TP1 - PSF

Autor: William's Limonchi Sandoval

1. Ejercicio 1:

Código: En la Figura 1 y Figura 2 se muestra el código de las funciones Senoidal, Cuadrada y Triangular.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

#PARTE 1
def senoidal(frecuencia_muestreo, frecuencia, amplitud, muestras, fase):
    n = np.arange(muestras)
    seno = amplitud * np.sin((2 * np.pi* frecuencia * n /frecuencia_muestreo) + fase)
    return (seno + amplitud)/2

def cuadrada(frecuencia_muestreo,frecuencia,amplitud,muestras):
    def_cuadrado = 0.5
    n = np.arange(muestras)
    cuadra = amplitud * signal.square((2*np.pi * frecuencia * n / frecuencia_muestreo),def_cuadrado)
    return (cuadra + amplitud)/2

def triangular(frecuencia_muestreo, frecuencia, amplitud, muestras):
    def_triang = 0.5
    n = np.arange(muestras)
    trian = amplitud * signal.sawtooth((2 * np.pi * frecuencia * n / frecuencia_muestreo), def_triang)
    return (trian + amplitud)/2
```

Figura 1. Se incluyen las librerías y se realizaron las funciones.

```python
if __name__ == '__main__':
    #PARTE 1
    print("TP1 Parte 1")
    sen = senoidal(100000, 250, 1, 1000, 0)
    plt.title("Ejercicio 1 - Senoidal")
    plt.plot(sen)
    plt.grid()
    plt.show()

    cua = cuadrada(100000, 250, 1, 1000)
    plt.title("Ejercicio 1 - Cuadrada")
    plt.plot(cua)
    plt.grid()
    plt.show()

    tri = triangular(100000, 250, 1, 1000)
    plt.title("Ejercicio 1 - Triangular")
    plt.plot(tri)
    plt.grid()
    plt.show()
```

Figura 2. Se incluye el código para mostrar las funciones.

En la Figura 3 se muestra la señal Senoidal.

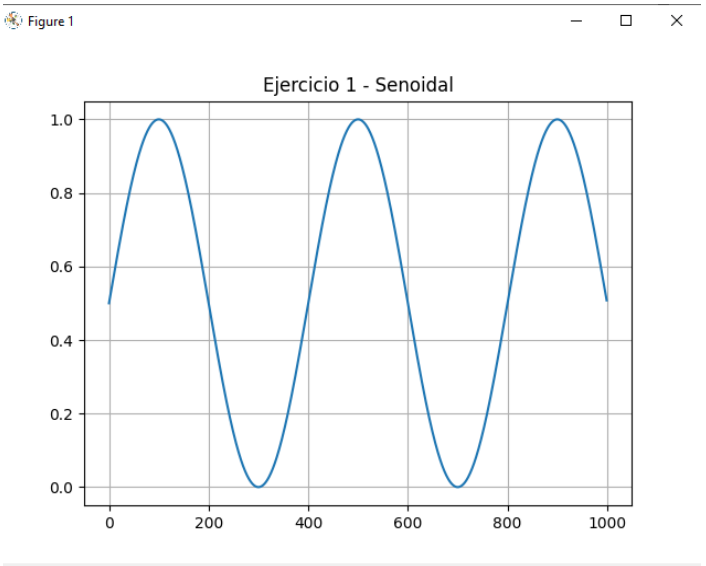

Figura 3. Gráfica de la señal senoidal.

En la Figura 4 se muestra la señal Cuadrada.

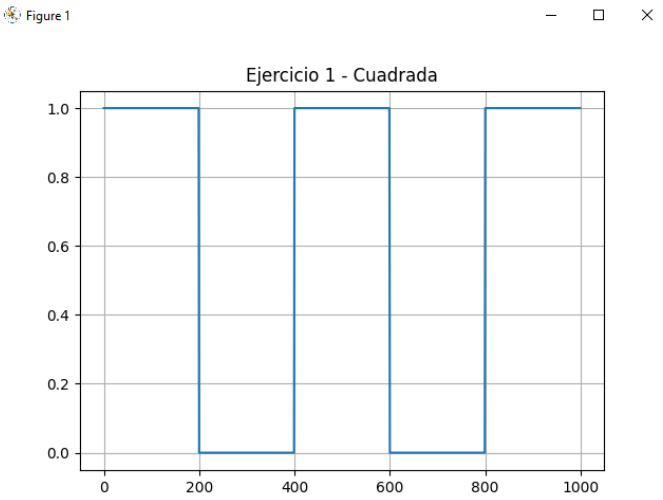

Figura 4. Gráfica de la señal cuadrada.

En la Figura 5 se muestra la señal Triangular.

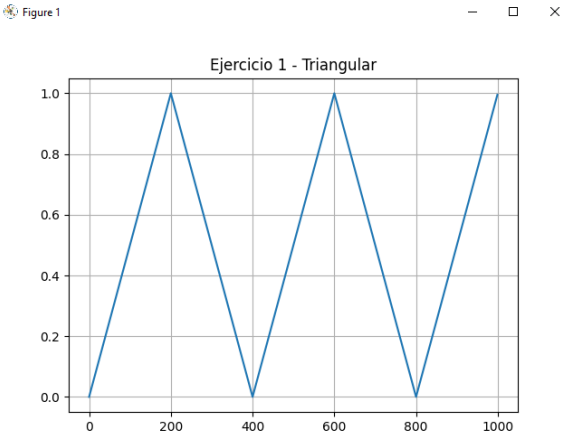

Figura 5. Gráfica de la señal senoidal.

2. Ejercicio 2.1:

En la Figura 6 se muestra como se agregó la frecuencia de muestreo (fs), número de muestras (N), fase y amplitud (amp). Además se crean dos señales senoidales.

```
#PARTE 2
fs = 1000
N = 1000
fase = 0
amp = 1

#PARTE 2.1
print("TP1 Parte 2.1")
senal1 = senoidal(fs, 0.1*fs, amp, N, fase)
senal2 = senoidal(fs, 1.1*fs, amp, N, fase)
plt.title("Ejercicio 2.1 ")
plt.plot(senal1, label = '0.1 fs')
plt.plot(senal2, label = '1.1 fs')
plt.grid()
plt.legend()
plt.show()
```

Figura 6. Gráfica del código.

En la Figura 7 se muestran las señales generadas superpuestas y no se distingue ninguna de las dos.

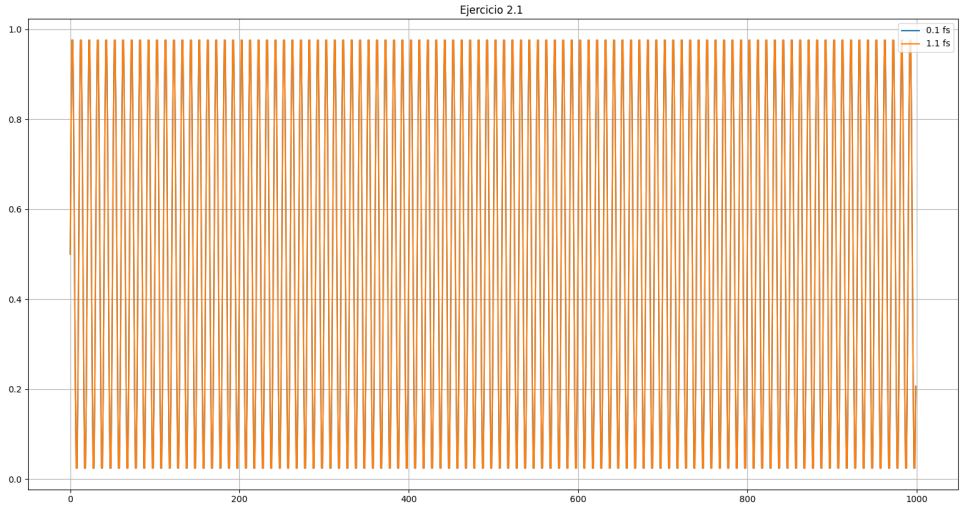

Figura 7. Gráfica de las señales generadas.

3. Ejercicio 2.2:

En la Figura 8 se muestra de las señales con frecuencia 0.49*fs y 1.51*fs.

```
#PARTE 2.2
print("TP1 Parte 2.2")
senal1 = senoidal(fs, 0.49*fs, amp, N, fase)
senal2 = senoidal(fs, 1.51*fs, amp, N, fase)
plt.title("Ejercicio 2.2 ")
plt.plot(senal1, label = '0.49 fs')
plt.plot(senal2, label = '1.51 fs')
plt.grid()
plt.legend()
plt.show()
```

Figura 8. Gráfica del código.

En la Figura 8 se muestran las señales generadas superpuestas y se logra distinguir cada una de ellas.