

Genere un paquete/modulo o archivo que permita sintetizar las señales descriptas en la imagen. Se espera un link a un pdf con el código y algunas capturas que validen su funcionamiento.

El código se encuentra en el siguiente repositorio.

https://github.com/jdalvaradocol/PSF_MSE.git

```
import numpy as np
import matplotlib.pyplot as plt
np.set_printoptions(precision=3, suppress=False)
```

```
class return_values:
    def __init__(self, t,y):
        self.t = t
        self.y = y
```

```
def Generador(Fo,Fs,A,N,fi,tipo):
```

```
    Tm = 1/Fs
    w = 2*np.pi*Fo
    t = np.linspace(0,Tm*N,int(N))
    y = np.zeros(len(t))
```

```
    if tipo == 'senoidal':
```

```
        y = A * np.sin(w*t+fi)
```

```
    if tipo == 'coseno':
```

```
        y = A * np.cos(w*t+fi)
```

```
    if tipo == 'cuadrada':
```

```
        To = 1/Fo
        ciclo = Fo/2
        x = 1
```

```
    for i in range( 0, len(t), 1):
```

```
        if t[i] > To * x:
```

```
            ciclo = ciclo + To
            x = x + 1
```

```
        if t[i] < ciclo:
```

```
            y[i] = A
```

```
        elif t[i] > ciclo:
```

```
            y[i] = 0
```

```
    if tipo == 'sierra':
```

```

To = 1/Fo
dA = A / Fs
amp = 0
x = 1

```

```

for i in range( 0, len(t), 1):

```

```

    if t[i] > To * x:

```

```

        amp = 0
        x = x + 1

```

```

        amp = amp + dA
        y[i] = amp

```

```

if tipo == 'triangular':

```

```

    To = 1/Fo
    ciclo = Fo/2
    dA = (A / Fs) * 2
    amp = 0
    x = 1

```

```

for i in range( 0, len(t), 1):

```

```

    if t[i] > To * x:

```

```

        ciclo = ciclo + To
        x = x + 1

```

```

    if t[i] < ciclo:

```

```

        y[i] = amp
        amp = amp + dA

```

```

    elif t[i] > ciclo:

```

```

        y[i] = amp
        amp = amp - dA

```

```

x = return_values(t,y)

```

```

return x

```

Generador de señales senoidal, Cuadrada, Triangular.

```

Fo = 1    # Frecuencia de la señal.
Fs = 10   # Frecuencia de Muestreo.
fi = 0    # Angulo desface en radianes.
A = 1     # Amplitud de la señal.
N = 50    # Numero de Muestras.

```

tipo = 'triangular' # Especifica el tipo de señal a generar. se puede definir senoidal, cuadrada, coseno, triangular, 'sierra'.

```

signal= Generador(Fo,Fs,A,N,fi,tipo)

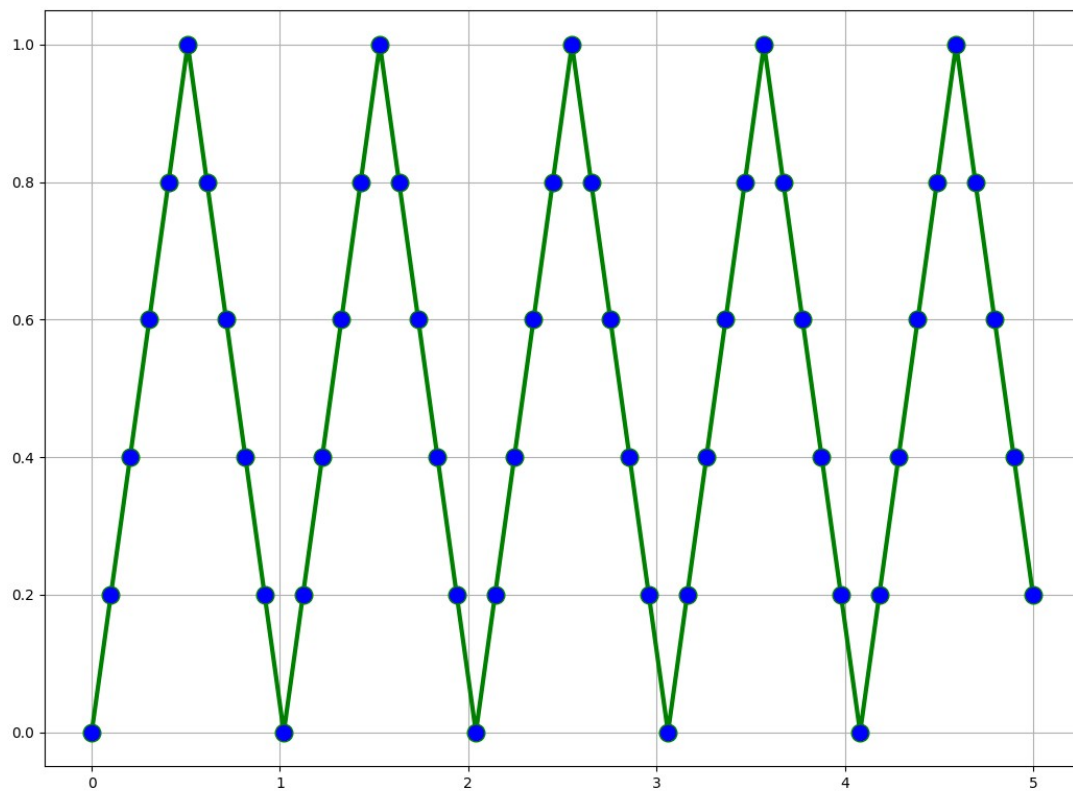
```

```
#print(signal.t)
#print(signal.y)

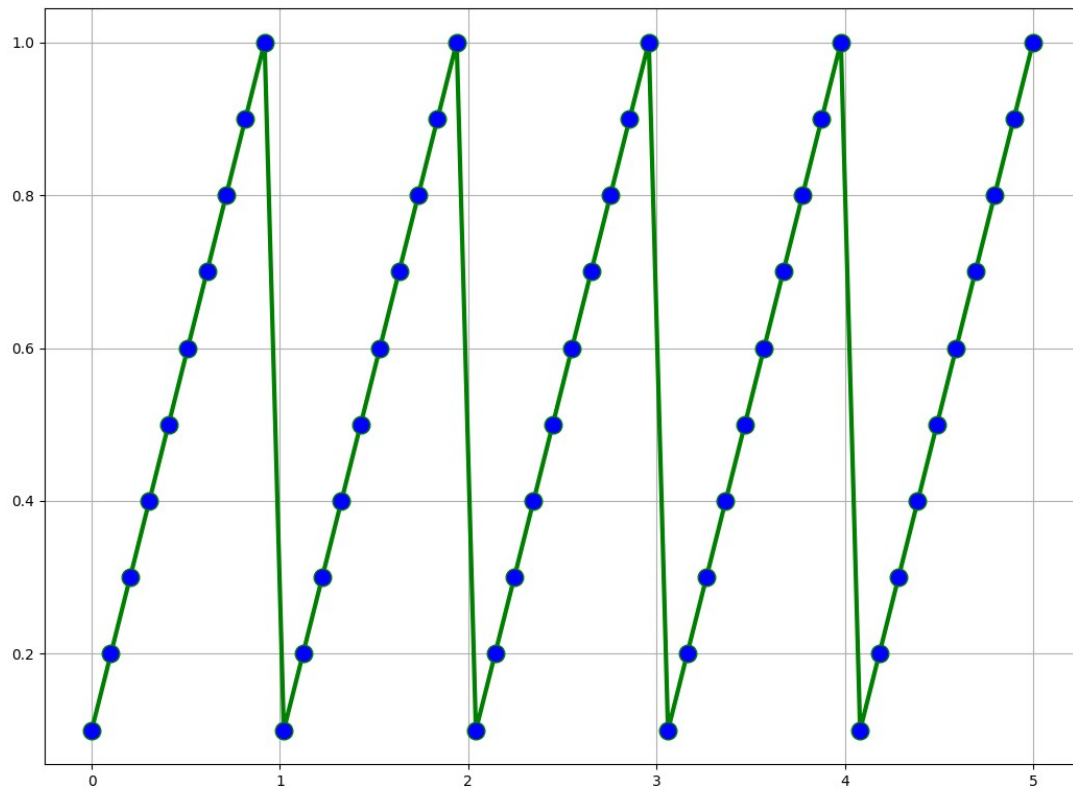
#grafico con matplotlib
plt.plot(signal.t, signal.y, color='green', linestyle='solid', linewidth = 3, marker='o',
markerfacecolor='blue', markersize=12)
plt.grid(True)
plt.show()
```

CAPTURA DE SEÑALES GENERADA

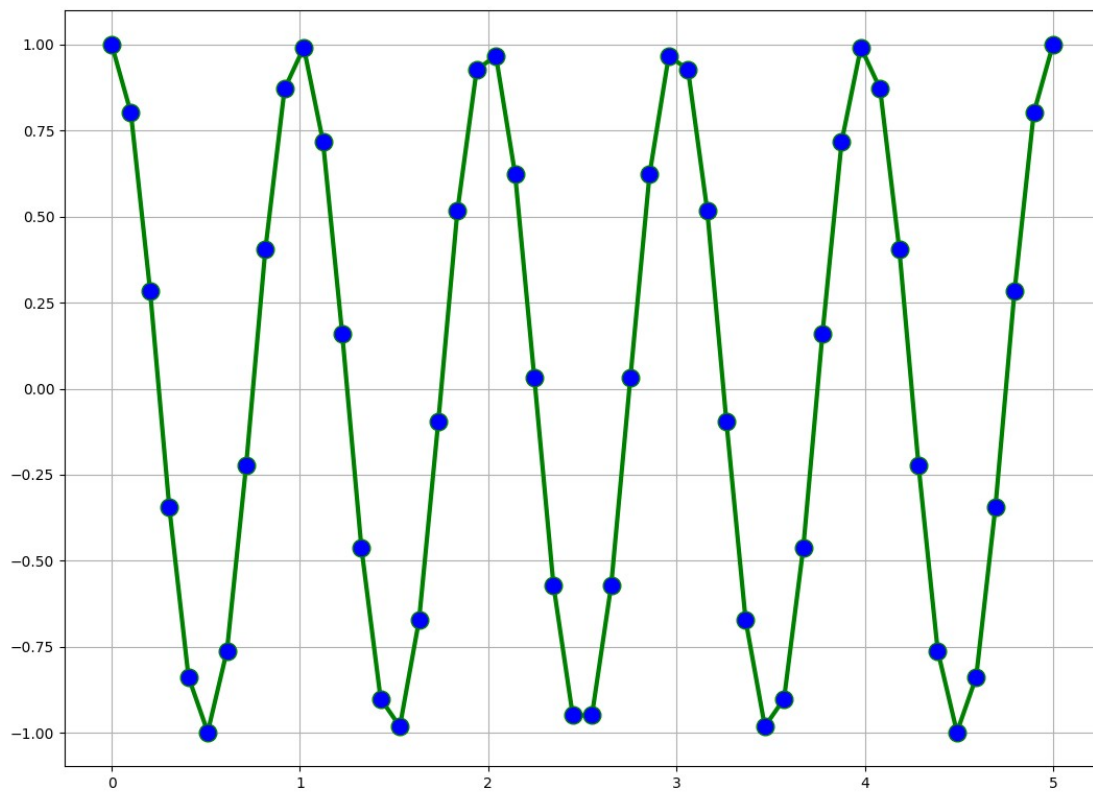
$F_o = 1\text{Hz}$, $F_s = 10\text{Hz}$, $A = 1$, $N = 50$ - Triangular



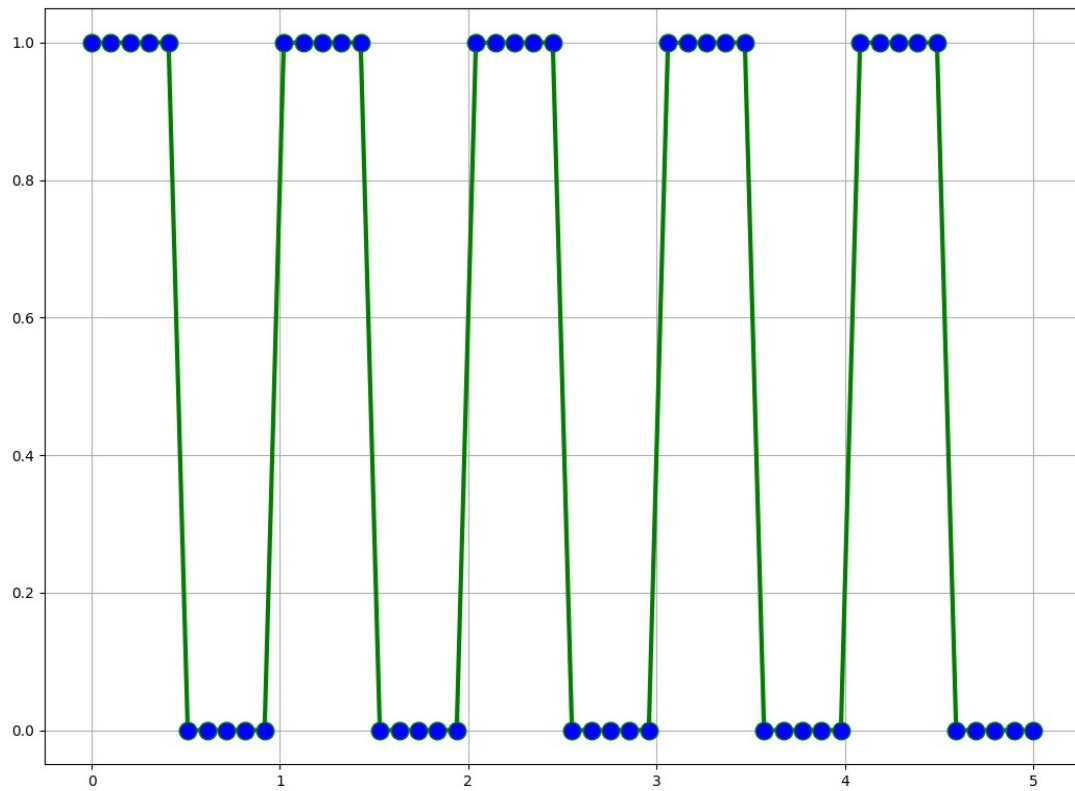
$F_o = 1\text{Hz}$, $F_s = 10\text{Hz}$, $A = 1$, $N = 50$ - Sierra



Fo = 1Hz, Fs = 10Hz, A = 1, N = 50 - Coseno



Fo = 1Hz, Fs = 10Hz, A = 1, N = 50 - Cuadrada



Fo = 1Hz, Fs = 10Hz, A = 1, N = 50 - Senoidal

