

# Practica 1: Teorema de muestreo

## **1. Objetivos**

Comprender el teorema de muestreo y introducirse en el uso de MATLAB.

## **2. Generación de impulsos**

- Representar las siguientes secuencias:

$$X_1[n] = 0.9 * \delta[n - 5], \quad 1 \leq n \leq 20.$$

$$X_2[n] = 4.5 * \delta[n + 7], \quad -10 \leq n \leq 0.$$

Para representar las 2 secuencias, usaremos el siguiente código:

*%Implementacion Señal X1*

```
L=21;%Longitud
nn1=0:(L-1);%Vector de tiempos
imp1=zeros(L,1);%Matriz de cero con L filas y 1 columna
imp1(6)=0.9;%Localizamos el punto donde queremos el impulso, en nuestro caso sera la posicion 6
que representa a n=5
```

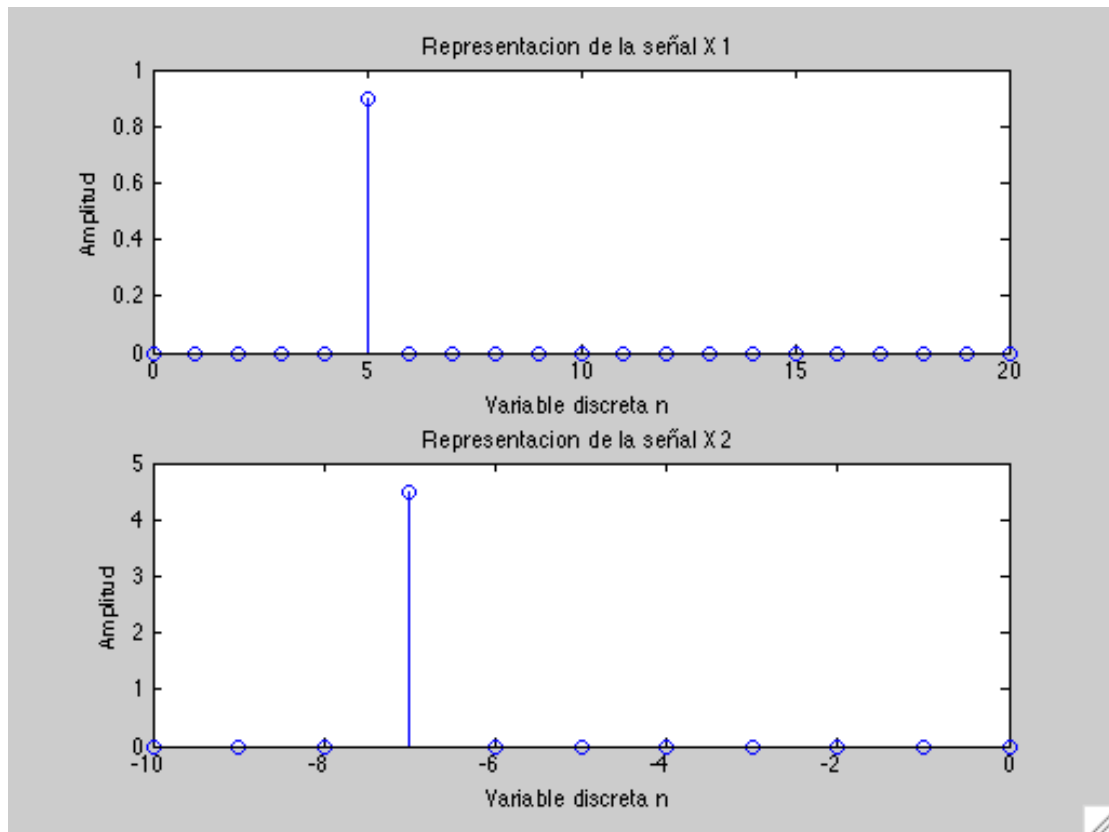
*%Implementacion Señal X2*

```
L=11;%Longitud
nn2=-10:0;%Vector de tiempos
imp2=zeros(L,1);%Matriz de cero con L filas y 1 columna
imp2(4)=4.5;%Localizamos el punto donde queremos el impulso, en nuestro caso sera la posicion 6
que representa a n=5
```

*%Representacion grafica*

```
subplot(2,1,1)
stem(nn1,imp1)
title('Representacion de la señal X1')
xlabel('Variable discreta n');
ylabel('Amplitud');
```

```
subplot(2,1,2)
stem(nn2,imp2)
title('Representacion de la señal X2')
xlabel('Variable discreta n');
ylabel('Amplitud');
```



- Generar y representar un tren de impulsos de amplitud 2, periodo 5 y longitud n.

```

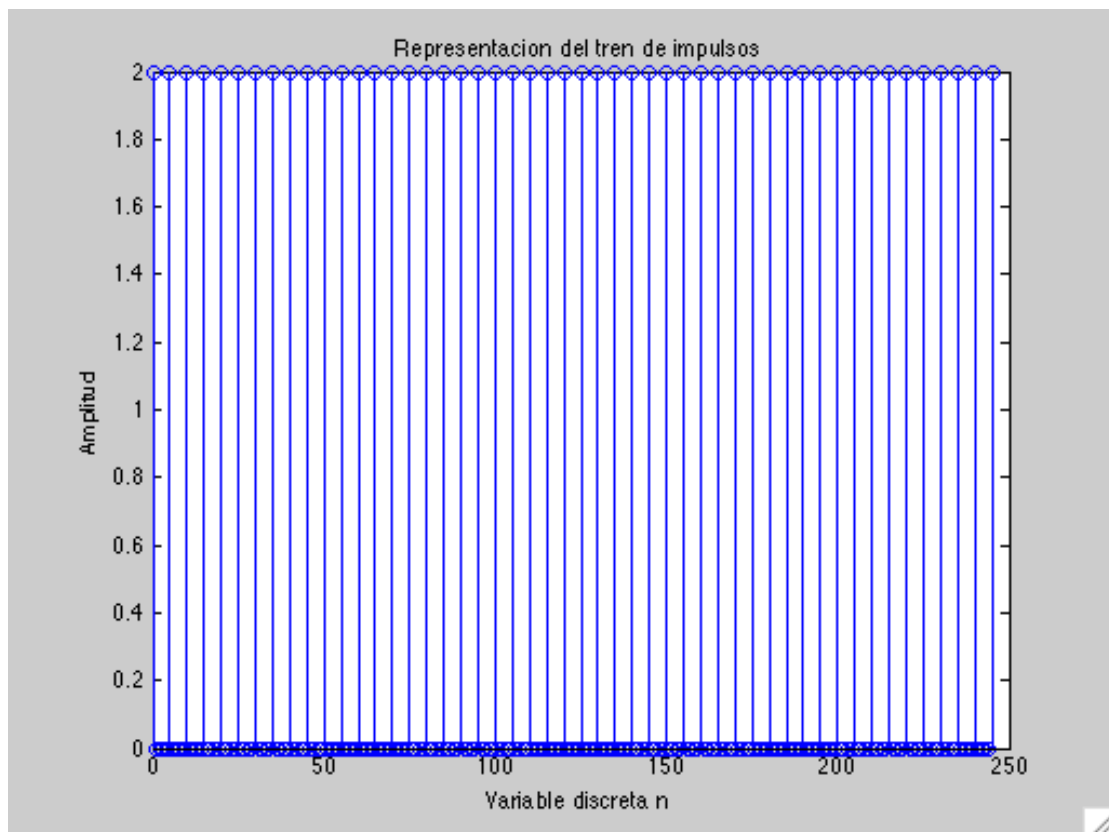
M=50;%Longitud
A=2;%Amplitud
P=5;%Periodo
nn=(0:1:((M-1)*P));%Vector de tiempos, de 0 a M-1
imp=zeros(M,1);%Matriz de Ceros de M filas y una columna

%Bucle donde se genera un tren de pulsos de amplitud A(2), en intervalos de
%a 5, hasta M-1(Periodo), porque empezamos en 0 se le resta 1

for l=1:5:(M*P)
    imp(l)=A;
end

```

Si las representamos obtenemos:



### **3. Generación de señales sinusoidales**

- Generar y representar graficamente las siguientes secuencias:

$$X_1[n] = \sin\left(\frac{\pi}{17}n\right) \quad 0 \leq n \leq 25.$$

$$X_2[n] = \sin\left(3\pi n + \frac{\pi}{2}\right) \quad -10 \leq n \leq 10.$$

$$X_3[n] = \cos\left(\frac{\pi}{\sqrt{23}}n\right) \quad 0 \leq n \leq 50.$$

Utilizaremos el siguiente código para definir cada una de las señales:

*%Creamos los vectores de tiempos:*

*nn1=[0:1:25];*

*nn2=[-10:1:10];*

*nn3=[0:1:50];*

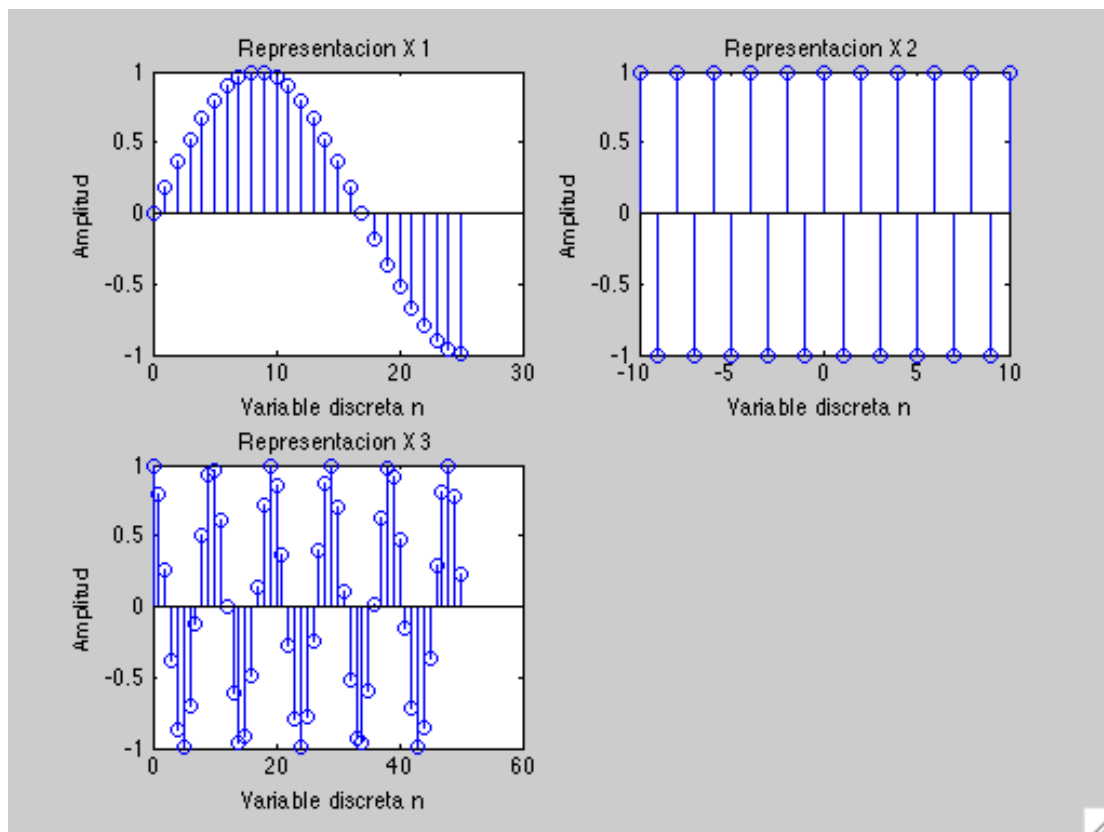
*%Creamos las señales*

*x1=sin((pi/17)\*nn1);*

*x2=sin((pi/2)+3\*pi\*nn2);*

*x3=cos((pi/sqrt(23))\*nn3);*

Representamos las señales:



#### **4. Funciones exponenciales y exponenciales complejas**

- Generar exponencial decreciente y sumar sus valores.

Creamos en una m-file una función con el siguiente código:

```
function y = genexp(b,n0,L)

%Usa:Y=genexp(B,N0,L)
%B-> Entrada escalar
%N0-> Instante de comienzo (entero)
%L-> Longitud de la señal generada (entero)
%Y-> Señal de salida Y(1:L)

if(L<=0)
    error('GENEXP:LONGITUD NO POSITIVA')
end
nn=n0+[1:L];
y=b.^(-abs(nn))% Hemos modificado la funcion original elevandola a la potencia negativa que es similar
a realizar el inverso
end
```

Ejecutamos las siguiente instrucción en MATLAB para generar nuestra exponencial:

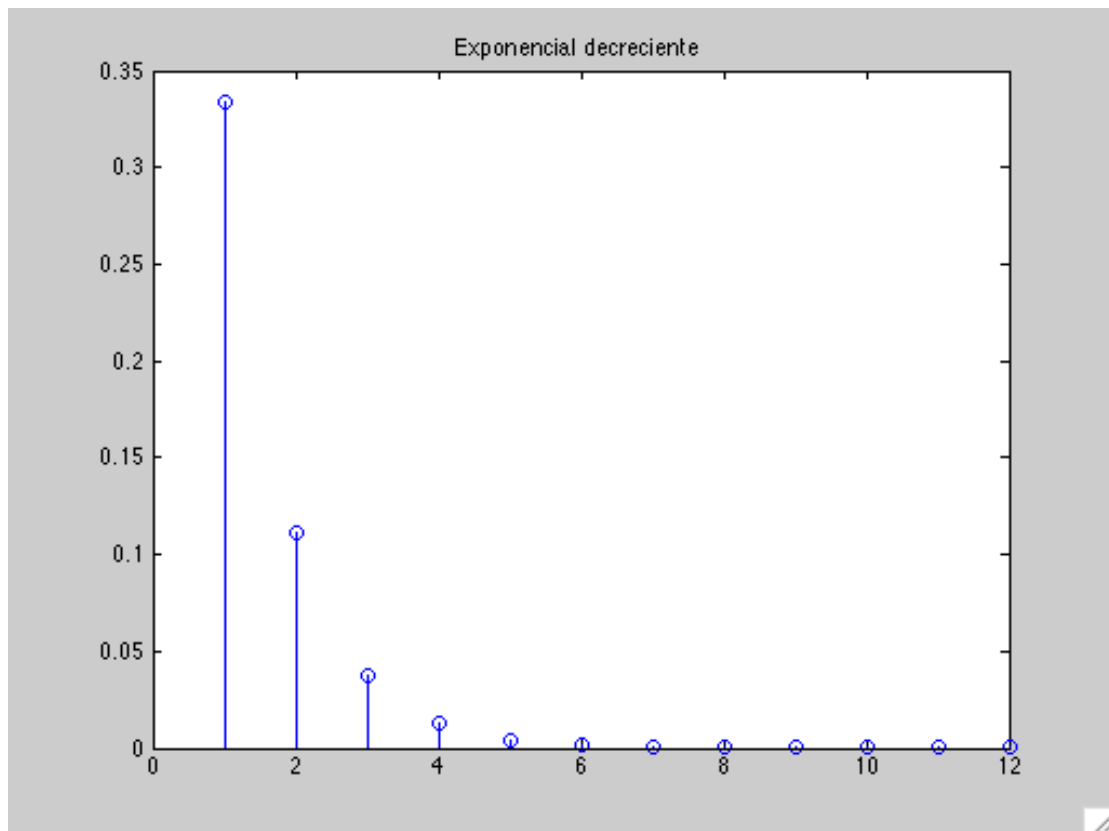
```
y=genexp(3,0,12)
```

Una vez generada, sumamos todos sus valores y creamos nuestro vector de tiempos:

```
suma=sum(y)%Sumamos los valores  
n=1:12
```

El resultado de la suma es 0.5

Si la representamos obtenemos:



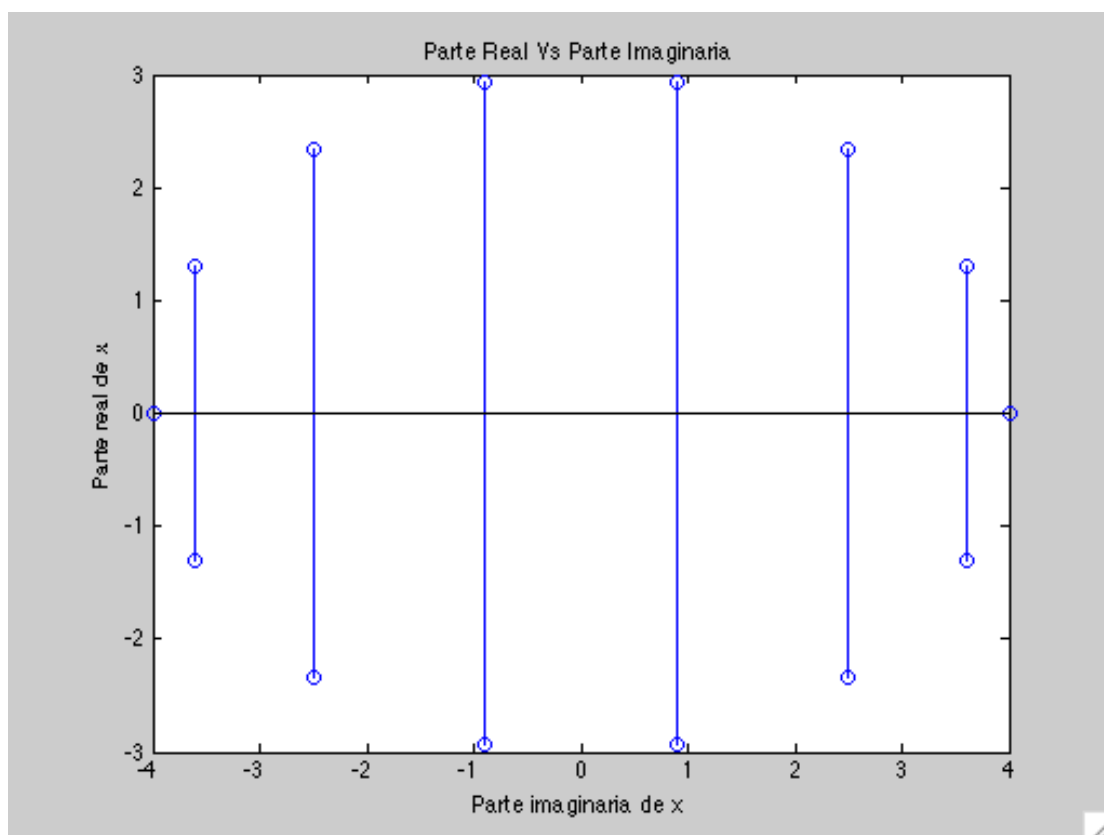
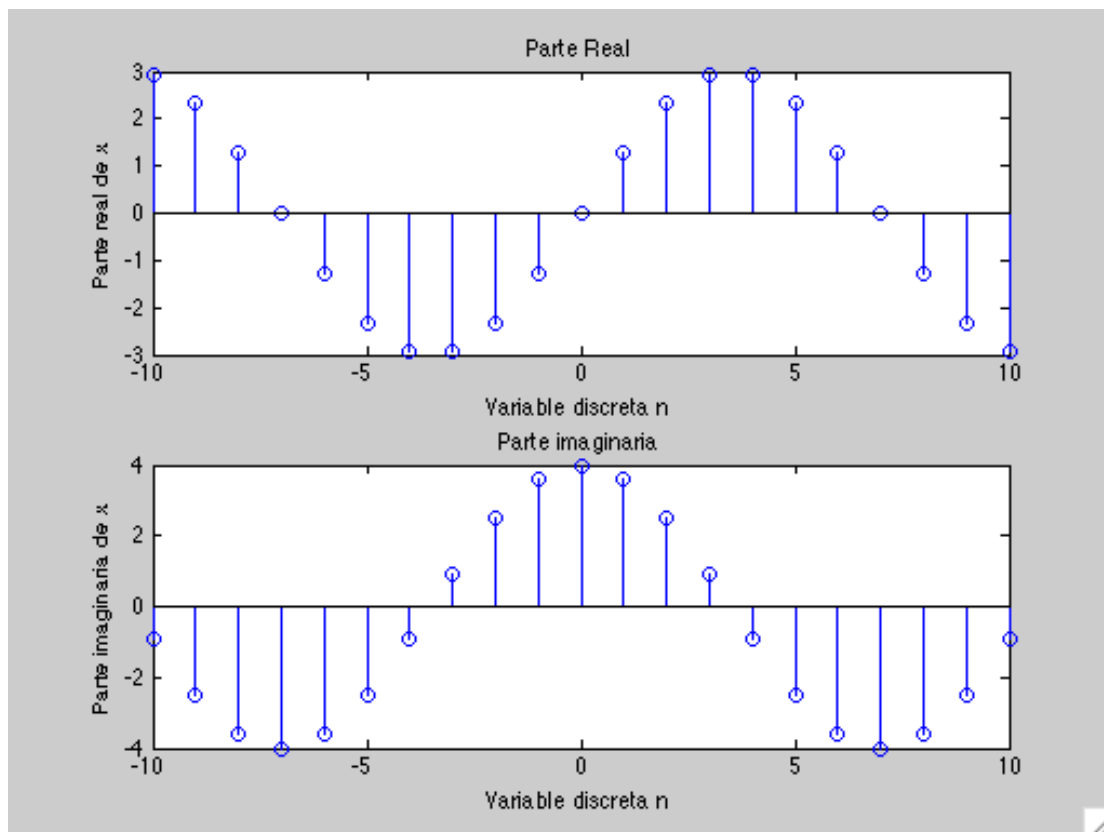
- Representar la parte real e imaginaria y la parte real en función de la imaginaria de la siguiente función:

$$X[n] = 3\sin\left(\frac{\pi}{7}n\right) + j4\cos\left(\frac{\pi}{7}n\right)$$

Generamos nuestra señal y el vector de tiempos:

```
nn=-10:1:10;%Vector de tiempos  
x=3*sin((pi/7)*nn)+j*4*cos((pi/7)*nn);%Implementacion de la funcion
```

Las representaciones gráficas obtenidas son las siguientes:



## **5. Teorema de muestreo**

- Escribir un programa que obtenga muestras de  $s(t)$  para crear una función de tiempo finita. Representar la señal para 3 frecuencias de muestreo distintas.
- Representar el módulo del espectro de las 3 señales.

Para realizar ambos apartados he creado una función en una m-file que contiene el siguiente código y que tendré que ejecutar 3 veces (una para cada frecuencia de muestreo):

```
function y = muestreo2(tinicial,tfinal,fmuestreo,a,f0,teta)
%Vector de tiempos del muestreo
nn=[tinicial:1/fmuestreo:tfinal];

%funcion en tiempo continuo
senal=a*cos((2*pi*f0*nn)+teta);

%Representamos dichos valores
subplot(2,1,1)
stem(nn,senal)
title('Señal muestreada')

%Definimos el numero de puntos que queremos usar para el calculo de la
%transformada
nnfft=256;

%Hallamos la transformada de Fourier.
modx=abs(fft(senal, nnfft));

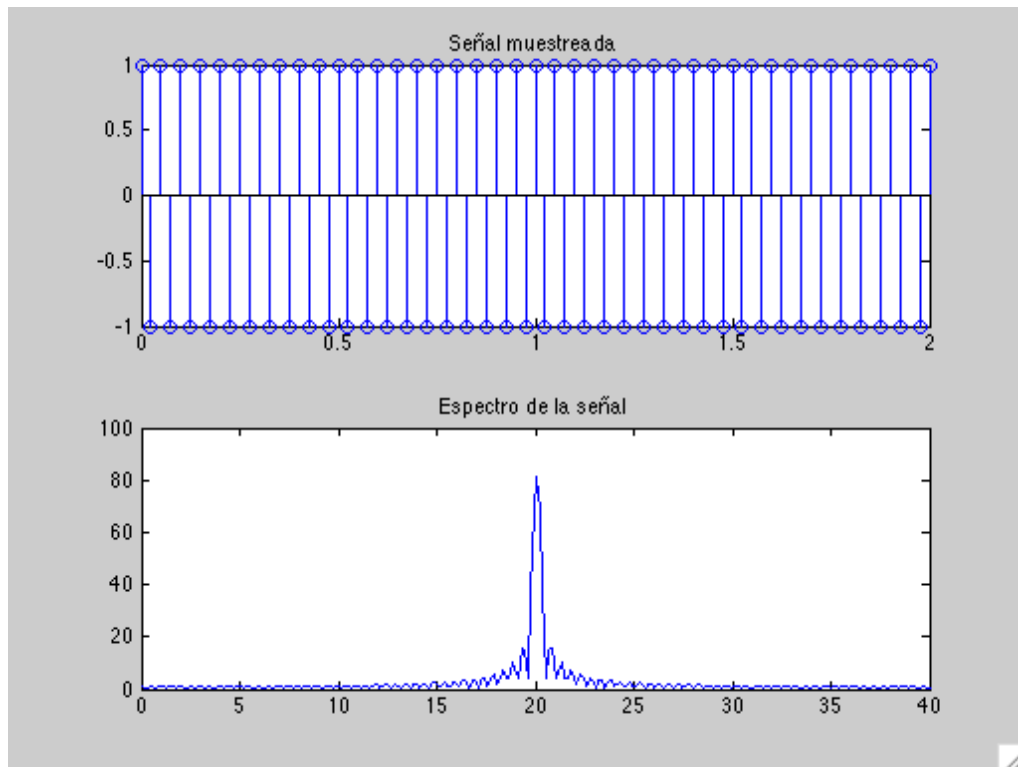
%Hallamos el vector de tiempos de la frecuencia
frec=0:fmuestreo/(nnfft-1):fmuestreo;

%Representamos el espectro
subplot(2,1,2)
plot(frec,modx)
title('Espectro de la señal')
```

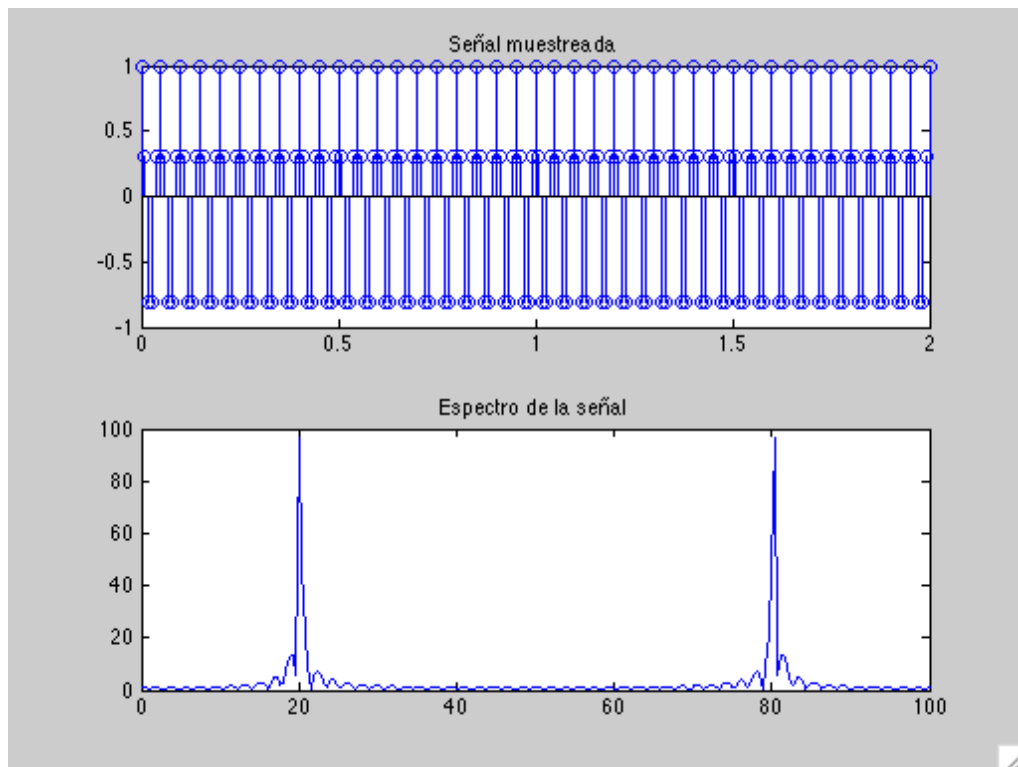


Estos son los resultados obtenidos:

- Frecuencia de muestreo de 40 Hz:



- Frecuencia de muestreo de 100 Hz:



- Frecuencia de muestreo de 30 Hz:

