

Lab 1 - TELC11A

Experimento 1

Código:

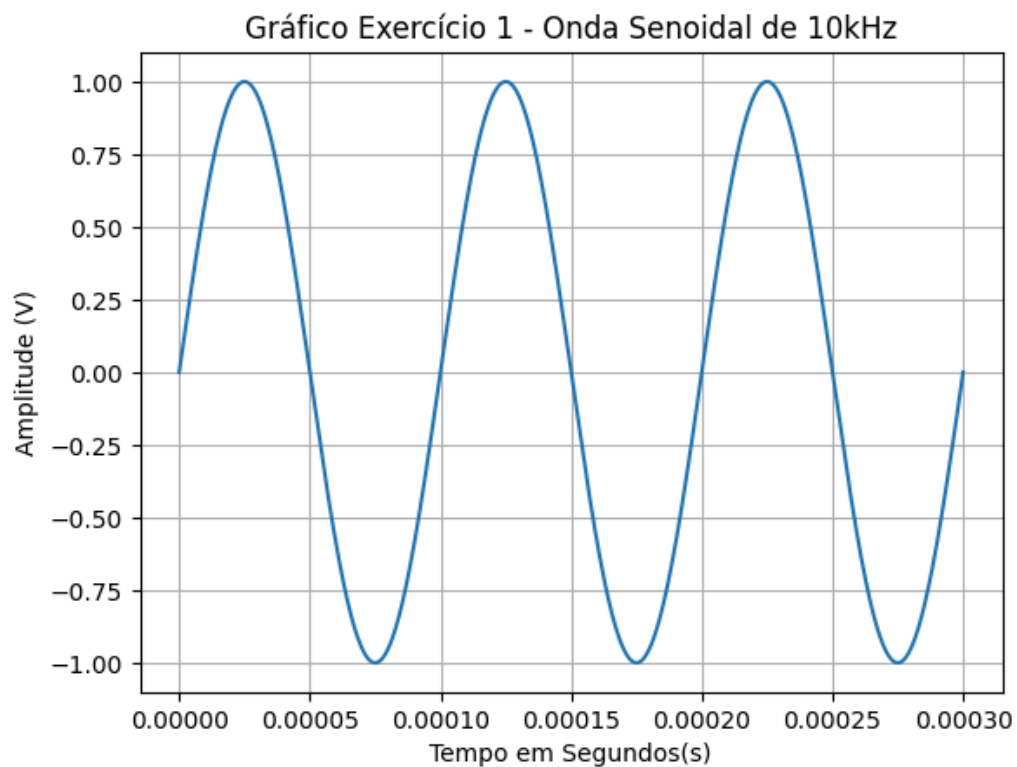
```
f = 10000
A = 1
T = 1/f

tempo = np.linspace(0, 3*T, 1000)
y = A*np.sin(2*np.pi*f*tempo)

plt.plot(tempo, y)

plt.xlabel("Tempo em Segundos(s)")
plt.ylabel("Amplitude (V)")
plt.title("Gráfico Exercício 1 - Onda Senoidal de 10kHz")
plt.grid(True)
plt.figure(1)
plt.show()
```

Gráfico Gerado:



No gráfico acima vemos uma onda senoidal com 1V de amplitude e frequência de 10kHz, que pode ser visualizada no primeiro período da onda que vai até 0,00010. Assim temos $f = 1/t$ que seria $f = 1/0,00010 = 10.000 \text{ Hz}$.

Experimento 2

Código:

```
f = 10000
A = 1
T = 1/f

tempo = np.linspace(0,3*T,100)
y = A*np.sin(2*np.pi*f*tempo)

print("Tabela de Amostras ")
print("  N° | Tempo (s) | Amplitude (V)")
for i in range(len(tempo)):
    print(f" {i+1:3d} | {tempo[i]:.8f} | {y[i]:.8f}")

plt.plot(tempo, y)
plt.xlabel("Tempo em Segundos(s)")
plt.ylabel("Amplitude (V)")
plt.title("Gráfico Exercício 2 - Onda Senoidal de 10kHz")
plt.grid(True)
plt.figure(2)
plt.show()
```

Gráfico Gerado:

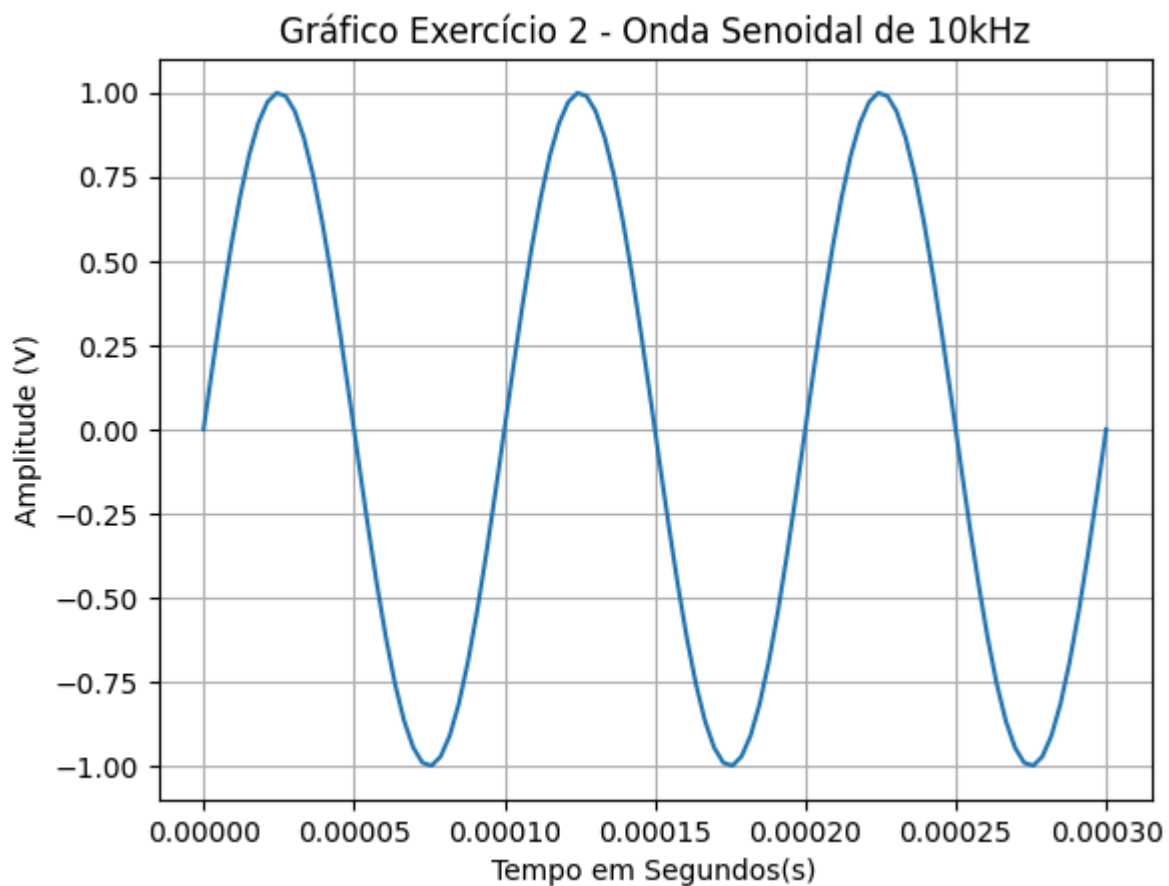


Tabela Gerada com 100 amostras:

| Nº | Tempo (s) | Amplitude (V) |
|----|------------|---------------|
| 1 | 0.00000000 | 0.00000000 |
| 2 | 0.00000303 | 0.18925124 |
| 3 | 0.00000606 | 0.37166246 |
| 4 | 0.00000909 | 0.54064082 |
| 5 | 0.00001212 | 0.69007901 |
| 6 | 0.00001515 | 0.81457595 |
| 7 | 0.00001818 | 0.90963200 |
| 8 | 0.00002121 | 0.97181157 |
| 9 | 0.00002424 | 0.99886734 |
| 10 | 0.00002727 | 0.98982144 |
| 11 | 0.00003030 | 0.94500082 |
| 12 | 0.00003333 | 0.86602540 |
| 13 | 0.00003636 | 0.75574957 |
| 14 | 0.00003939 | 0.61815899 |

| | | | | |
|----|--|------------|--|-------------|
| 15 | | 0.00004242 | | 0.45822652 |
| 16 | | 0.00004545 | | 0.28173256 |
| 17 | | 0.00004848 | | 0.09505604 |
| 18 | | 0.00005152 | | -0.09505604 |
| 19 | | 0.00005455 | | -0.28173256 |
| 20 | | 0.00005758 | | -0.45822652 |
| 21 | | 0.00006061 | | -0.61815899 |
| 22 | | 0.00006364 | | -0.75574957 |
| 23 | | 0.00006667 | | -0.86602540 |
| 24 | | 0.00006970 | | -0.94500082 |
| 25 | | 0.00007273 | | -0.98982144 |
| 26 | | 0.00007576 | | -0.99886734 |
| 27 | | 0.00007879 | | -0.97181157 |
| 28 | | 0.00008182 | | -0.90963200 |
| 29 | | 0.00008485 | | -0.81457595 |
| 30 | | 0.00008788 | | -0.69007901 |
| 31 | | 0.00009091 | | -0.54064082 |
| 32 | | 0.00009394 | | -0.37166246 |
| 33 | | 0.00009697 | | -0.18925124 |
| 34 | | 0.00010000 | | 0.00000000 |
| 35 | | 0.00010303 | | 0.18925124 |
| 36 | | 0.00010606 | | 0.37166246 |
| 37 | | 0.00010909 | | 0.54064082 |
| 38 | | 0.00011212 | | 0.69007901 |
| 39 | | 0.00011515 | | 0.81457595 |
| 40 | | 0.00011818 | | 0.90963200 |
| 41 | | 0.00012121 | | 0.97181157 |
| 42 | | 0.00012424 | | 0.99886734 |
| 43 | | 0.00012727 | | 0.98982144 |
| 44 | | 0.00013030 | | 0.94500082 |
| 45 | | 0.00013333 | | 0.86602540 |
| 46 | | 0.00013636 | | 0.75574957 |
| 47 | | 0.00013939 | | 0.61815899 |
| 48 | | 0.00014242 | | 0.45822652 |
| 49 | | 0.00014545 | | 0.28173256 |
| 50 | | 0.00014848 | | 0.09505604 |
| 51 | | 0.00015152 | | -0.09505604 |
| 52 | | 0.00015455 | | -0.28173256 |
| 53 | | 0.00015758 | | -0.45822652 |
| 54 | | 0.00016061 | | -0.61815899 |

55 | 0.00016364 | -0.75574957
56 | 0.00016667 | -0.86602540
57 | 0.00016970 | -0.94500082
58 | 0.00017273 | -0.98982144
59 | 0.00017576 | -0.99886734
60 | 0.00017879 | -0.97181157
61 | 0.00018182 | -0.90963200
62 | 0.00018485 | -0.81457595
63 | 0.00018788 | -0.69007901
64 | 0.00019091 | -0.54064082
65 | 0.00019394 | -0.37166246
66 | 0.00019697 | -0.18925124
67 | 0.00020000 | 0.00000000
68 | 0.00020303 | 0.18925124
69 | 0.00020606 | 0.37166246
70 | 0.00020909 | 0.54064082
71 | 0.00021212 | 0.69007901
72 | 0.00021515 | 0.81457595
73 | 0.00021818 | 0.90963200
74 | 0.00022121 | 0.97181157
75 | 0.00022424 | 0.99886734
76 | 0.00022727 | 0.98982144
77 | 0.00023030 | 0.94500082
78 | 0.00023333 | 0.86602540
79 | 0.00023636 | 0.75574957
80 | 0.00023939 | 0.61815899
81 | 0.00024242 | 0.45822652
82 | 0.00024545 | 0.28173256
83 | 0.00024848 | 0.09505604
84 | 0.00025152 | -0.09505604
85 | 0.00025455 | -0.28173256
86 | 0.00025758 | -0.45822652
87 | 0.00026061 | -0.61815899
88 | 0.00026364 | -0.75574957
89 | 0.00026667 | -0.86602540
90 | 0.00026970 | -0.94500082
91 | 0.00027273 | -0.98982144
92 | 0.00027576 | -0.99886734
93 | 0.00027879 | -0.97181157
94 | 0.00028182 | -0.90963200

| | | | | |
|-----|--|------------|--|-------------|
| 95 | | 0.00028485 | | -0.81457595 |
| 96 | | 0.00028788 | | -0.69007901 |
| 97 | | 0.00029091 | | -0.54064082 |
| 98 | | 0.00029394 | | -0.37166246 |
| 99 | | 0.00029697 | | -0.18925124 |
| 100 | | 0.00030000 | | 0.00000000 |

Experimento 3

Código:

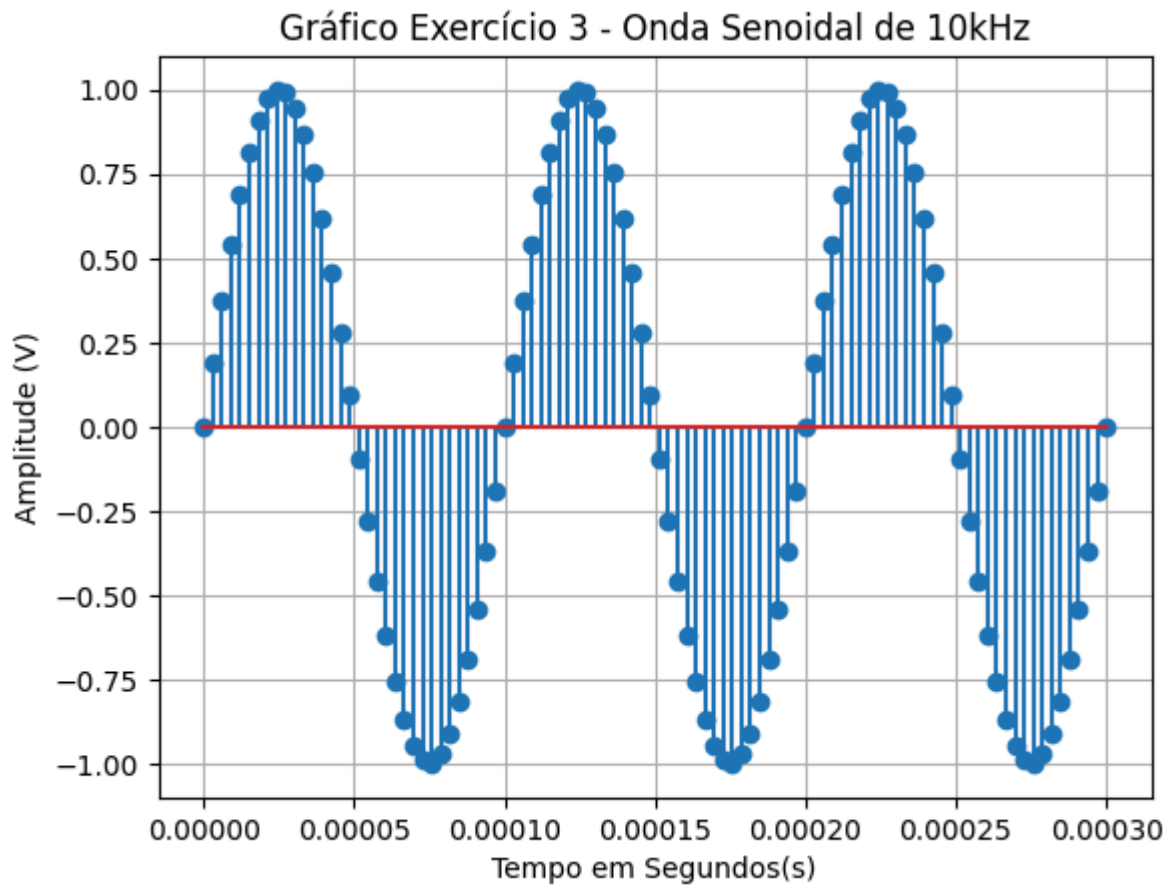
```
f = 10000
A = 1
T = 1/f

tempo = np.linspace(0, 3*T, 100)
y = A*np.sin(2*np.pi*f*tempo)

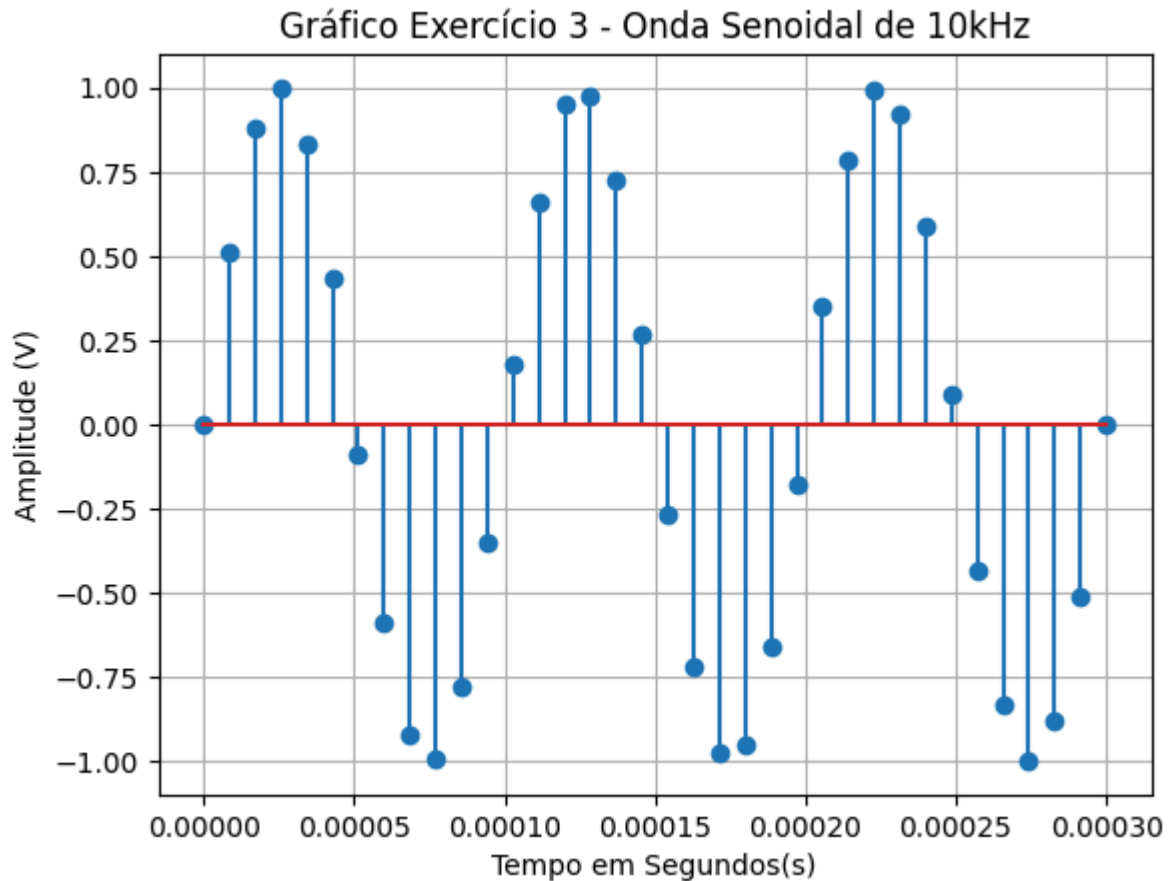
print("Tabela de Amostras ")
print("  N° | Tempo (s) | Amplitude (V)")
for i in range(len(tempo)):
    print(f" {i+1:3d} | {tempo[i]:.8f} | {y[i]:.8f}")

plt.stem(tempo, y)
plt.xlabel("Tempo em Segundos(s)")
plt.ylabel("Amplitude (V)")
plt.title("Gráfico Exercício 3 - Onda Senoidal de 10kHz")
plt.grid(True)
plt.figure(3)
plt.show()
```

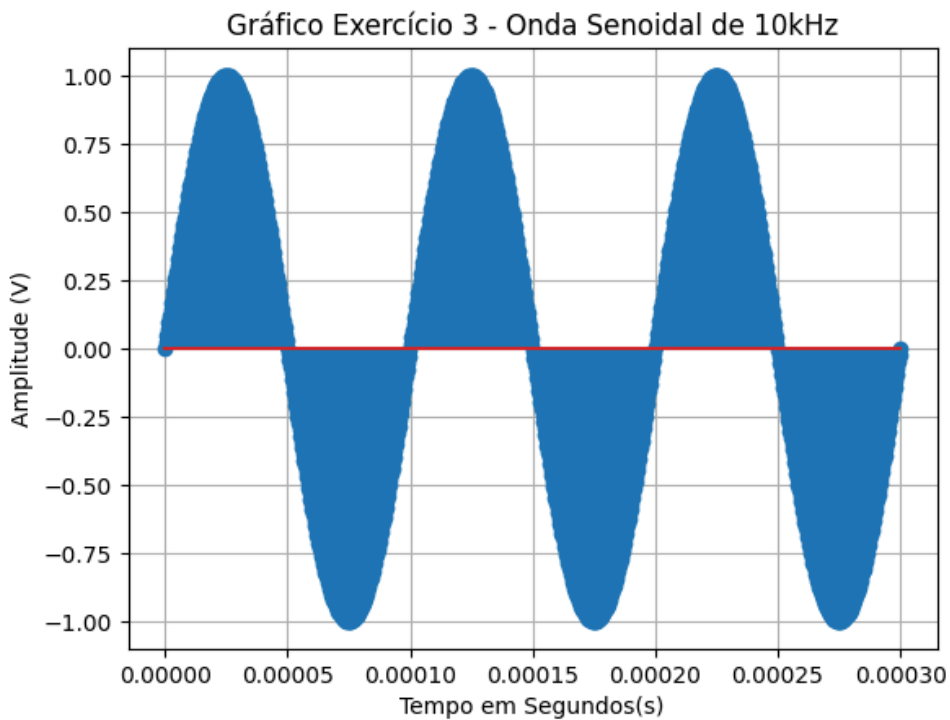
Gráfico Gerado com 100 amostras:



No gráfico acima foi utilizada a função stem e suas demais funções de apresentação gráfica e foram utilizadas 100 amostras, como visto acima suas 100 bolinhas nos 3 períodos. A tabela seria a mesma do experimento 2. Na minha opinião deveriam haver no mínimo 100 amostras para que possa ser representado o sinal gerado com maior fidelidade à sua forma.



No gráfico acima temos 36 amostras. Analisando, percebemos que perdemos certa fidelidade na forma da onda.



Aqui temos outro exemplo de um gráfico gerado com 800 amostras, que fica totalmente preenchido, perdendo certa precisão do gráfico também.