

Prática 02

Filipe Augusto Parreira Almeida, RA: 2320622

3 de Setembro - 2023

Questão – 1

- 1) Crie uma série **s1** composto por um cosseno de amplitude 1 com 200 pontos e 4 ciclos.
- 2) Crie um sinal **s2** composto por um cosseno de amplitude 1 com 200 pontos 20 ciclos.
- 3) Crie um sinal **s3** composto pela soma de **s1** e **s2**.
- 4) Crie um sinal **h** com 200 amostras onde os primeiros 9 valores são iguais 1/9 e os demais iguais a zero
- 5) Crie o sinal **y** composto pela convolução de s3 e h no modo "same" (também com 200 pontos).

- Código:

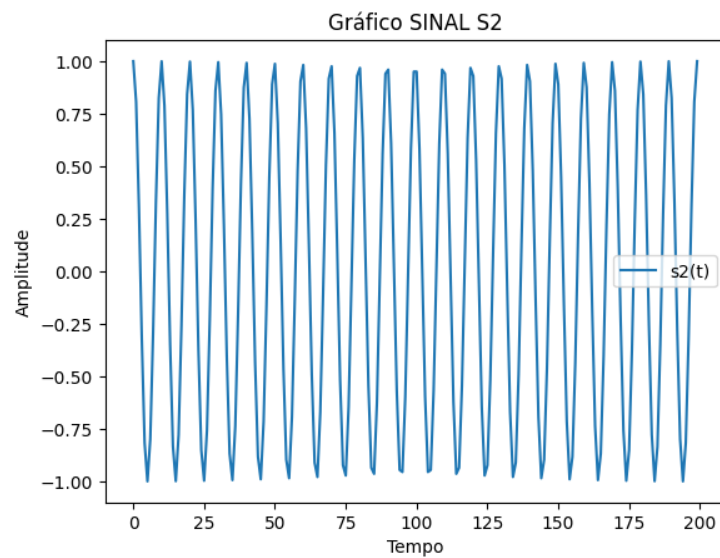
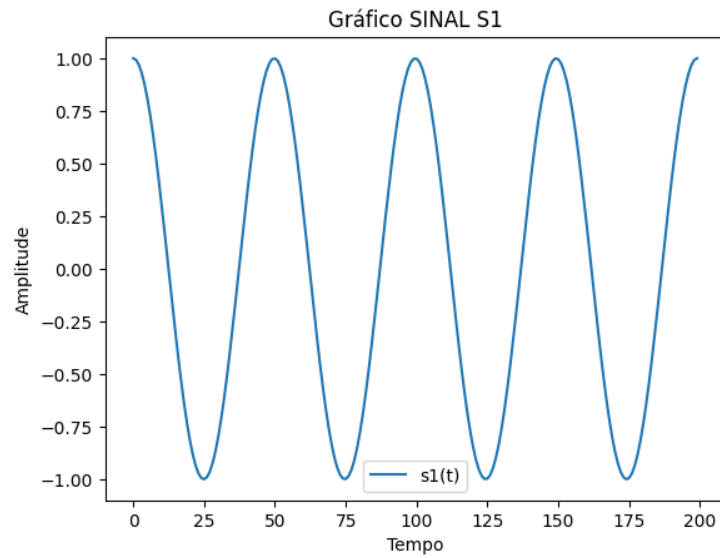
```
1 t1 = np.linspace(0, 4*2*np.pi, 200);
2 s1 = np.cos(t1);
3
4 t2 = np.linspace(0, 20*2*np.pi, 200);
5 s2 = np.cos(t2);
6
7 s3 = s1 + s2
8
9 h1 = np.array([1/9] * 9)
10 h0 = np.array([0] * (200-9))
11 h = np.concatenate((h1, h0), axis = None)
12
13 y = np.convolve(h, s3, 'full')[:200]
14
15 plot(s1, 's1(t)', 'Tempo', 'Amplitude', 'Gráfico SINAL S1')
16 plot(s2, 's2(t)', 'Tempo', 'Amplitude', 'Gráfico SINAL S2')
```

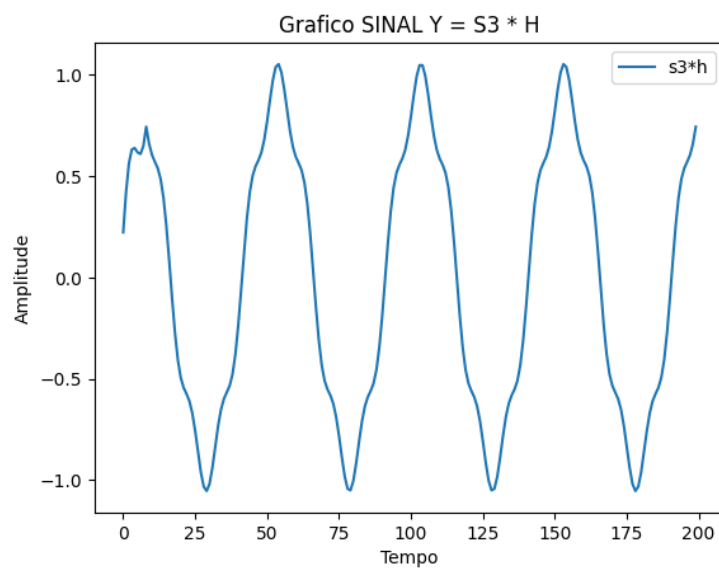
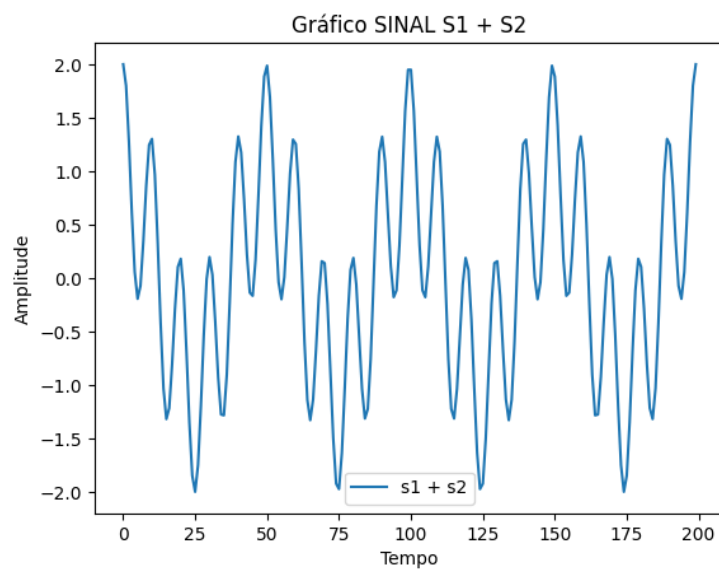
```

17 plot(s3, 's1 + s2', 'Tempo', 'Amplitude', 'Gráfico SINAL S1 + S2')
18 plot(y, 's3*h', 'Tempo', 'Amplitude', 'Gráfico SINAL Y = S3 * H')

```

- Plots:





Questão – 2

Efeitos em frequência de cada sinal

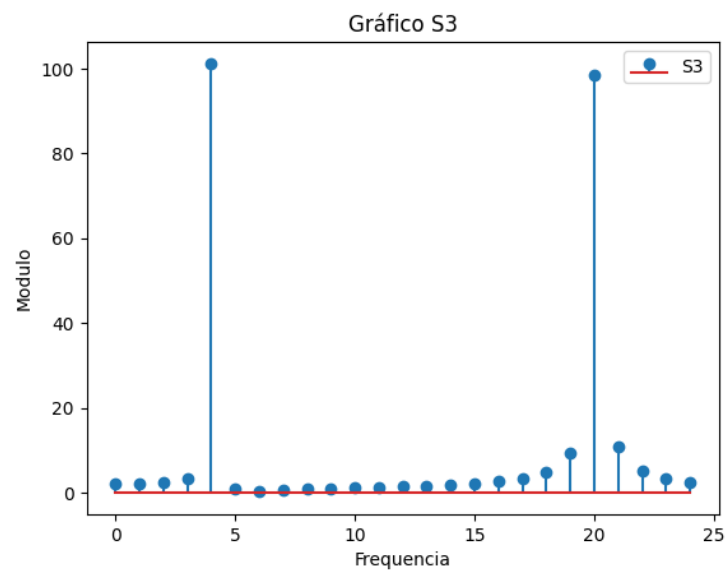
- Código:

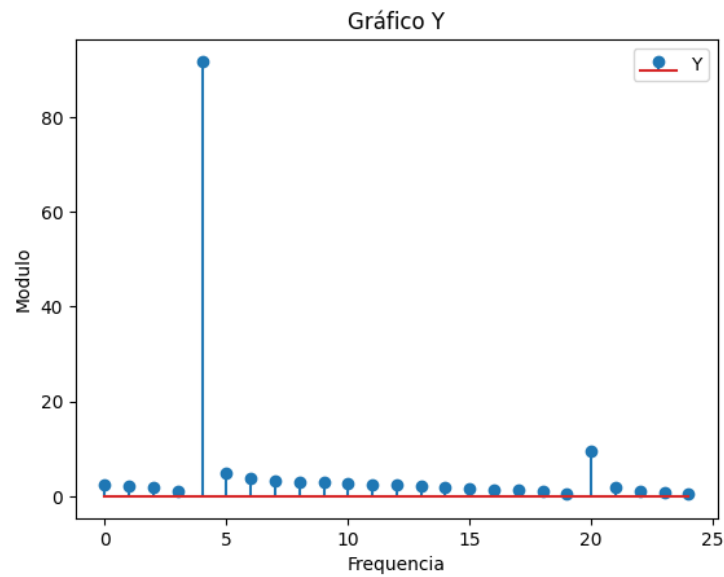
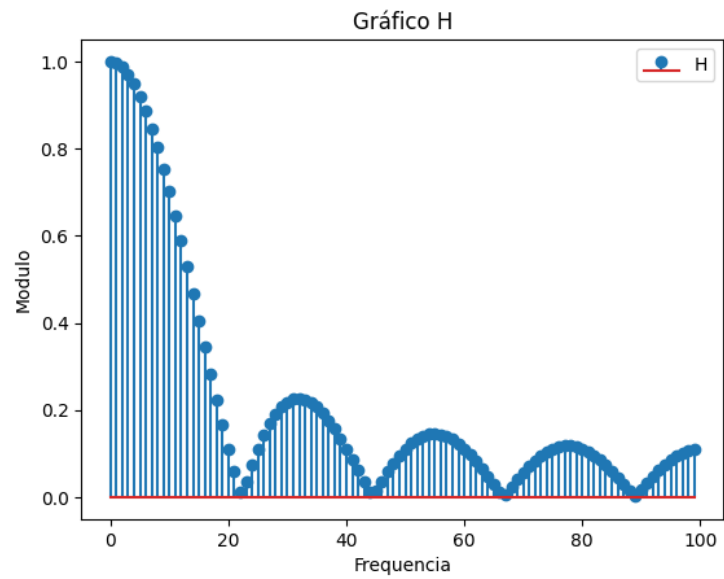
```

1 S3 = np.abs(np.fft.fft(s3))
2 H = np.abs(np.fft.fft(h))
3 Y = np.abs(np.fft.fft(y))
4
5 plotStem(S3[0:25], 'S3', 'Frequencia', 'Modulo', 'Gráfico S3')
6 plotStem(H[0:100], 'H', 'Frequencia', 'Modulo', 'Gráfico H')
7 plotStem(Y[0:25], 'Y', 'Frequencia', 'Modulo', 'Gráfico Y')

```

- Plots:





- Análise:

Analisando os gráficos em frequência das transformadas de Fourier de cada sinal, podemos ver as componentes que compõem cada sinal, onde cada componente representa uma senoide de diferentes frequências, logo, a soma de todas as senoides resulta no sinal transformado.

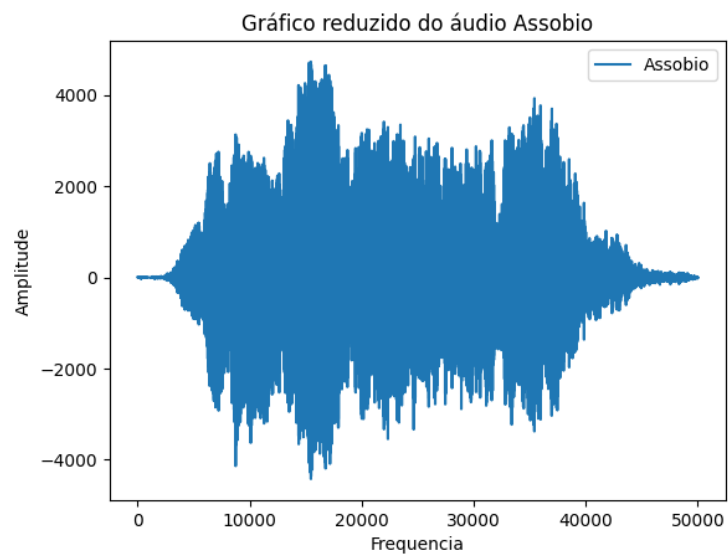
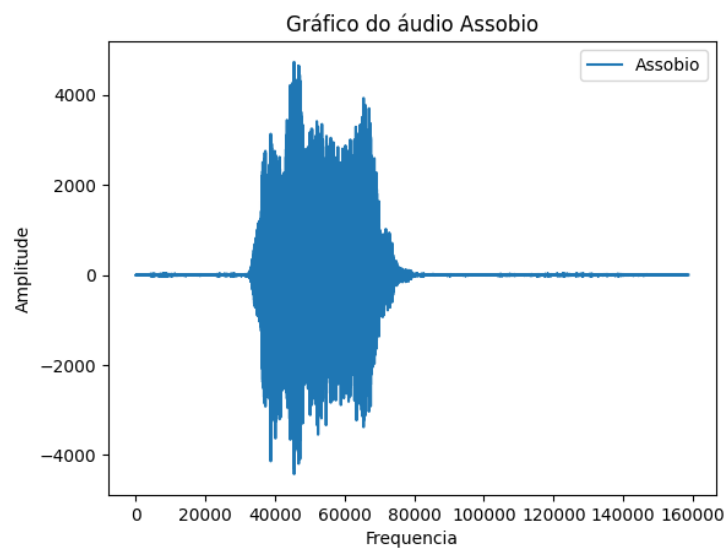
Questão – 3

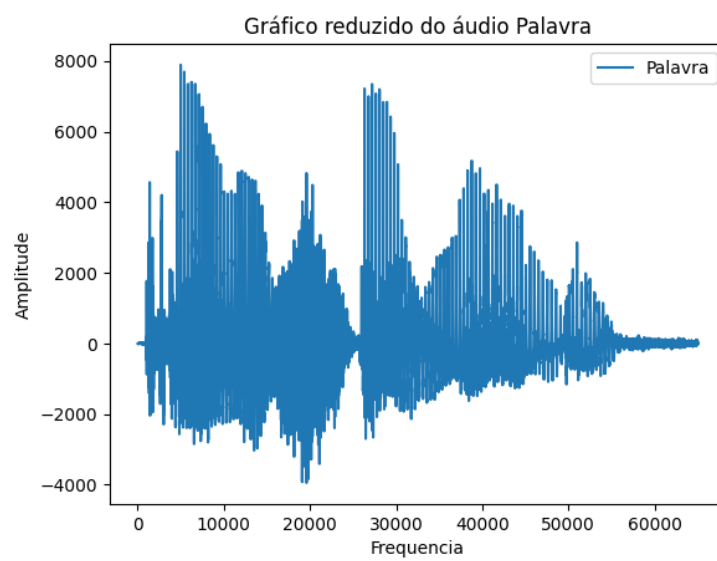
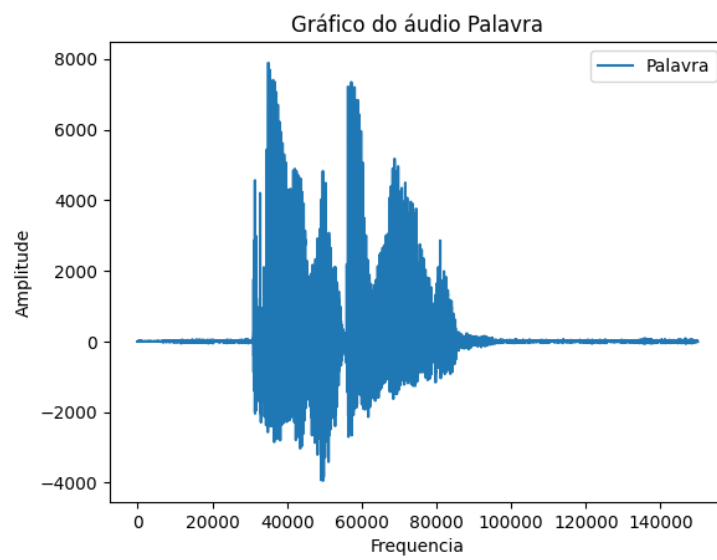
Análise sonora de dois áudios (um de assobio e outro falando a palavra “transformada”), plot de 200 amostras de cada áudio e de seus espectros em frequência, respectivamente.

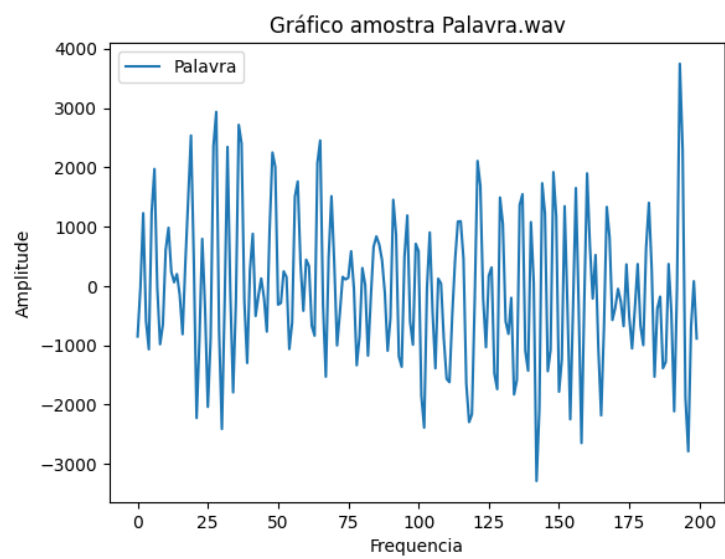
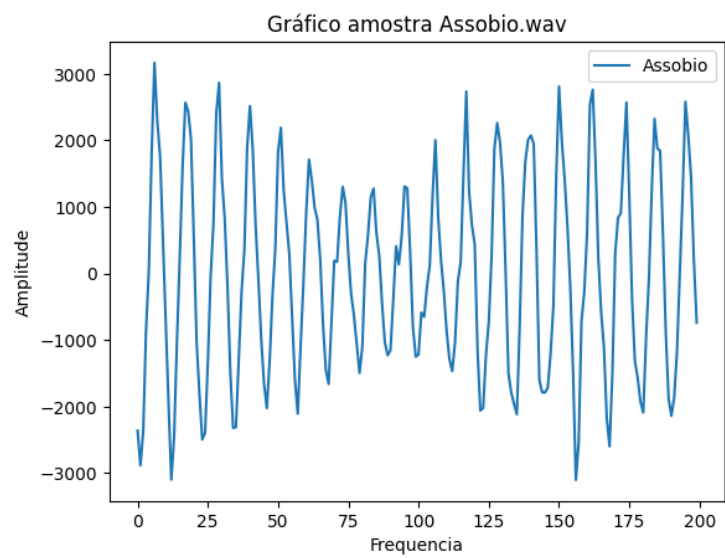
- Código:

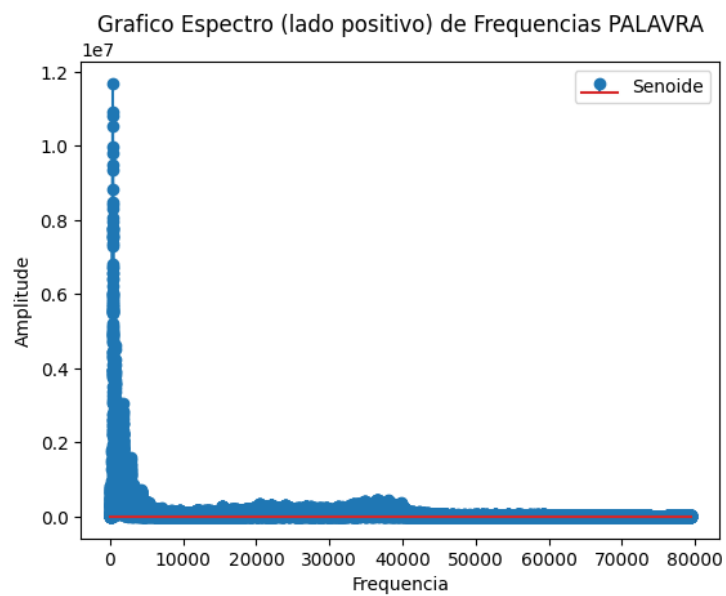
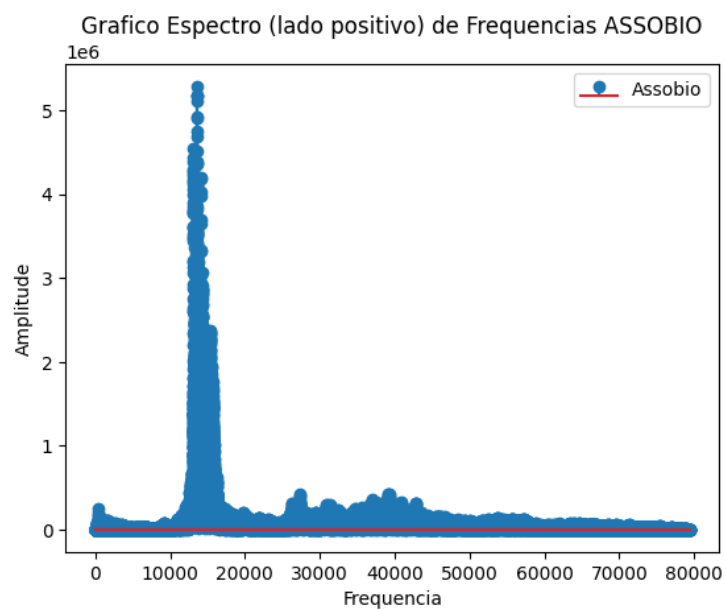
```
1 rateA, dataA = wavfile.read('Assobio.wav')
2 plot(dataA, 'Assobio', 'Frequencia', 'Amplitude', 'Gráfico do ...
   udio Assobio')
3
4 plot(dataA[30000:80000], 'Assobio', 'Frequencia', 'Amplitude', ...
   'Gráfico reduzido do udio Assobio')
5
6 rateP, dataP = wavfile.read('Palavra.wav')
7
8 plot(dataP, 'Palavra', 'Frequencia', 'Amplitude', 'Gráfico do ...
   udio Palavra')
9
10 plot(dataP[30000:95000], 'Palavra', 'Frequencia', 'Amplitude', ...
   'Gráfico reduzido do udio Palavra')
11
12 plot(dataA[50000:50200], 'Assobio', 'Frequencia', 'Amplitude', ...
   'Gráfico amostra Assobio.wav')
13
14 plot(dataP[50000:50200], 'Palavra', 'Frequencia', 'Amplitude', ...
   'Gráfico amostra Palavra.wav')
15
16 ASSOPIO = np.abs(np.fft.fft(dataA))
17 PALAVRA = np.abs(np.fft.fft(dataP))
18 plotStem(ASSOPIO[0:int(dataA.size/2)], 'Senoide', 'Frequencia', ...
   'Amplitude', 'Gráfico Espectro (lado positivo) de ...
   Frequências ASSOPIO')
19
20 plotStem(PALAVRA[0:int(dataA.size/2)], 'Senoide', 'Frequencia', ...
   'Amplitude', 'Gráfico Espectro (lado positivo) de ...
   Frequências PALAVRA')
```

- Plots:









- Análise:

Analisando os gráficos e tendo em mente que os gráficos de frequência absoluta representam as frequências de cada senoide, conclui-se que em frequência em

Hz os dois sinais são bem **semelhantes** tendo uma concentração predominante de valores entre **30000 e 80000**, porém em **amplitude** o sinal da **Palavra** varia em um pouco mais de **4000** em alguns momentos. Já na **frequência absoluta** tempos que o sinal do **Assobio** é composto por senoides em alta frequência (em torno de 10000 a 20000), já o sinal da **Palavra** é composto por senoides em mais baixa frequência (0-7000).

Questão – 4

Criar um sinal **y2** sendo a soma dos dois áudios, calcular e plotar a **fft** dos sinal **y2**, identificar as frequências do assobio e gerar um vetor de 1's chamado **F**, onde terão 0's nas posições das frequências dos assobio, multiplicar o vetor **F** pelo sinal **Y2** gerando assim o sinal **Y3**, plotar o modulo de **Y3**, realizar a transformada inversa do sinal, escutar e analisar o resultado.

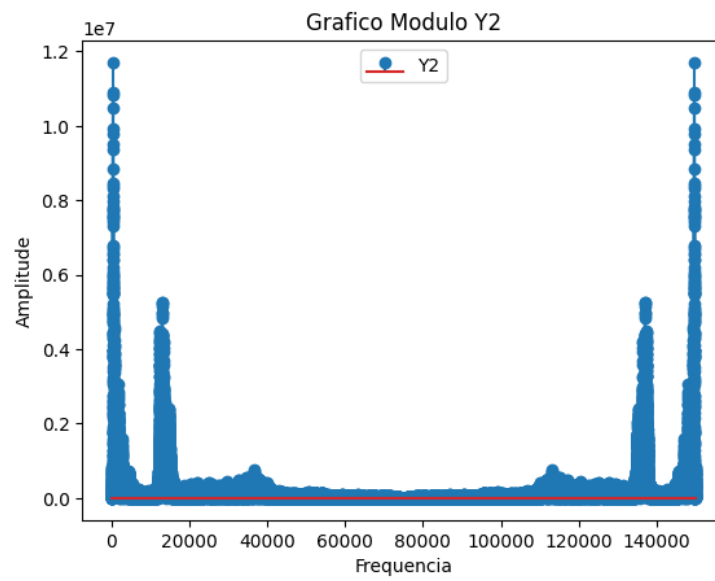
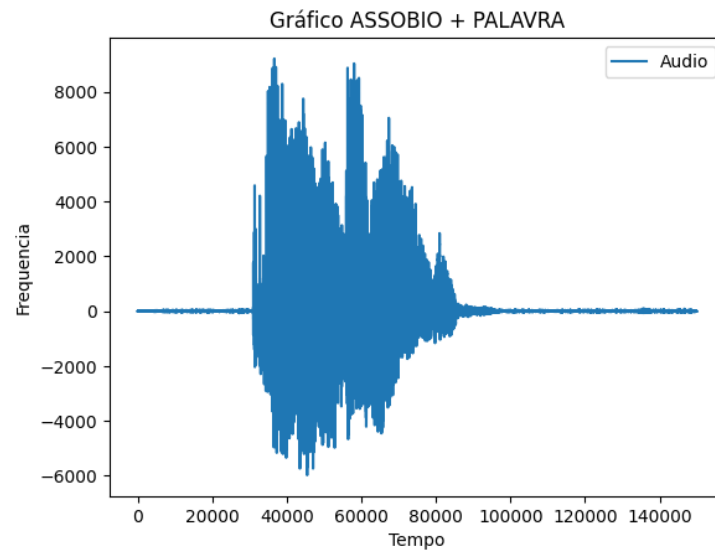
- Código:

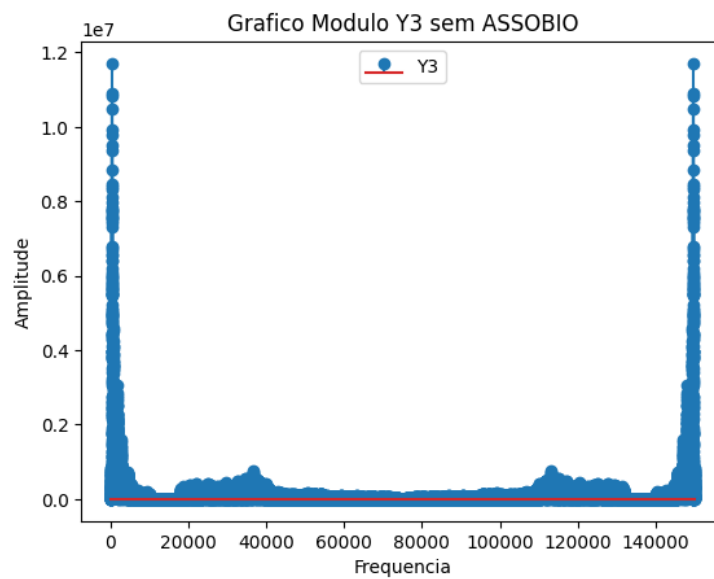
```

1  y2 = dataA[0:len(dataP)] + dataP
2  plot(y2, 'Audio', 'Tempo', 'Frequencia', 'Gráfico ASSOBIO + ...
    PALAVRA')
3  wavfile.write(filename="y2.wav", rate=rateA, data=y2)
4
5  Y2 = np.fft.fft(y2)
6  Y2a = np.abs(Y2)
7  plotStem(Y2a, 'Y2', 'Frequencia', 'Amplitude', 'Gráfico Modulo ...
    Y2 com ASSOBIO')
8
9  F = np.ones(len(Y2))
10 F[1600:1700] = 0
11 F[(len(Y2)-1700):((len(Y2)-1700)+100)] = 0
12 Y3 = Y2*F
13
14 plotStem(np.abs(Y3), 'Y3', 'Frequencia', 'Amplitude', 'Gráfico ...
    Modulo Y3 sem ASSOBIO')
15
16 F = np.ones(len(Y2))
17 F[10000:18000] = 0
18 F[(len(Y2)-18000):((len(Y2)-18000)+8000)] = 0
19 Y3 = Y2*F
20
21 plotStem(np.abs(Y3), 'Y3', 'Frequencia', 'Amplitude', 'Gráfico ...
    Modulo Y3 sem ASSOBIO')
22
23 y3 = np.abs(np.fft.ifft(Y3))
24 x_normalized = np.clip(y3, -1, 1)
25 x_int16 = (x_normalized * 32767).astype(np.int16)
26 print(y3.dtype)
27 print(y2.dtype)
28 wavfile.write(filename="y3.wav", rate=rateA, data=y3)

```

- Plot:





- Análise:

Realizando a transformada inversa de Fourier no sinal **Y3** é perceptível que **não há mais o assobio**, porém percebe-se também que o áudio da palavra ficou com um pouco de **ruido**, danificando um pouco a qualidade do mesmo, resultando em uma **discrepância** perceptível entre o **sinal transformado** e o **sinal original**, pesquisando, conclui que isso pode ser por conta do método utilizado para realizar a transformada e a transformada inversa, resultando em possíveis artefatos numéricos ou truncamentos e aproximações.