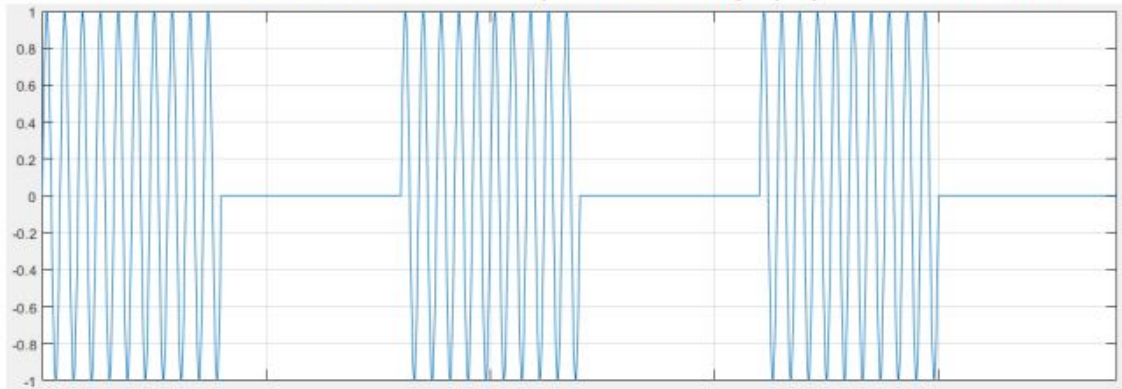


Ejercicio 5

La figura presenta solo tres “pulsos de RF” pero la señal es periódica, es decir, es una sucesión infinita de estos pulsos de RF. Se pide hallar:

- El espectro del módulo de la transformada de Fourier
 $f_c = 20 \text{ MHz}$ $t_{\text{activo}} = 0,5 \text{ uS}$ $t_{\text{reposo}} = 0,5 \text{ uS}$
- Idem al a) pero duplicando la frecuencia de senoidal ($f_c = 40 \text{ MHz}$).
- Idem al a) pero aumentando al doble los tiempos activo y de reposo (1 uS).
- Idem al a) pero aumentando al doble sólo el tiempo activo (1 uS).

Para ambos casos, señale métodos alternativos para resolverlo y aplique uno a su elección.



a)

```
import matplotlib.pyplot as plt
import numpy as np

# Definimos las constantes
fc = 20e6      # Frecuencia del pulso RF
t_on = 0.5e-6  # Tiempo activo
A = 1          # Amplitud del pulso RF
T = 1e-6       # Periodo del pulso
M = 10000      # Cantidad de muestras por periodo

# Creamos el vector de tiempo
t = np.linspace(0, 10*T, 10*M)

# Definimos el pulso RF
pRF = A*np.sin(2*np.pi*fc*t)

# Creamos el vector de la señal
x = np.zeros_like(t)
```

```

# Generamos los pulsos RF
for i in range(len(t)):
    if i % int(T*(int(M/T))) < int(t_on*(int(M/T))):
        x[i] = pRF[i]

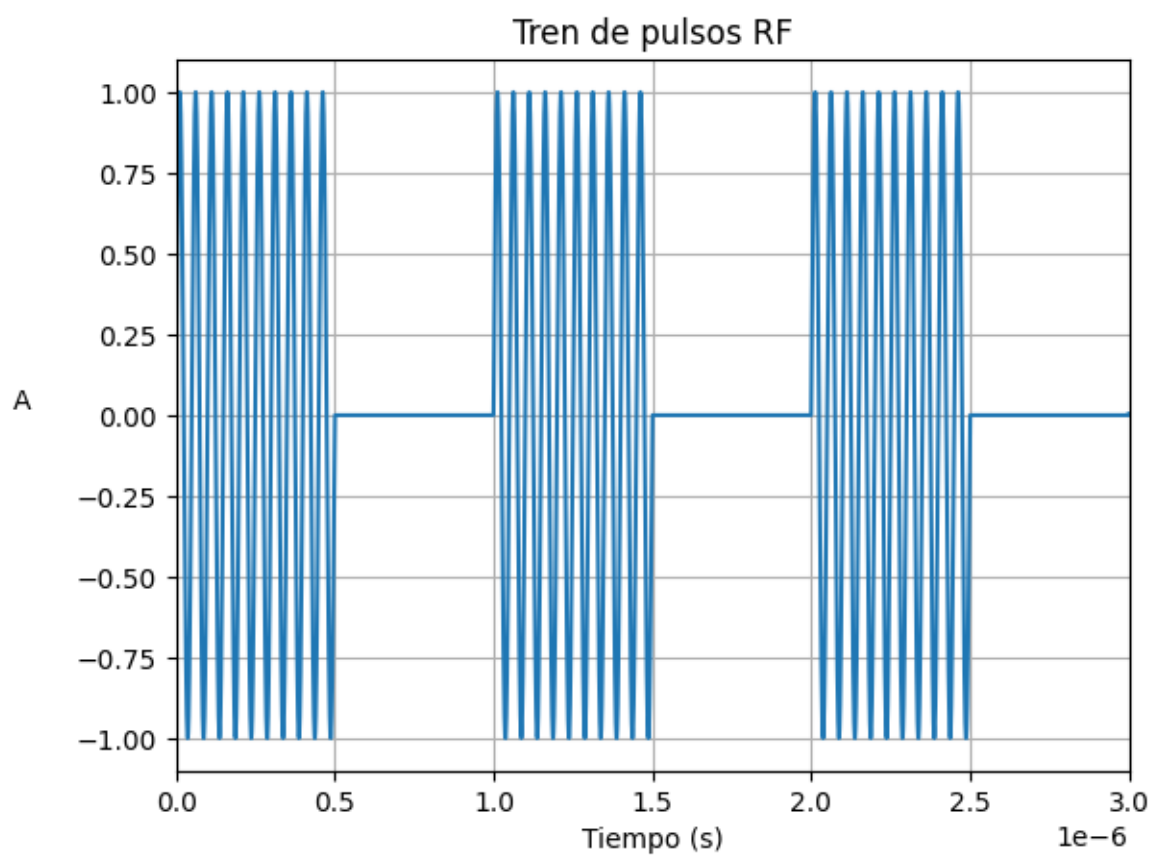
# Graficamos la señal
plt.figure()
plt.plot(t, x)
plt.xlim(0,3e-6)
plt.title('Tren de pulsos RF')
plt.xlabel('Tiempo (s)')
plt.ylabel('A', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

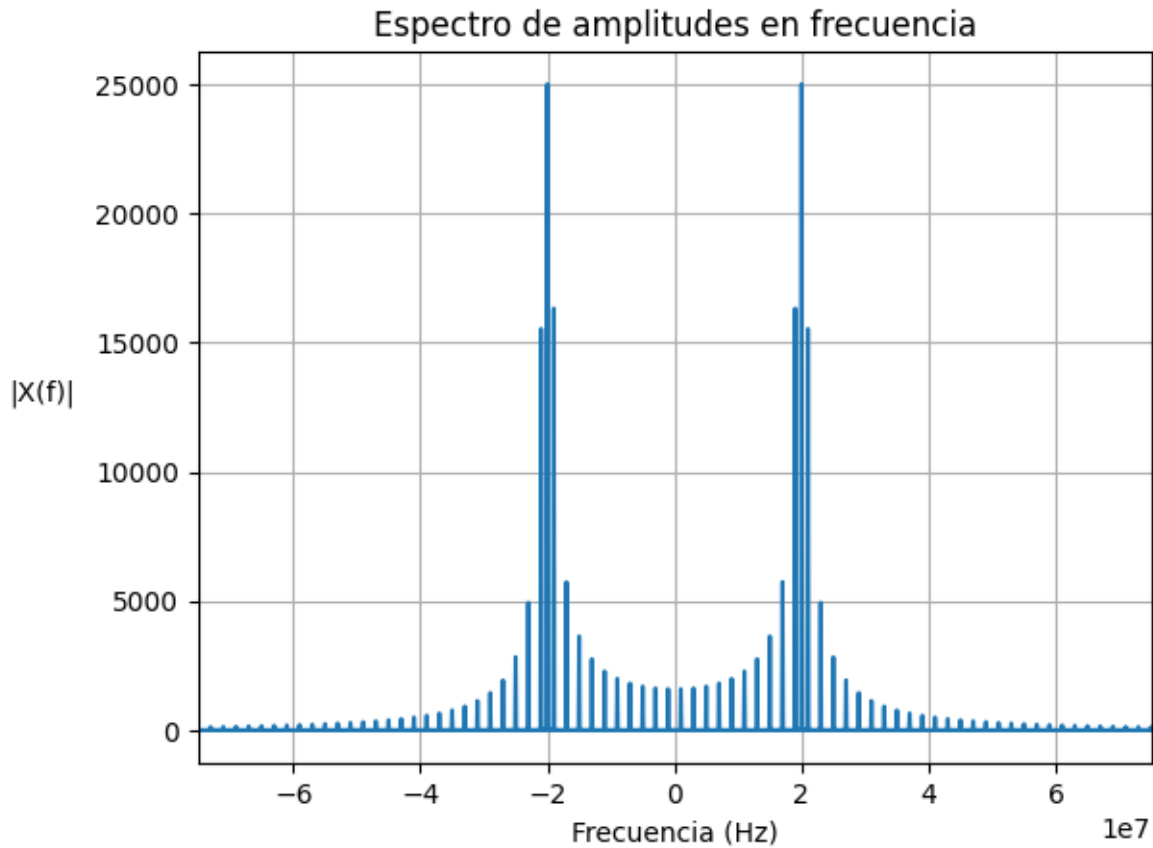
# Calculamos la transformada de Fourier y su espectro de amplitudes
X = np.fft.fft(x)
X_amp = np.abs(X)

# Creamos el vector de frecuencias
freq = np.fft.fftfreq(len(t), t[1] - t[0])

# Graficamos el espectro de amplitudes
plt.figure()
plt.plot(freq, X_amp)
plt.xlim(-0.75e8,0.75e8)
plt.title('Espectro de amplitudes en frecuencia')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('|X(f)|', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

```





5.b)

```
import matplotlib.pyplot as plt
import numpy as np

# Definimos las constantes
fc = 40e6      # Frecuencia del pulso RF
t_on = 0.5e-6  # Tiempo activo
A = 1          # Amplitud del pulso RF
T = 1e-6       # Periodo del pulso
M = 10000      # Cantidad de muestras por periodo

# Creamos el vector de tiempo
t = np.linspace(0, 10*T, 10*M)

# Definimos el pulso RF
pRF = A*np.sin(2*np.pi*fc*t)
```

```

# Creamos el vector de la señal
x = np.zeros_like(t)

# Generamos los pulsos RF
for i in range(len(t)):
    if i % int(T*(int(M/T))) < int(t_on*(int(M/T))):
        x[i] = pRF[i]

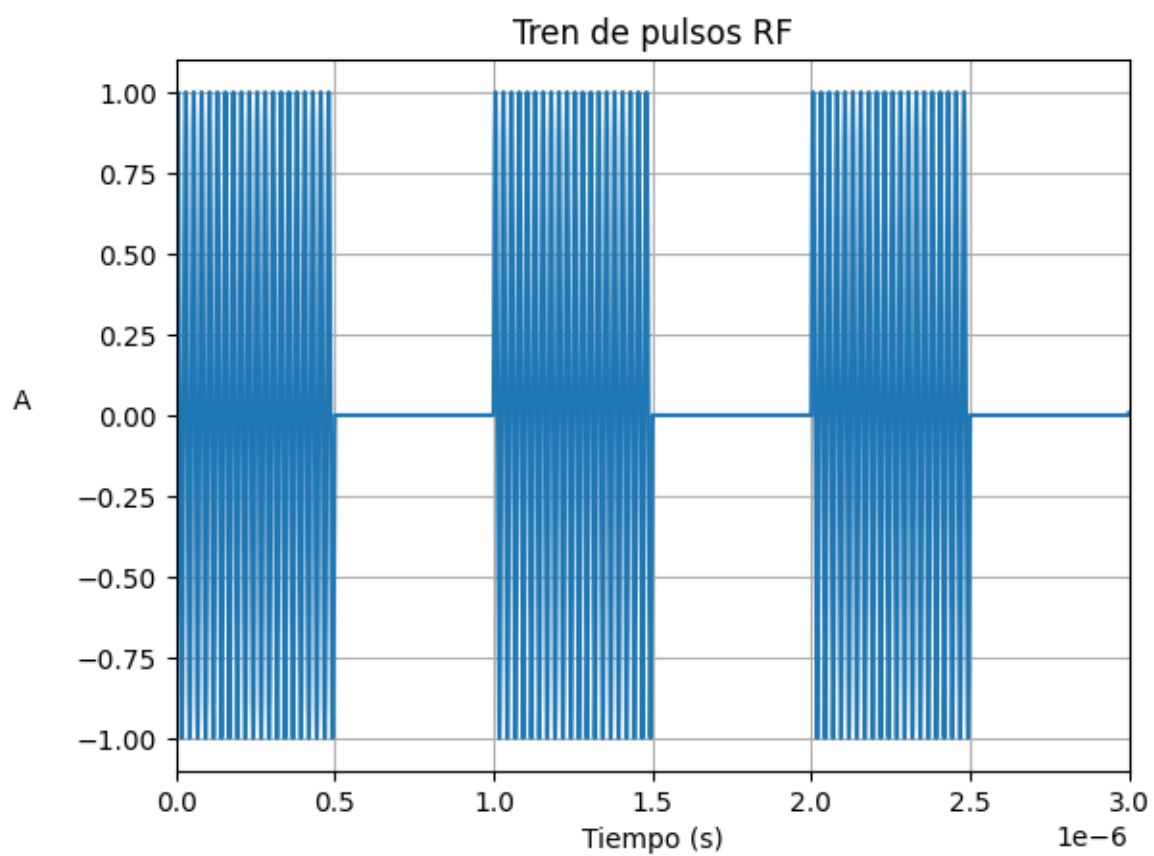
# Graficamos la señal
plt.figure()
plt.plot(t, x)
plt.xlim(0,3e-6)
plt.title('Tren de pulsos RF')
plt.xlabel('Tiempo (s)')
plt.ylabel('A', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

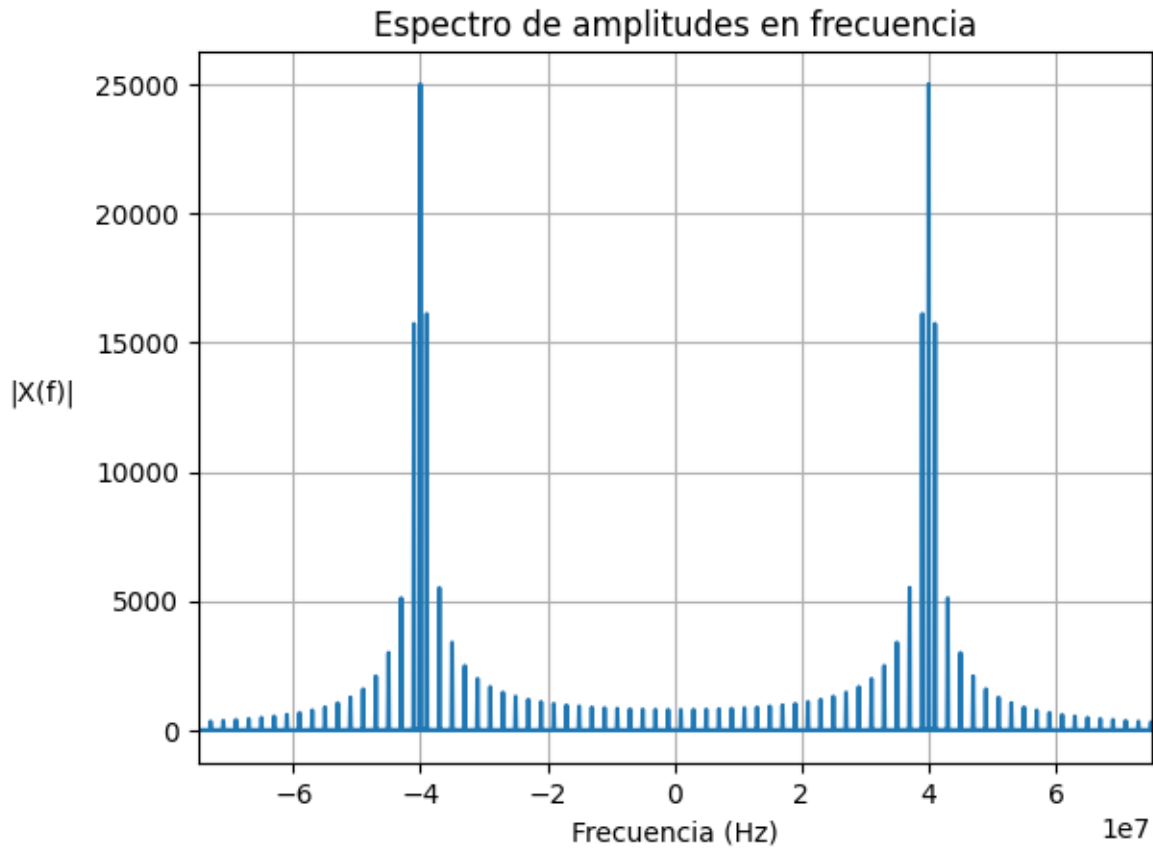
# Calculamos la transformada de Fourier y su espectro de amplitudes
X = np.fft.fft(x)
X_amp = np.abs(X)
# Y_amp = Y.real

# Creamos el vector de frecuencias
freq = np.fft.fftfreq(len(t), t[1] - t[0])

# Graficamos el espectro de amplitudes
plt.figure()
plt.plot(freq, X_amp)
plt.xlim(-0.75e8,0.75e8)
plt.title('Espectro de amplitudes en frecuencia')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('|X(f)|', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

```





5.c)

```
import matplotlib.pyplot as plt
import numpy as np

# Definimos las constantes
fc = 20e6 # Frecuencia del pulso RF
t_on = 1e-6 # Tiempo activo
A = 1 # Amplitud del pulso RF
T = 2e-6 # Periodo del pulso
M = 10000 # Cantidad de muestras por periodo

# Creamos el vector de tiempo
t = np.linspace(0, 10*T, 10*M)

# Definimos el pulso RF
pRF = A*np.sin(2*np.pi*fc*t)
```

```

# Creamos el vector de la señal
x = np.zeros_like(t)

# Generamos los pulsos RF
for i in range(len(t)):
    if i % int(T*(int(M/T))) < int(t_on*(int(M/T))):
        x[i] = pRF[i]

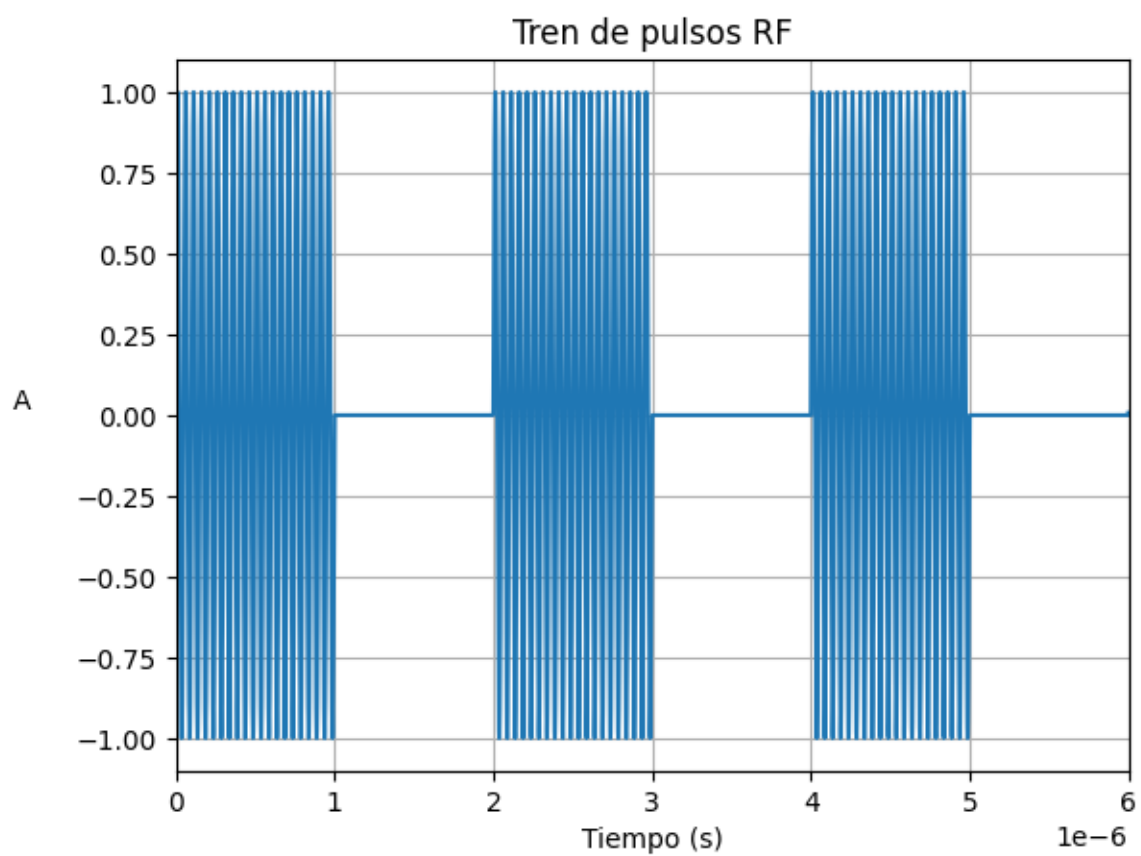
# Graficamos la señal
plt.figure()
plt.plot(t, x)
plt.xlim(0,6e-6)
plt.title('Tren de pulsos RF')
plt.xlabel('Tiempo (s)')
plt.ylabel('A', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

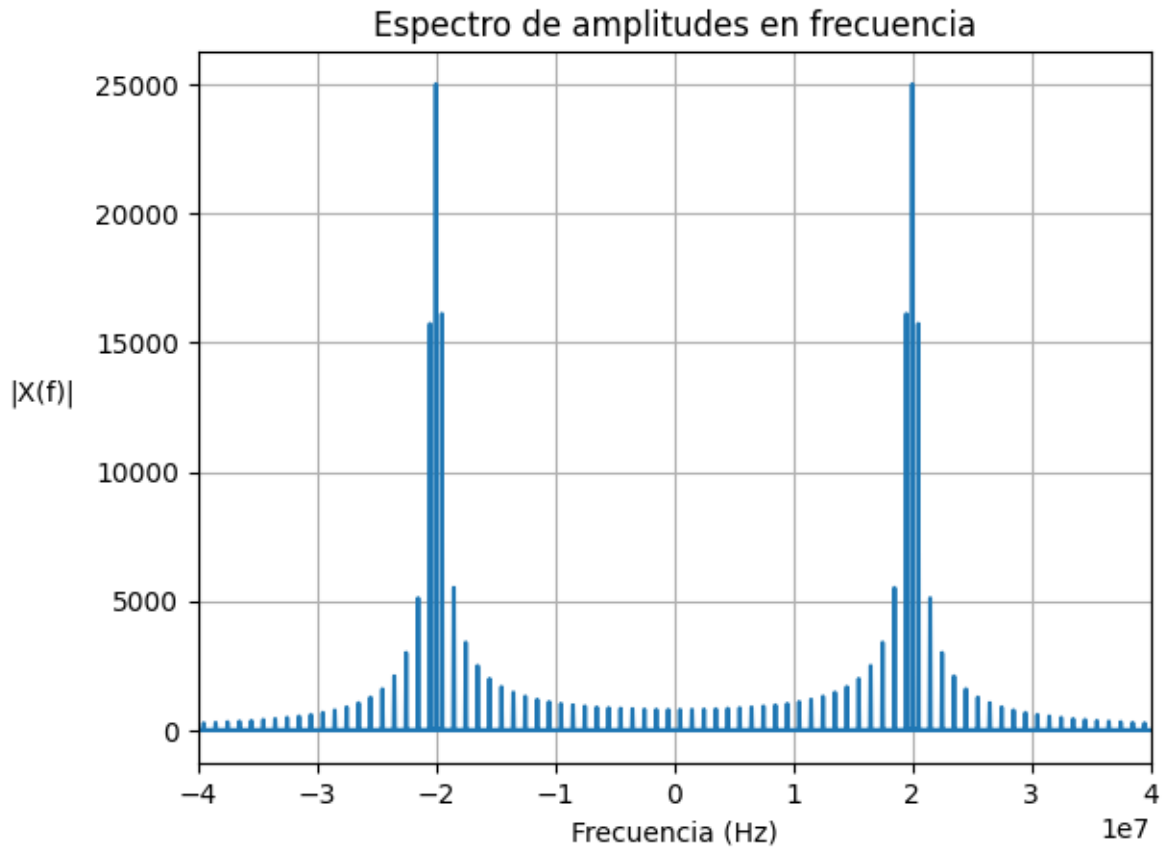
# Calculamos la transformada de Fourier y su espectro de amplitudes
X = np.fft.fft(x)
X_amp = np.abs(X)
# Y_amp = Y.real

# Creamos el vector de frecuencias
freq = np.fft.fftfreq(len(t), t[1] - t[0])

# Graficamos el espectro de amplitudes
plt.figure()
plt.plot(freq, X_amp)
plt.xlim(-4e7,4e7)
plt.title('Espectro de amplitudes en frecuencia')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('|X(f)|', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

```



5.d)

```
import matplotlib.pyplot as plt
import numpy as np

# Definimos las constantes
fc = 20e6 # Frecuencia del pulso RF
t_on = 1e-6 # Tiempo activo
A = 1 # Amplitud del pulso RF
T = 1.5e-6 # Periodo del pulso
M = 10000 # Cantidad de muestras por periodo

# Creamos el vector de tiempo
t = np.linspace(0, 10*T, 10*M)

# Definimos el pulso RF
pRF = A*np.sin(2*np.pi*fc*t)
```

```

# Creamos el vector de la señal
x = np.zeros_like(t)

# Generamos los pulsos RF
for i in range(len(t)):
    if i % int(T*(int(M/T))) < int(t_on*(int(M/T))):
        x[i] = pRF[i]

# Graficamos la señal
plt.figure()
plt.plot(t, x)
plt.xlim(0,4.5e-6)
plt.title('Tren de pulsos RF')
plt.xlabel('Tiempo (s)')
plt.ylabel('A', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

# Calculamos la transformada de Fourier y su espectro de amplitudes
X = np.fft.fft(x)
X_amp = np.abs(X)
# Y_amp = Y.real

# Creamos el vector de frecuencias
freq = np.fft.fftfreq(len(t), t[1] - t[0])

# Graficamos el espectro de amplitudes
plt.figure()
plt.plot(freq, X_amp)
plt.xlim(-4e7,4e7)
plt.title('Espectro de amplitudes en frecuencia')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('|X(f)|', rotation=0, labelpad=20)
plt.grid(True)
plt.show()

```

