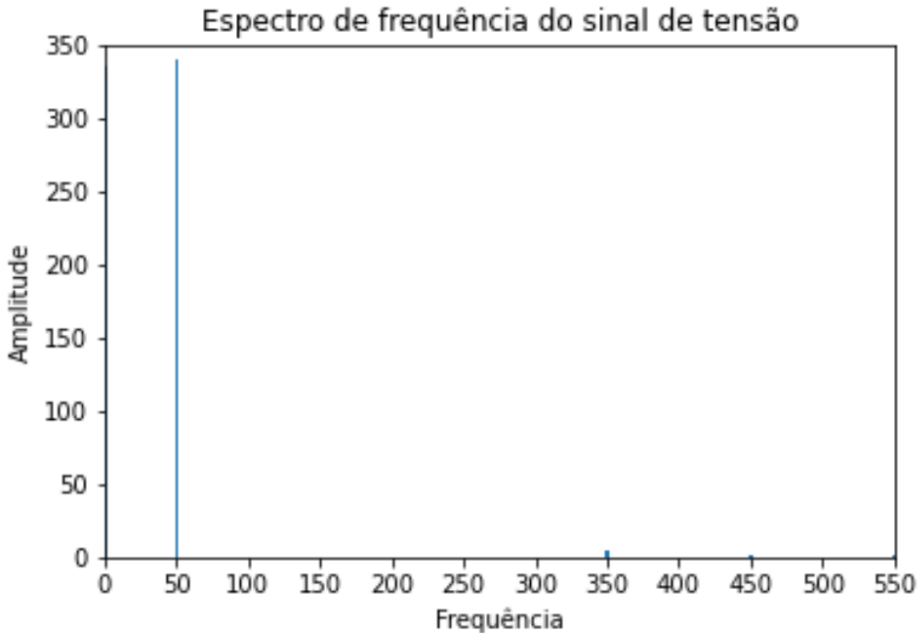

Figura 3: Espectro de frequência do sinal de tensão

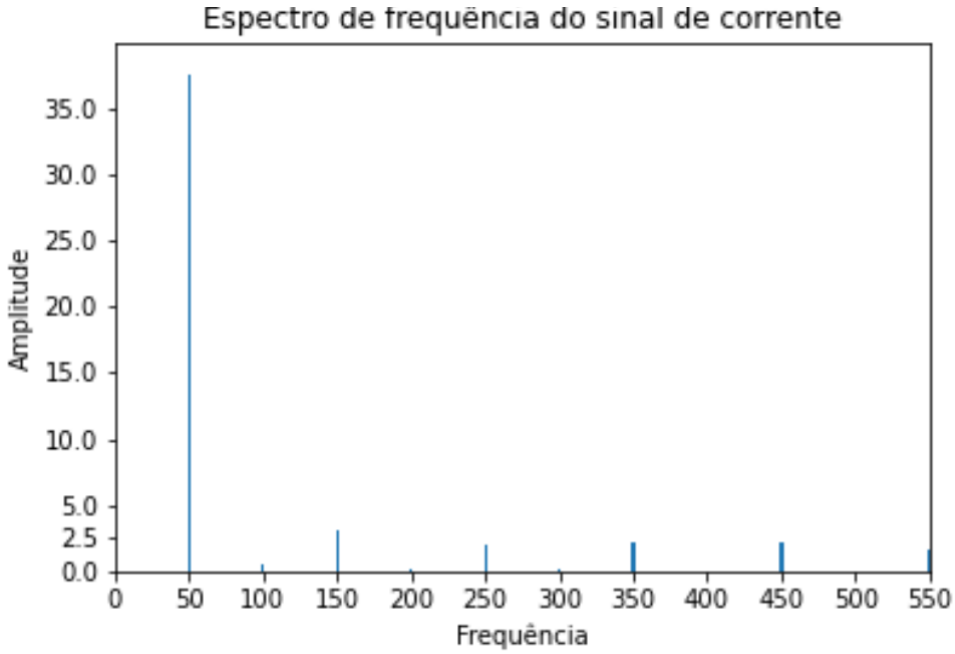

Figura 4: Espectro de frequência do sinal da corrente

Como era esperado, é possível ver no espectro da corrente distorções causadas pela 3°, 5°, 7° e 9° harmónicas que podem comprometer o bom funcionamento do sistema.

1.3- Cálculo das taxas de distorção harmónica das grandezas em causa

O cálculo da taxa de distorção harmónica (THD) é descrito pela fórmula abaixo. Levam-se em conta o peso de cada tensão eficaz de cada harmónico em relação a tensão eficaz da frequência fundamental.

$$\mathrm{THD} = \frac{\sqrt{\sum_{h>1} M_h^2}}{M_1}$$

Dessa forma, para cada amplitude das frequências harmónicas obtida pela FFT, foram extraídos o valor eficaz e inseridos na fórmula de distorção harmónica. Há na concepção das linhas de comando a formação de listas com 11 posições para posterior construção de uma tabela em uma planilha que informe diferentes valores de THD, amplitudes de tensão e corrente para as frequências de harmónicos de 1 a 10, dentre outros parâmetros. Assim, o código e a tabela de parâmetros segue abaixo:

```
#cálculo de DHT para tensão

fft_rms1=(fft_abs1/np.sqrt(2)) #tensão rms de cada frequência
tensao_rms = [11] #para tabela posterior
tensao_rms[0] = np.sqrt(sum(fft_rms1[:4095]**2)) #tensão rms do sinal como um todo
tensao_rms_frequencia_fundamental = [11] #para tabela posterior
tensao_rms_frequencia_fundamental[0] = fft_rms1[1]
tensao_rms_harmonicas = [11] #para tabela posterior
tensao_rms_harmonicas[0] = np.sqrt(sum(fft_rms1[2:4095]**2)) #tensão rms do somatório de
harmónicas
DHT_tensao_porcentagem = [11] #para tabela posterior
DHT_tensao_porcentagem[0] = (tensao_rms_harmonicas / fft_rms1[1]) * 100 #DHT de
harmónicas de tensão

#cálculo de DHT para corrente

fft_rms2=(fft_abs2/np.sqrt(2)) #corrente rms de cada frequêcnia
corrente_rms = [11] #para tabela posterior
corrente_rms[0] = np.sqrt(sum(fft_rms2[:4095]**2)) #corrente rms do sinal como um todo
corrente_rms_frequencia_fundamental = [11] #para tabela posterior
corrente_rms_frequencia_fundamental[0] = fft_rms2[1]
corrente_rms_harmonicas = [11] #para tabela posterior
corrente_rms_harmonicas[0] = np.sqrt(sum(fft_rms2[2:4095]**2)) #corrente rms do
somatório de harmónicas
DHT_corrente_porcentagem = [11] #para tabela posterior
DHT_corrente_porcentagem[0] = (corrente_rms_harmonicas / fft_rms2[1]) * 100 #DHT de
harmónicas de corrente

#criacao de tabela excel contendo dados das duas análises acima
linha = 'Frequência Amplitude_de_tensão Tensão_rms_do_sinal[V]
Tensão_rms_frequência_fundamental[V] Tensão_rms_harmónicos[V] DHT(%)_tensão
```

```
Amplitude_de_corrente Corrente_rms_do_sinal[A] Corrente_rms_frequência_fundamental[A]
Corrente_rms_harmónicos[A] DHT(%)_corrente'.split()
dados = [frequencia[:11], fft_abs1[:11], tensao_rms, tensao_rms_frequencia_fundamental,
tensao_rms_harmonicas, DHT_tensao_porcentagem, fft_abs2[:11], corrente_rms,
corrente_rms_frequencia_fundamental, corrente_rms_harmonicas,
DHT_corrente_porcentagem]
tabela = pd.DataFrame(data = dados, index = linha)
tabela.to_excel('Dados espectro de frequência de tensao e corrente.xls')
```

Frequência	0	50	100	150	200	250	300	350	400	450	500
Amplitude_de_tensão	0,00	338,83	0,14	0,51	0,03	0,75	0,10	4,64	0,07	1,19	0,03
Tensão_rms_do_sinal[V]	239,65										
Tensão_rms_frequência_fundamental[V]	239,59										
Tensão_rms_harmónicos[V]	5,32										
DHT(%)_tensão	[2.21891866]										
Amplitude_de_corrente	0,00	37,49	0,52	3,08	0,12	2,11	0,14	2,13	0,08	2,29	0,04
Corrente_rms_do_sinal[A]	27,08										
Corrente_rms_frequência_fundamental[A]	26,51										
Corrente_rms_harmónicos[A]	5,51										
DHT(%)_corrente	[20.77223146]										

Figura 5: Tabela de dados de tensão e corrente

1.4- Comentários à luz da norma EN 50160

Segundo a norma EN 50160, precisamente no item 2.11, que descreve os parâmetros relativos às tensões harmónicas, *"a distorção harmónica total (THD) da tensão de alimentação (incluindo as harmónicas até a ordem 40) não devem ultrapassar 8%".* Como visto na tabela acima, a DHT da tensão de alimentação é 2,21 %, dentro do limite aceitável. A 7° harmónica possui o maior peso dentre todas as harmónicas citadas, mas nada que ultrapasse o peso relativo de 5% sobre a tensão da frequência fundamental citada pela norma.

Por outro lado, os valores obtidos em relação a corrente são um pouco preocupantes e devem ser levados em consideração, embora a norma supracitada não preconize nada relativavamente aos harmónicos do sinal de corrente. Neste caso, um estudo maior poderia ser feito recorrendo a norma IEC 61000-3-12, *"Limits for harmonic currents produced by equipment connected to public low-voltage systems with input current >16 A and ≤ 75 A per phase*", para análise dessa grandeza, algo que extrapola o enunciado do trabalho.

2- Análise do ficheiro Dados_audit

A análise dos dados referentes à planilha “Dados_audit” em conformidade com a norma EN50160 foi realizada também por meio dos scripts abaixo de Python. Diferentemente da planilha do enunciado anterior, há em “Dados_audit” 1008 dados contidos em cada um dos 943 parâmetros de qualidade de energia de um determinado local. Cada dado foi tomado em uma amostra de 10 minutos durante uma semana e caracterizam a situação da tensão, corrente, frequência, harmónicos e consumo de potências. Nem todos foram utilizados na resolução do enunciado, mas somente aqueles que possuem relevância para os cálculos necessários. Ao contrário do enunciado anterior, a extração da planilha foi feita inteiramente e salva em um data frame listado como dados_audit:

```
import statistics as st
import pandas as pd
import numpy as np

#inserção de dados de auditoria
dados_audit = pd.read_excel('Dados_audit.xlsx')
```

2.1- Análise da tensão RMS

Segundo o item 2.3 da norma EN5016, “*durante cada período de uma semana, 95% dos valores eficazes médios de 10 min devem situar-se na gama de Un +- 10%” e “todos os valores valores eficazes médios de 10 min devem situar-se na gama de Un +10% e -15%”.* Assim, para avaliação, foram tomados todos os valores médios eficazes de tensão de cada fase e submetidos a avaliações condicionais. Para obtenção de 95% dos valores contidos nessas grandezas, utilizou-se a estratégia de distribuição gaussiana, que prevê que aproximadamente 95% dos dados de uma distribuição situam-se dentro de 2 desvios padrões.

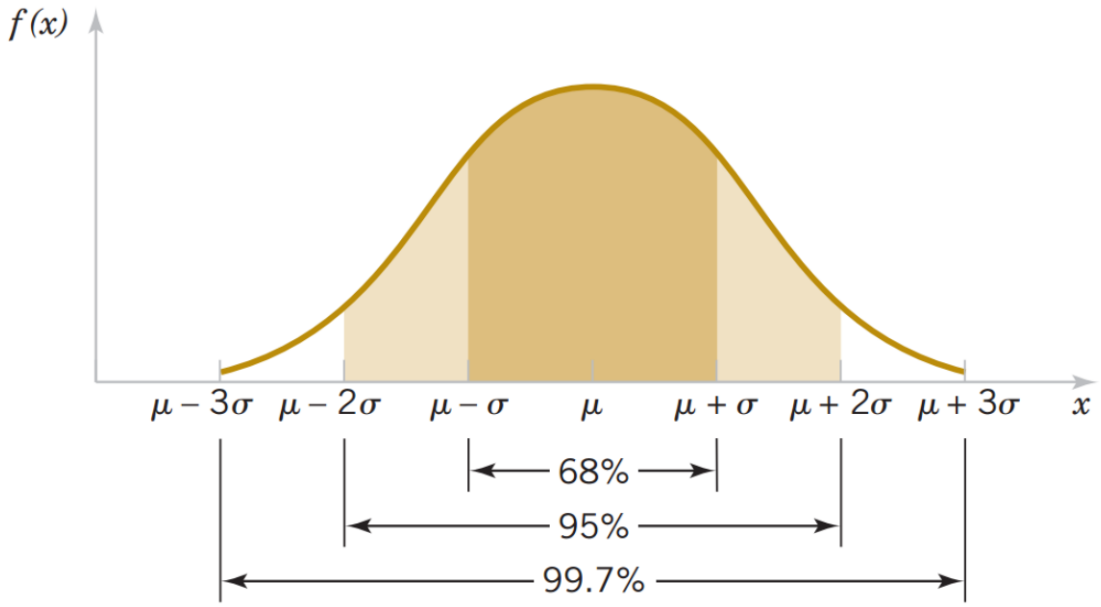

Figura 6: Distribuição gaussiana

Dessa forma, as linhas de código foram concebidas da seguinte maneira:

```
# análise de variação de tensão de alimentação, item 2.3
media_AveUrms1 = dados_audit['AveUrms1'].mean()
media_AveUrms2 = dados_audit['AveUrms2'].mean()
media_AveUrms3 = dados_audit['AveUrms2'].mean()
desvio_AveUrms1 = st.stdev(dados_audit['AveUrms1'])
desvio_AveUrms2 = st.stdev(dados_audit['AveUrms2'])
desvio_AveUrms3 = st.stdev(dados_audit['AveUrms3'])
limite_superior_AveUrms1 = media_AveUrms1 + 2*desvio_AveUrms1
limite_superior_AveUrms2 = media_AveUrms2 + 2*desvio_AveUrms2
limite_superior_AveUrms3 = media_AveUrms2 + 2*desvio_AveUrms3
limite_inferior_AveUrms1 = media_AveUrms1 - 2*desvio_AveUrms1
limite_inferior_AveUrms2 = media_AveUrms2 - 2*desvio_AveUrms2
limite_inferior_AveUrms3 = media_AveUrms3 - 2*desvio_AveUrms3

#item 2.3 sobretensão e subtensão
if (limite_superior_AveUrms1 > (230*1.1)):
  print('Valor eficaz médio na fase 1 acima do limite permitido de 10% em 95% dos dados')
if (limite_superior_AveUrms2 > (230*1.1)):
  print('Valor eficaz médio na fase 2 acima do limite permitido de 10% em 95% dos dados')
if (limite_superior_AveUrms3 > (230*1.1)):
  print('Valor eficaz médio na fase 3 acima do limite permitido de 10% em 95% dos dados')
if (limite_inferior_AveUrms1 < (230*0.9)):
  print('Valor eficaz médio na fase 1 abaixo do limite permitido de 10% em 95% dos dados')
if (limite_inferior_AveUrms2 < (230*0.9)):
  print('Valor eficaz médio na fase 2 abaixo do limite permitido de 10% em 95% dos dados')
if (limite_inferior_AveUrms3 < (230*0.9)):
  print('Valor eficaz médio na fase 3 abaixo do limite permitido de 10% em 95% dos dados')
if any ( dados_audit['AveUrms1'] < (230*0.85)) or any (dados_audit['AveUrms1'] > (230*1.1)):
  print('Valor eficaz médio na fase 1 fora do limite de +10%/-15% em algum dos dados')
if any ((230*0.85) > dados_audit['AveUrms2']) or any (dados_audit['AveUrms2']  > (230*1.1)):
  print('Valor eficaz médio na fase 2 fora do limite de +10%/-15% em algum dos dados')
if any ((230*0.85) > dados_audit['AveUrms3']) or any (dados_audit['AveUrms3'] > (230*1.1)):
  print('Valor eficaz médio na fase 3 fora do limite de +10%/-15% em algum dos dados')
else:
  print('Valor eficaz médio das 3 fases não ultrapassam os limites permitidos de +- 10% em
nenhum momento')
```

Como resultado, o programa emitiu seguinte mensagem, não detectando nenhuma inconformidade: *"Valor eficaz médio das 3 fases não ultrapassam os limites permitidos de +- 10% em nenhum momento".*

2.2- Análise de frequência

Relativamente à frequência nominal, o item 2.1 determina que "*a frequência nominal da tensão de alimentação deve ser igual a 50 Hz" e "no caso de redes com ligação síncrona a redes interligadas, entre 47 Hz e 52 Hz durante 100% do tempo".* Logo, foram concebidas as seguintes linhas de códigos que avaliam essas condições:

```
if any (dados_audit['MaxFreq'] > 52):
  print ('Frequência máxima extrapola o limite de 52 Hz')
  print ('A maior frequência é', dados_audit['MaxFreq'].max())
if any (dados_audit['MinFreq'] < 47) :
  print ('Frequência miníma extrapola o limite de 47 Hz')
  print ('A menor frequência é', dados_audit['MinFreq'].min())
else:
  print ('Frequência dentro do limite de 47 Hz a 52 Hz')
```

Ao correr o código, o programa expediu a seguinte mensagem: "*Frequência dentro do limite de 47 Hz a 52 Hz".*

2.3- Análise da taxa de distorção harmónica

Quanto à distorção harmónica total (THD), o item 2.11 preconiza que "*a distorção harmónica total (THD) da tensão de alimentação (incluindo as harmónicas até à ordem 40) não deve ultrapassar 8%".* Dessa forma, foram concebidas as seguintes linhas de código que realiza os testes condicionais:

```
if any (dados_audit['MaxUthd1'] > 8) :
  print ('Em algum momento a THD na fase 1 ultrapassa 8%')
if any (dados_audit['MaxUthd2'] > 8) :
  print ('Em algum momento a THD na fase 2 ultrapassa 8%')
if any (dados_audit['MaxUthd3'] > 8) :
  print ('Em algum momento a THD na fase 1 ultrapassa 8%')
else:
  print ('THD não ultrapassa 8% em nenhuma das fases')
```

Ao correr o código, o programa expediu a seguinte mensagem: "*THD não ultrapassa 8% em nenhuma das fases".*

2.4- Análise do peso relativo dos harmónicos individuais

Já em relação ao peso relativo dos harmónicos individuais, cada um dos 25 primeiros harmónicos possui um valor limite determinado pelo 2.11. Segundo a norma, "*Em condições normais de exploração, para cada período de uma semana, 95% dos valores eficazes médios de 10 min de cada tensão harmónica não devem exceder os valores indicados no quadro 1."*

Harmónicas ímpares				Harmónicas pares	
Não múltiplas de 3		Múltiplas de 3			
Ordem h	Tensão relativa (%)	Ordem h	Tensão relativa (%)	oOdem h	Tensão relativa (%)
5	6,0	3	5,0	2	2,0
7	5,0	9	1,5	4	1,0
11	3,5	15	0,5	6 ... 24	0,5
13	3,0	21	0,5		
17	2,0				
19	1,5				
23	1,5				
25					
NOTA: *Não são indicados valores para harmónicas de ordem superior a 25, por serem em geral de pequena amplitude, mas muito imprevisíveis devido a efeitos de ressonância.*					

Figura 7: Valores de tensões harmónicas até a ordem 25

Realizar essa tarefa de análise de cada peso relativo das harmónicas foi complexa, visto que devem ser levados em consideração 95% dos dados de cada harmonico de cada fase e cada um possui uma condição limite de funcionamento diferente. Para tal, utilizando-se da estratégia de distribuição gaussiana, foi criado uma matriz 3 X 24 chamada z2score_harmonicos que armazena os valores de cada limiar de tensão de harmónica no limiar de 95%. As linhas dessa matriz refere-se às 3 fases, enquanto as colunas referem-se às ordens de harmónicas de 2 à 25 citadas pela norma. Como os valores de tensão de cada harmónica já está em termos percentuais (tensão rms de ordem 1 = 100), bastou realizar o desvio padrão e média do grupo amostral para obtenção dos valores. Outro fator importante na realização das linhas de código está na coluna em que se inicia cada um dos valores de tensão rms dos harmonicos de cada fase sob análise. Assim, o código foi concebido da seguinte maneira:

```
zscore2_harmonicos = np.zeros((3,24))
for i, j in enumerate(range(90, 162, 3)):  #insercao da media de harmonicos da fase1
   zscore2_harmonicos[0][i] = dados_audit.iloc[0:,j].mean() + 2 * st.stdev(dados_audit.iloc[0:,j])
for i, j in enumerate(range(240, 312, 3)): #insercao da media de harmonicos da fase2
   zscore2_harmonicos[1][i] = dados_audit.iloc[0:,j].mean() + 2 * st.stdev(dados_audit.iloc[0:,j])
for i, j in enumerate(range(390, 462, 3)): #insercao da media de harmonicos da fase3
   zscore2_harmonicos[2][i] = dados_audit.iloc[0:,j].mean() + 2 * st.stdev(dados_audit.iloc[0:,j])

#criacao de tabela excel zscore2_harmonicos
linha = 'Fase1 Fase2 Fase3'.split()
coluna ='2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25'.split()
tabela = pd.DataFrame(data = zscore2_harmonicos, index = linha, columns = coluna)
tabela.to_excel('Peso realativo de harmonicos de tensao.xls')
```

Ordem	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Fase1	0,04	0,30	0,03	0,67	0,05	1,48	0,04	0,50	0,03	0,66	0,02	0,23	0,01	0,14	0,01	0,27	0,01	0,11	0,01	0,09	0,01	0,15	0,01	0,09
Fase2	0,04	0,37	0,04	0,64	0,04	1,59	0,04	0,48	0,03	0,76	0,01	0,32	0,01	0,17	0,01	0,22	0,02	0,18	0,01	0,14	0,01	0,08	0,01	0,11
Fase3	0,03	0,39	0,03	0,72	0,04	1,46	0,04	0,50	0,03	0,75	0,01	0,35	0,01	0,14	0,02	0,27	0,01	0,19	0,02	0,16	0,02	0,14	0,00	0,07

Figura 8: Limiar de tensão média eficaz (%) de 95% dos dados amostrais das harmónicas de 2 a 25 nas 3 fases

As linhas de código para avaliação das condições citadas na tabela do item 2.11 seguem abaixo:

```
for i in range (3):
  if (zscore2_harmonicos[i][0] > 2):
     print('Tensão relativa no harmonico 2 da fase ',[i],'maior que 2%'  )
for i in range (3):
  if (zscore2_harmonicos[i][1] > 5):
     print('Tensão relativa no harmonico 3 da fase ',[i],'maior que 5%'  )
for i in range (3):
  if (zscore2_harmonicos[i][2] > 1):
     print('Tensão relativa no harmonico 4 da fase ',[i],'maior que 1%'  )
for i in range (3):
  if (zscore2_harmonicos[i][3] > 6):
     print('Tensão relativa no harmonico 5 da fase ',[i],'maior que 6%'  )
for i in range (3):
  for j in range (4,22,2):
     if (zscore2_harmonicos[i][j] > 0.5):
```

```
        print('Tensão relativa no harmonico par de 6...24 da fase ',[i],'maior que 0,5%'  )
for i in range (3):
  if (zscore2_harmonicos[i][5] > 5):
     print('Tensão relativa no harmonico 7 da fase ',[i],'maior que 5%')
for i in range (3):
  if (zscore2_harmonicos[i][7] > 1.5):
     print('Tensão relativa no harmonico 9 da fase ',[i],'maior que 1.5%')
for i in range (3):
  if (zscore2_harmonicos[i][9] > 3.5):
     print('Tensão relativa no harmonico 11 da fase ',[i],'maior que 3.5%')
for i in range (3):
  if (zscore2_harmonicos[i][11] > 3):
     print('Tensão relativa no harmonico 13 da fase ',[i],'maior que 3%')
for i in range (3):
  if (zscore2_harmonicos[i][13] > 0.5):
     print('Tensão relativa no harmonico 15 da fase ',[i],'maior que 0.5%')
for i in range (3):
  if (zscore2_harmonicos[i][15] > 2):
     print('Tensão relativa no harmonico 17 da fase ',[i],'maior que 2%')
for i in range (3):
  if (zscore2_harmonicos[i][17] > 1.5):
     print('Tensão relativa no harmonico 19 da fase ',[i],'maior que 1.5%')
for i in range (3):
  if (zscore2_harmonicos[i][19] > 0.5):
     print('Tensão relativa no harmonico 21 da fase ',[i],'maior que 0.5%')
for i in range (3):
  if (zscore2_harmonicos[i][21] > 1.5):
     print('Tensão relativa no harmonico 23 da fase ',[i],'maior que 1.5%')
for i in range (3):
  if (zscore2_harmonicos[i][23] > 1.5):
     print('Tensão relativa no harmonico 25 da fase ',[i],'maior que 1.5%')
else:
  print('As tensões em cada harmónica de ordem 2 a 25 nas 3 fases não excede o limite
estabelecido')
```

Como resultado, ao correr as linhas de código, o programa emite a mensagem: “As tensões em cada harmónica de ordem 2 a 25 nas 3 fases não excede o limite estabelecido”

2.5- Análise de desequilíbrio de tensões

Quanto ao desequilíbrio das tensões de alimentação, o item 2.10 cita que “*Em condições normais de exploração, para cada período de uma semana, 95% dos valores eficazes médios de 10 min da componente inversa das tensões não devem ultrapassar 2% da correspondente componente directa”.* Dessa forma, utilizando-se do mesmo recurso de distribuição gaussiana, forma concebidos os seguintes códigos:

```
#item 2.10 desequilíbrio
media_MaxUunb = dados_audit['MaxUunb'].mean()
```

```
desvio_MaxUunb = st.stdev(dados_audit['MaxUunb'])
limite_MaxUunb  = media_MaxUunb  + 2*desvio_MaxUunb
if (limite_MaxUunb  > 1.02 ):
  print('Desequilíbrio das tensões acima do limite tolerável' )
else:
  print('Desequilíbrio das tensões dentro do limite tolerável')
```

Ao correr o código, o programa expediu a seguinte mensagem: “*Desequilíbrio das tensões dentro do limite tolerável”.*

3- Análise da THD das correntes e do desequilíbrio entre fases

Conforme dito anteriormente, os valores limites de THD de correntes e seus desequilíbrios não são abrangidos pela norma EN50160 e sim pela norma IEC 61000-3-12. Embora essa análise não seja avaliada neste trabalho, é possível realizar considerações acerca desses parâmetros. Para isso, foi plotado a máxima THD das correntes das 3 fases durante o período de uma semana por meio das seguintes linhas de código:

```
x = list(range(1008))

fig1, ax = plt.subplots()
ax.plot(x, dados_audit['MaxIthd1'])
ax.set_ylabel('THD máxima de corrente [%]')
fig1.savefig('THD maxima de corrente fase1.png')

fig2, bx = plt.subplots()
bx.plot(x, dados_audit['MaxIthd2'])
bx.set_ylabel('THD máxima de corrente [%]')
fig2.savefig('THD maxima de corrente fase2.png')

fig3, cx = plt.subplots()
cx.plot(x, dados_audit['MaxIthd3'])
cx.set_ylabel('THD máxima de corrente [%]')
fig3.savefig('THD maxima de corrente fase3.png')
```

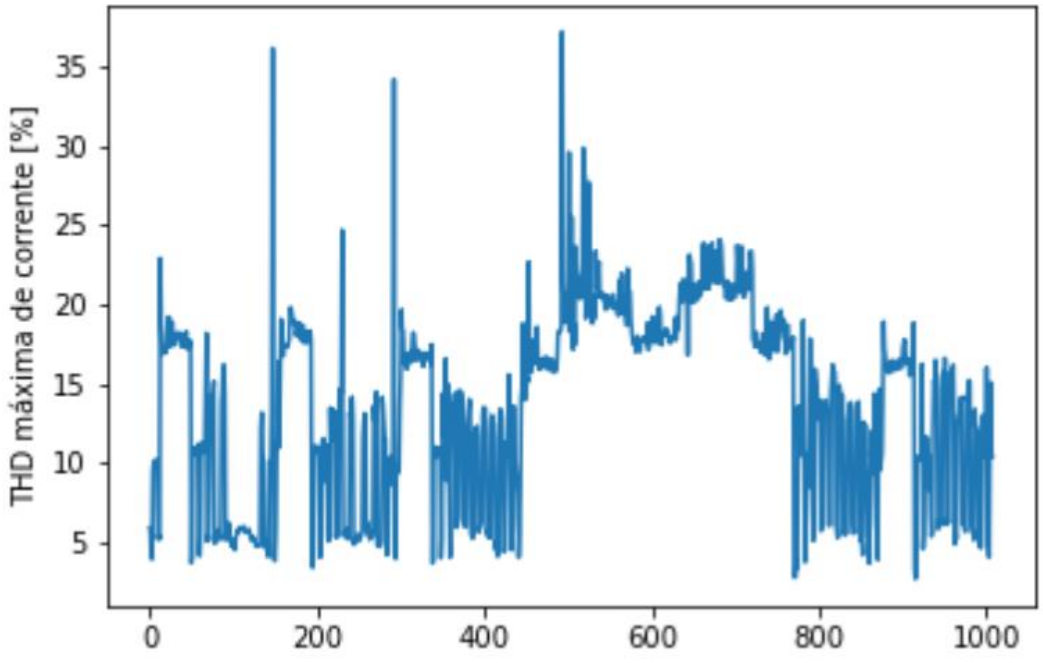

Figura 9: THD máxima da corrente da fase 1 em uma semana

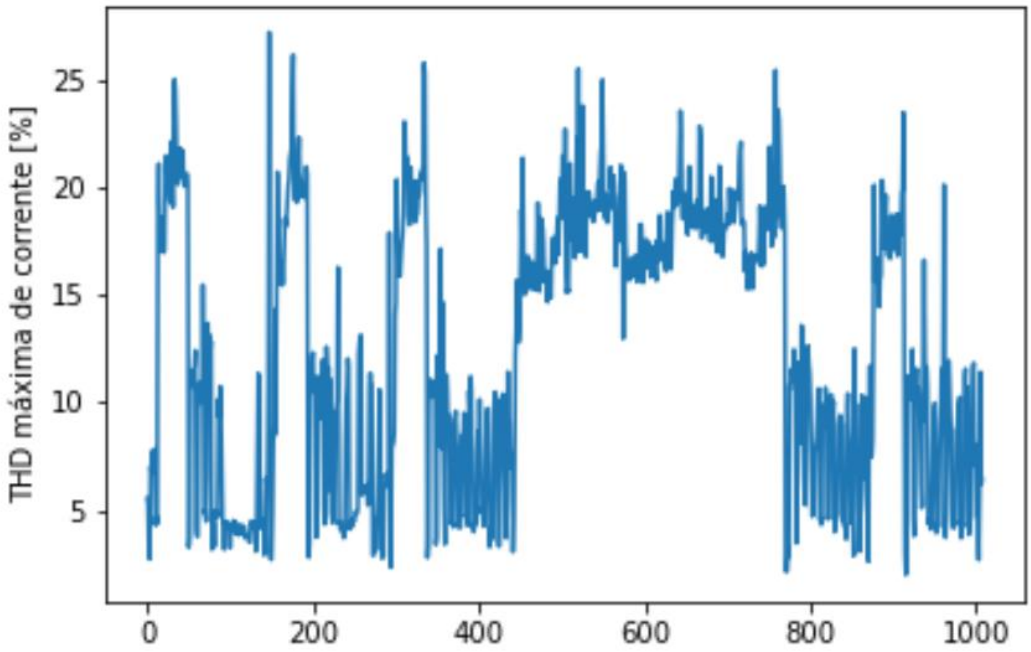

Figura 10: THD máxima da corrente da fase 2 em uma semana

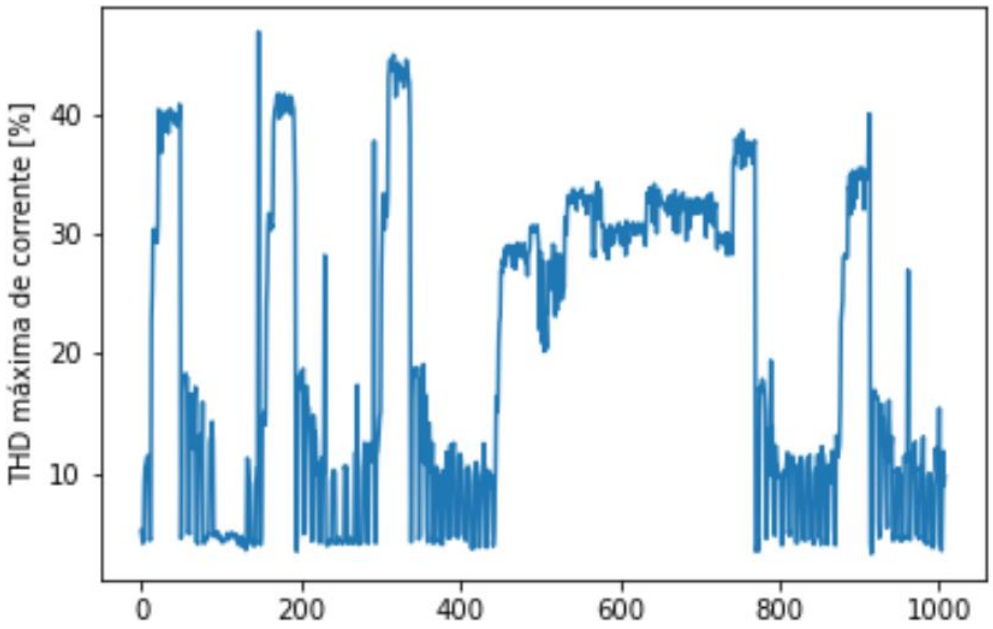

Figura 11: THD máxima da corrente da fase 3 em uma semana

Tal como os seus desequilíbrios:

```
fig4, dx = plt.subplots()
dx.plot(x, dados_audit['MaxIunb'])
dx.set_ylabel('Taxa de desequilíbrio das correntes')
fig4.savefig('Taxa de desequilíbrio das correntes.png')
```

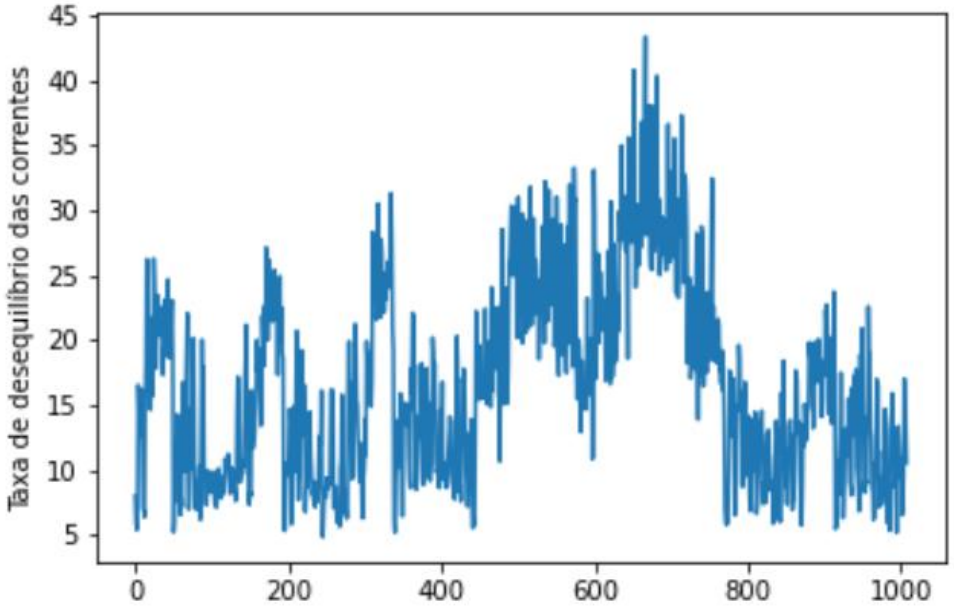

Figura 12: Desequilíbrio das correntes em uma semana

Realizando o teste de correlação de Pearson é possível ver uma forte correlação entre THD e o desequilíbrio entre as fases, pois os dois fenómenos estão ligados, sendo um consequência do outro.

```
from scipy import stats

print(stats.pearsonr(dados_audit['MaxIthd1'], dados_audit['MaxIunb'])) #coefieciente de pearson para fase 1

print(stats.pearsonr(dados_audit['MaxIthd2'], dados_audit['MaxIunb'])) #coefieciente de pearson para fase 2

print(stats.pearsonr(dados_audit['MaxIthd3'], dados_audit['MaxIunb'])) #coefieciente de pearson para fase 3
```

Coeficientes de Pearson:

(0.8312292099317101, 9.34835342700227e-259) para a fase 1

(0.8039125912944352, 2.9763435751185735e-229) para a fase 2

(0.7867811906367903, 4.9607577958763836e-213) para a fase 3