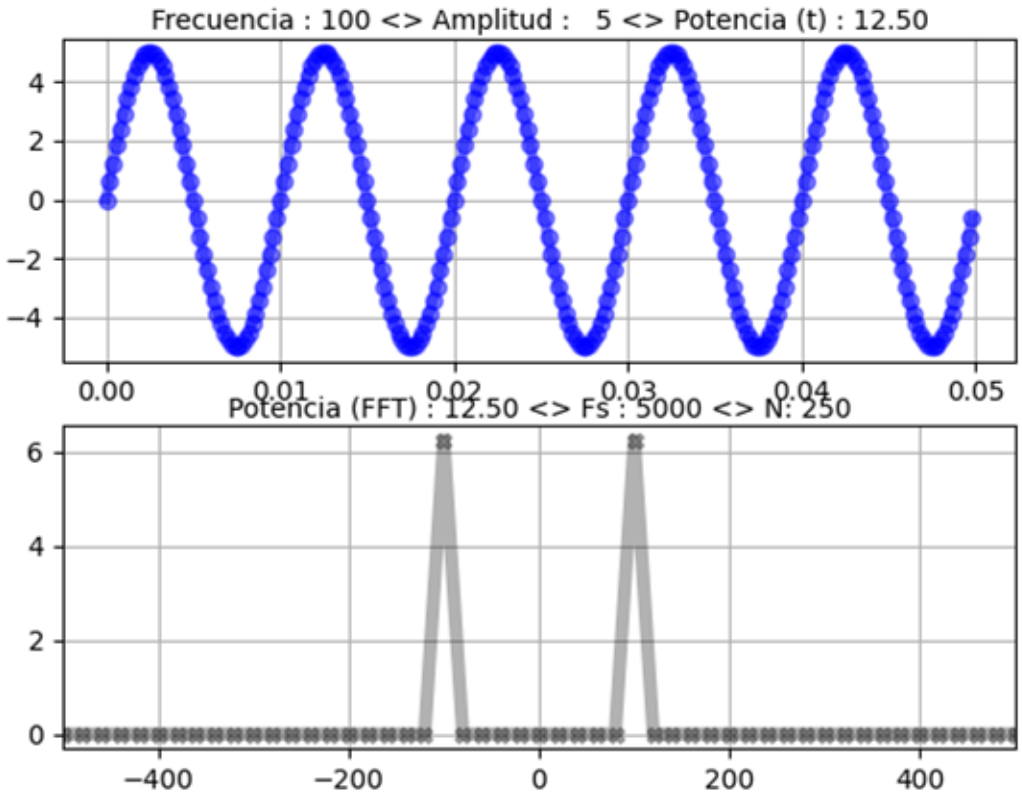
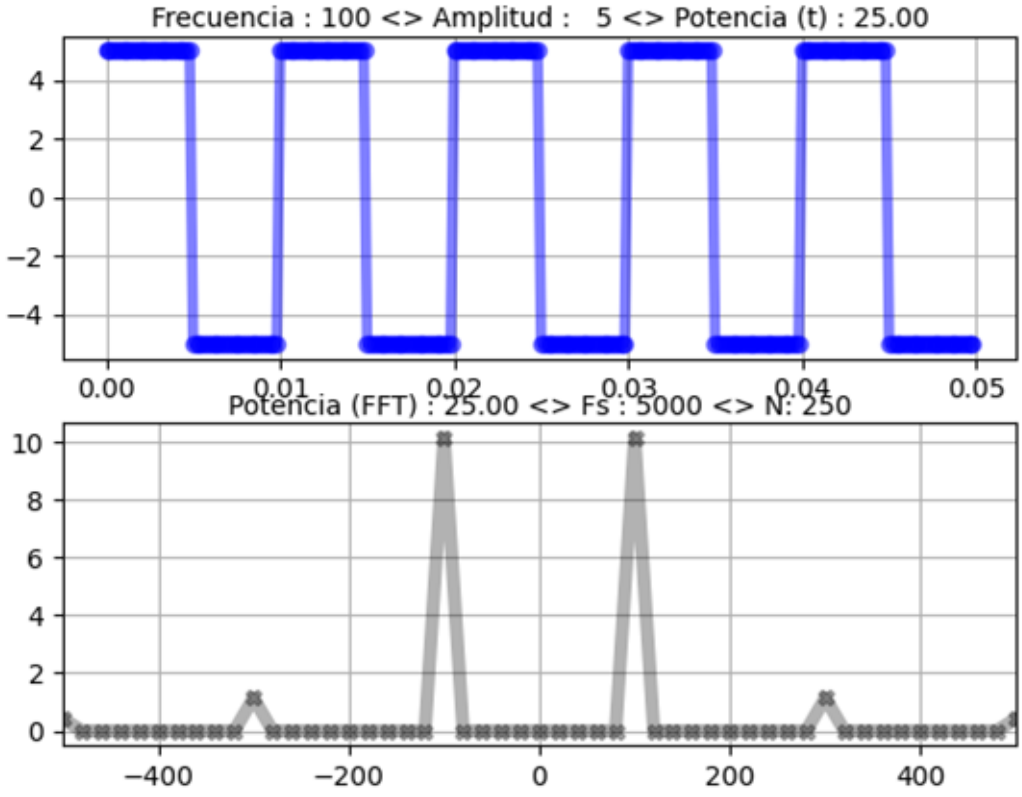
Grafique las siguientes señales lado a lado con su respectivo espectro en frecuencias: 1) Senoidal. 2) Cuadrada. 3) Triangular 4) Delta en t=0. Indicando en cada caso los siguientes parámetros (si corresponde) : 1) Frecuencia. B) Amplitud. C) Potencia promedio. D) Fs. E) N. 5) Pegue el link a un pdf con los códigos, gráficos y comentarios.

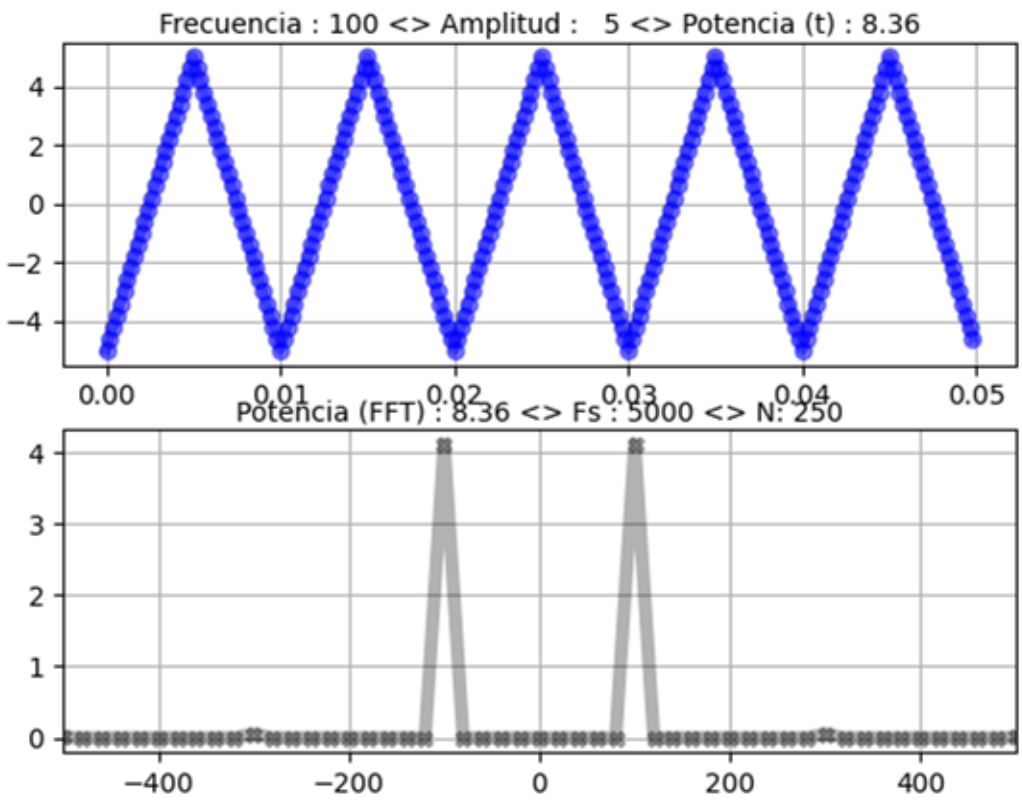
```
import numpy as np
import scipy.signal as sc
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

class return_values_Pm:
    def __init__(self, Pm_t,Pm_fft):
        self.Pm_t = Pm_t
        self.Pm_fft = Pm_fft

class return_values_FFT:
    def __init__(self, fftData,fftsignal):
        self.fftData = fftData
        self.fftsignal = fftsignal

def FFT(signalData):
    fftData = np.fft.fft(signalData)
    fftData = np.fft.fftshift(fftData)
    fftsignal = np.abs(fftData/N)**2

    x = return_values_FFT(fftData,fftsignal)

    return x

def PM(signalData, fftData):
    Pm_t = np.sum(np.abs(signalData)**2) / len(signalData)
    Pm_fft = np.sum(np.abs(fftData / len(fftData))**2)

    x = return_values_Pm(Pm_t,Pm_fft)

    return x

#--------------------------------------
fs         = 5000
N          = 250
signalFrec = 100
Amp        = 5

#--------------------------------------
nData      = np.arange(0,N,1) #arranco con numeros enteros para evitar errores de float
tData      = nData/fs
fData      = nData*(fs/((N)-(N)%2))-fs/2

#------------SIGNAL-------------------------

signalData1 = Amp*np.sin         (2*np.pi*signalFrec*nData*1/fs)
signalData2 = Amp*sc.square      (2*np.pi*signalFrec*nData*1/fs,0.5)
signalData3 = Amp*sc.sawtooth    (2*np.pi*signalFrec*nData*1/fs,0.5)
signalData4 = np.array([Amp if n == 0 else 0 for n in nData])

fft1       = FFT(signalData1)
```

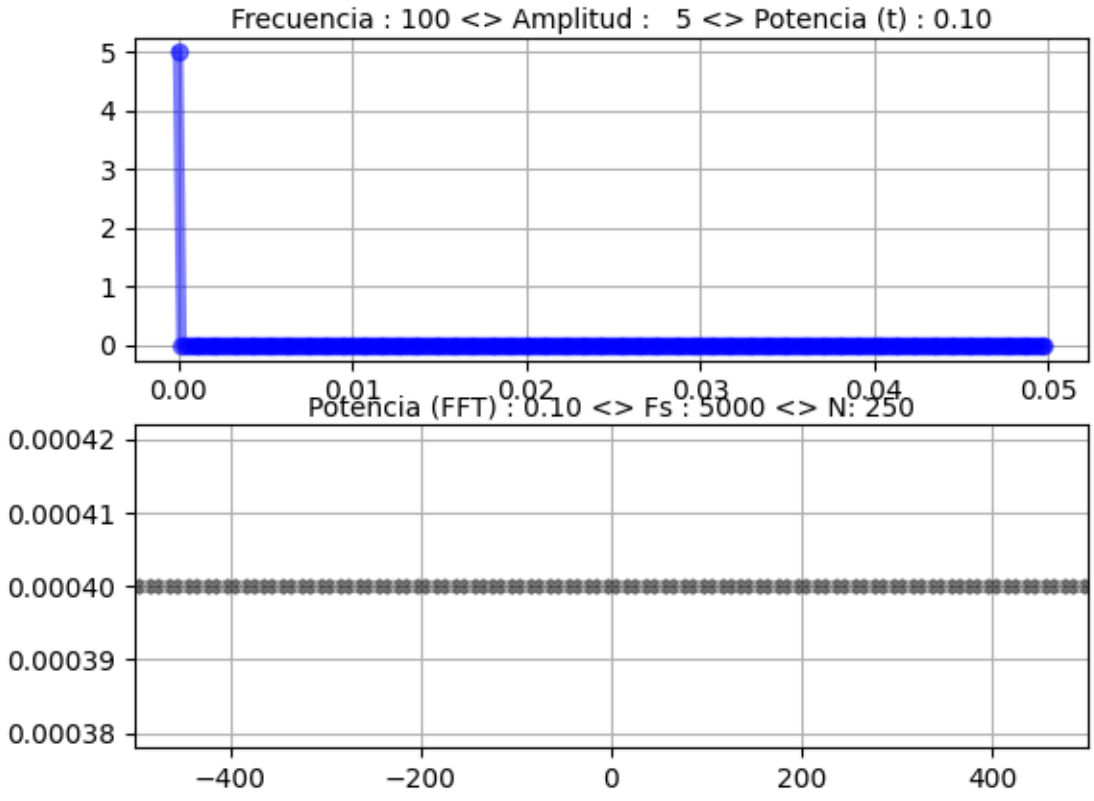
```
fftData1   = fft1.fftData
fftsignal1 = fft1.fftsignal
Pm1        = PM(signalData1, fftData1)
Pm_t1      = Pm1.Pm_t
Pm_fft1     = Pm1.Pm_fft

## Figura 1

plt.figure(1)
# Grafica de la señal 1
signalAxe  = plt.subplot(2,1,1)
signalRLn, = plt.plot(tData,np.real(signalData1),'b-o',linewidth=4,alpha=0.5,label="real")
signalAxe.set_title( "Frecuencia : {0:3d} <> Amplitud : {1:3d} <> Potencia (t) :
{2:.2f}".format(signalFrec, Amp, Pm_t1), rotation=0,fontsize=10,va="center")
signalAxe.grid(True)
# Grafica de la FFT 1
fftAxe     = plt.subplot(2,1,2)
fftAbsLn,  = plt.plot(fData,fftsignal1,'k-X' ,linewidth  = 5,alpha = 0.3,label="Potencia")
fftAxe.set_title("Potencia (FFT) : {0:.2f} <> Fs : {1:3d} <> N: {2:3d}".for-
mat(Pm_fft1,fs,N),rotation=0,fontsize=10,va="center")
plt.xlim([-signalFrec*5, signalFrec*5])
fftAxe.grid(True)
#-------------------------------------

fft2       = FFT(signalData2)
fftData2   = fft2.fftData
fftsignal2 = fft2.fftsignal
Pm2        = PM(signalData2, fftData2)
Pm_t2      = Pm2.Pm_t
Pm_fft2    = Pm2.Pm_fft

## Figura 2
plt.figure(2)
# Grafica de la señal 2
signalAxe  = plt.subplot(2,1,1)
signalRLn, = plt.plot(tData,np.real(signalData2),'b-o',linewidth=4,alpha=0.5,label="real")
signalAxe.set_title( "Frecuencia : {0:3d} <> Amplitud : {1:3d} <> Potencia (t) :
{2:.2f}".format(signalFrec, Amp, Pm_t2), rotation=0,fontsize=10,va="center")
signalAxe.grid(True)
# Grafica de la FFT 2
fftAxe     = plt.subplot(2,1,2)
fftAbsLn,  = plt.plot(fData,fftsignal2,'k-X' ,linewidth  = 5,alpha = 0.3,label="Potencia")
fftAxe.set_title("Potencia (FFT) : {0:.2f} <> Fs : {1:3d} <> N: {2:3d}".for-
mat(Pm_fft2,fs,N),rotation=0,fontsize=10,va="center")
plt.xlim([-signalFrec*5, signalFrec*5])
fftAxe.grid(True)
#-------------------------------------

fft3       = FFT(signalData3)
fftData3   = fft3.fftData
fftsignal3 = fft3.fftsignal
Pm3        = PM(signalData3, fftData3)
Pm_t3      = Pm3.Pm_t
Pm_fft3    = Pm3.Pm_fft

## Figura 3
plt.figure(3)
```

```
# Grafica de la señal 3
signalAxe  = plt.subplot(2,1,1)
signalRLn, = plt.plot(tData,np.real(signalData3),'b-o',linewidth=4,alpha=0.5,label="real")
signalAxe.set_title( "Frecuencia : {0:3d} <> Amplitud : {1:3d} <> Potencia (t) :
{2:.2f}".format(signalFrec, Amp, Pm_t3), rotation=0,fontsize=10,va="center")
signalAxe.grid(True)
# Grafica de la FFT 3
fftAxe     = plt.subplot(2,1,2)
fftAbsLn,  = plt.plot(fData,fftsignal3,'k-X' ,linewidth  = 5,alpha = 0.3,label="Potencia")
fftAxe.set_title("Potencia (FFT) : {0:.2f} <> Fs : {1:3d} <> N: {2:3d}".for-
mat(Pm_fft3,fs,N),rotation=0,fontsize=10,va="center")
plt.xlim([-signalFrec*5, signalFrec*5])
fftAxe.grid(True)
#--------------------------------------

fft4       = FFT(signalData4)
fftData4   = fft4.fftData
fftsignal4 = fft4.fftsignal
Pm4        = PM(signalData4, fftData4)
Pm_t4      = Pm4.Pm_t
Pm_fft4    = Pm4.Pm_fft

## Figura 4
plt.figure(4)
# Grafica de la señal 4
signalAxe  = plt.subplot(2,1,1)
signalRLn, = plt.plot(tData,np.real(signalData4),'b-o',linewidth=4,alpha=0.5,label="real")
signalAxe.set_title( "Frecuencia : {0:3d} <> Amplitud : {1:3d} <> Potencia (t) :
{2:.2f}".format(signalFrec, Amp, Pm_t4), rotation=0,fontsize=10,va="center")
signalAxe.grid(True)
# Grafica de la FFT 4
fftAxe     = plt.subplot(2,1,2)
fftAbsLn,  = plt.plot(fData,fftsignal4,'k-X' ,linewidth  = 5,alpha = 0.3,label="Potencia")
fftAxe.set_title("Potencia (FFT) : {0:.2f} <> Fs : {1:3d} <> N: {2:3d}".for-
mat(Pm_fft4,fs,N),rotation=0,fontsize=10,va="center")
plt.xlim([-signalFrec*5, signalFrec*5])
fftAxe.grid(True)
#--------------------------------------

plt.show()
```


Frecuencia : 100 <> Amplitud : 5 <> Potencia (t) : 12.50
Potencia (FFT) : 12.50 <> Fs : 5000 <> N: 250


Frecuencia : 100 <> Amplitud : 5 <> Potencia (t) : 25.00
Potencia (FFT) : 25.00 <> Fs : 5000 <> N: 250


Frecuencia : 100 <> Amplitud : 5 <> Potencia (t) : 8.36
Potencia (FFT) : 8.36 <> Fs : 5000 <> N: 250


Frecuencia : 100 <> Amplitud : 5 <> Potencia (t) : 0.10
Potencia (FFT) : 0.10 <> Fs : 5000 <> N: 250

Comentarios:

Para la señal senoidal se obtiene una FFT con un solo componente (armónico) de frecuencia en la parte positiva y su equivalente en la parte negativa.

Para la señal cuadrada se obtiene una FFT con un componente principal y un armónico de frecuencia en la parte positiva y su equivalente en la parte negativa.

Para la señal triangula se obtiene una FFT con un componente principal y un armónico de frecuencia en la parte positiva y su equivalente en la parte negativa.

Para la señal delta dirac se obtiene una FFT con una constante. En este caso esta señal no es periódica y este valor equivale a la potencia instantánea de la señal.