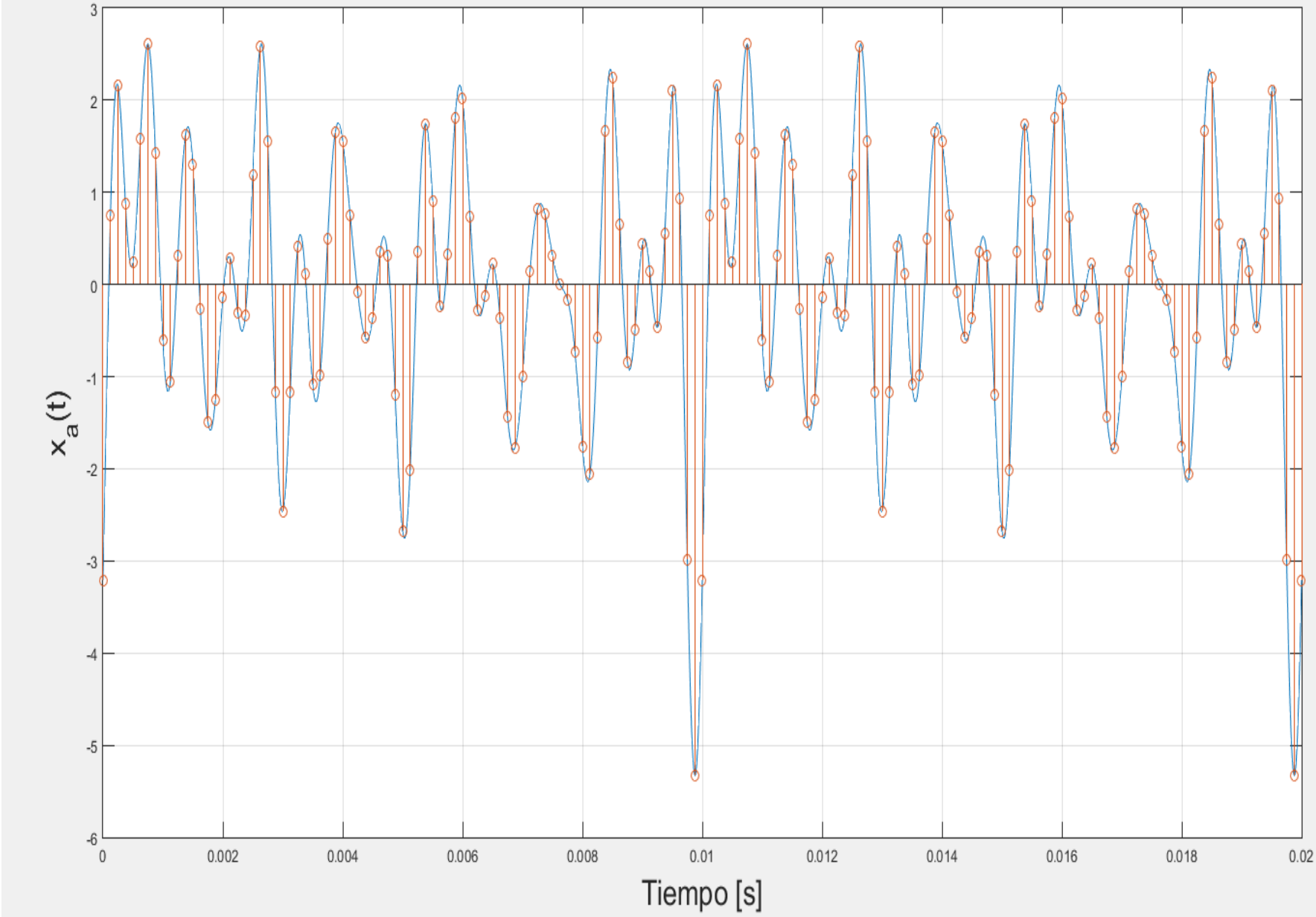
Nombre: Eguiarte Morett Luis Andrés.

Titulo Tarea: Cuantización

Tarea No. 7

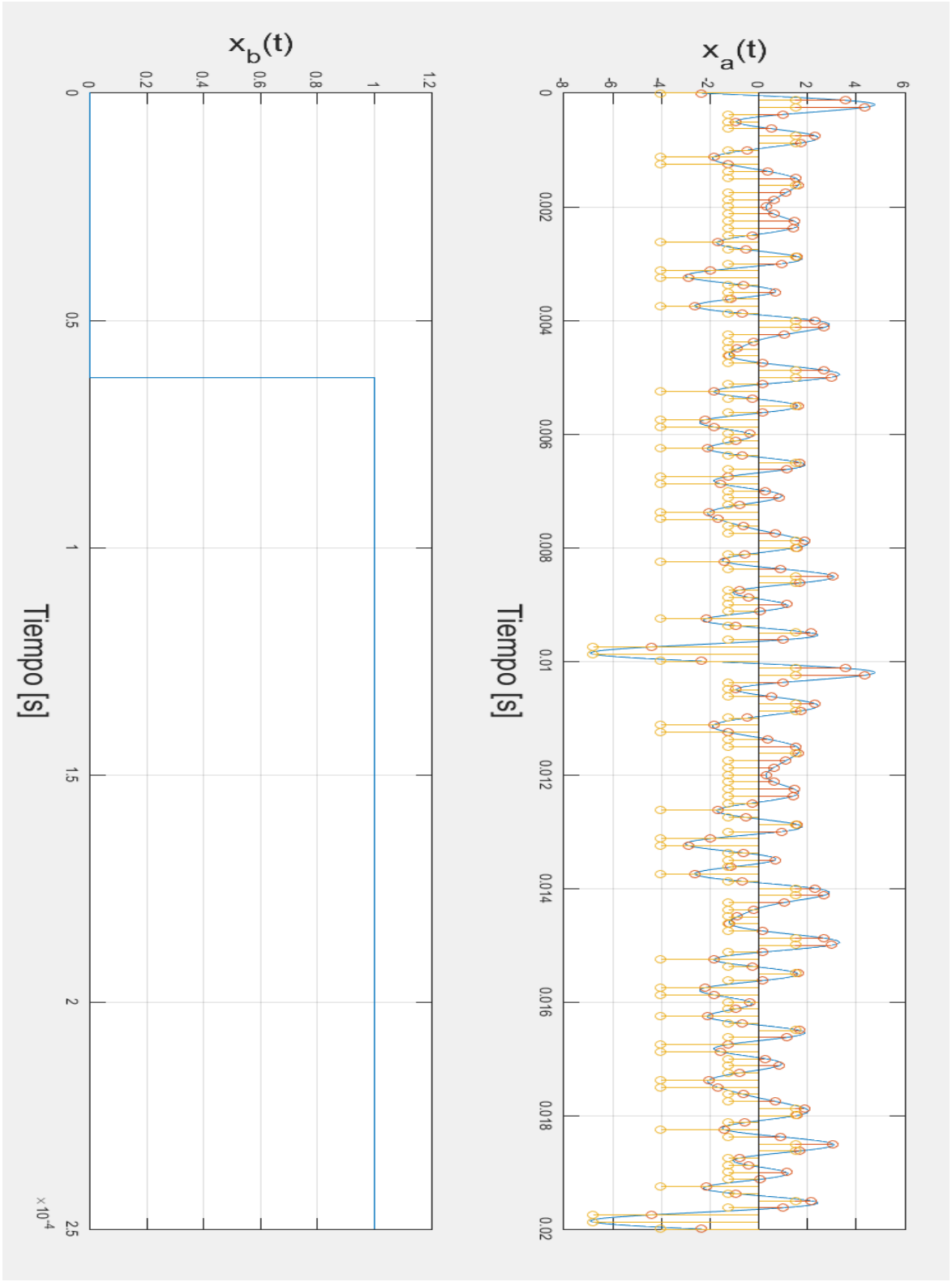
1. Señal pseudo-analógica en el tiempo
2. Señal muestreada en el tiempo


3
2
1
0
-1
-2
-3
-4
-5
-6
$x_a(t)$
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
Tiempo [s]

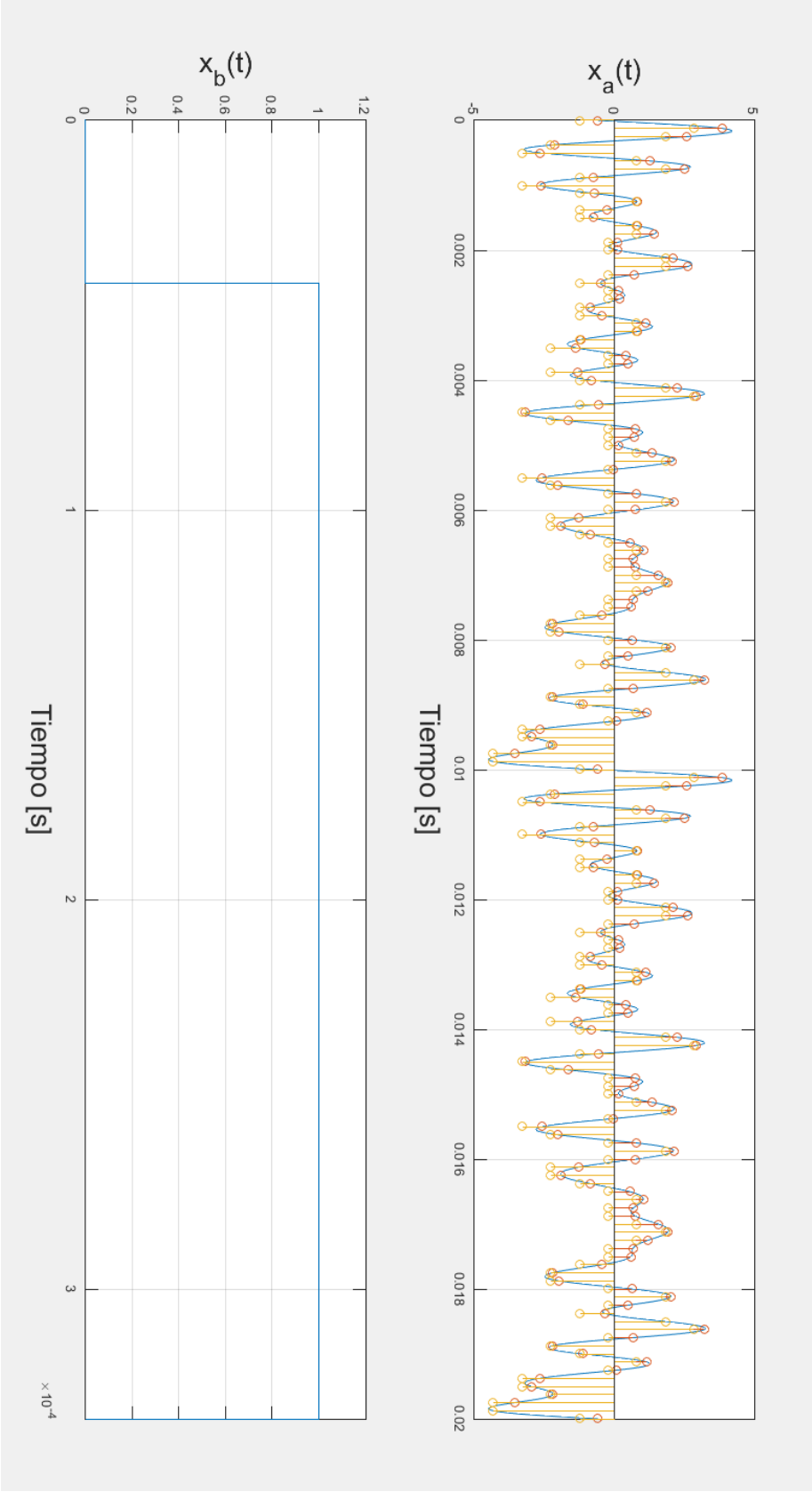
3) Señal cuantizada

4) Señal binaria en el tiempo

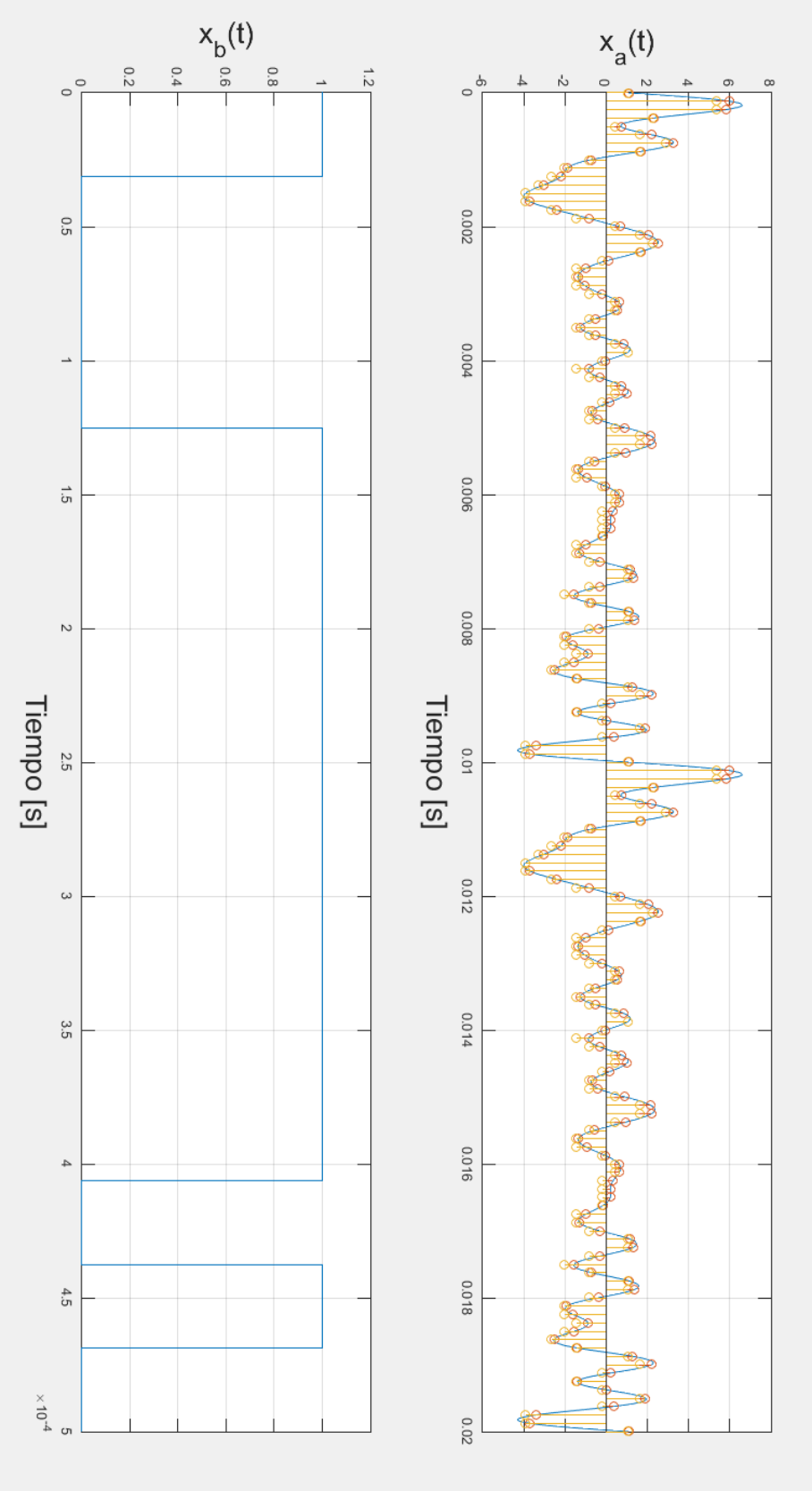
N = 4; Duración de bit=$6.25*10^{-5}$



N=8; Duración de bit= $4.166*10^{-5}$



N=16; duración de bit= $3.125*10^{-5}$



```
close all;
clear all;
clc;

Amp=1; N=20;                          % Parámetros de la señal x_a(t), N =
Número de coeficientes de la Serie de Fourier
FLAG_PLOT=1;                          % Indica si graficar (1) o no (0)

f0=100;                               % Frecuencia fundamental de la señal
x_a(t)
T0=1/f0;                              % Período fundamental de la señal x_a(t)
Res=32;                               % Muestras por cada período fundamental
de la señal pseudo-analógica, les había dicho 8, pueden dejarla como 32
fs_mat=Res*N*f0;                      % Frecuencia de muestreo para generar la
señal pseudo-analógica (NO confundir con frecuencia de muestreo para
discretizarla)
Ts_mat=1/fs_mat;                      % Intervalos de muestreo para generar la
señal pseudo-analógica
t=0:Ts_mat:2*T0;                      % Vector de tiempo, de 0 a 2 períodos
(para graficar)

%Cálculo de los coeficientes de la serie de Fourier y reconstrucción de la
señal x(t)
a0=0;                                 % Componente de DC
x_a=a0;
A=rand(1,N);
Theta=pi*rand(1,N);
for n=1:1:N
    x_temp=A(n)*cos(2*pi*f0*n*t-Theta(n));
    x_a=x_a+x_temp;
end

% Discretizar la señal pseudo-analógica

S=4;
fs=S*N*f0;      % Frecuencia de muestreo = S veces la frecuencia máxima
(N*f0) de la señal pseudo-analógica
Ts=1/fs;        % Período de muestreo [s]

% x_dis(1)=x_a(1+0*Res/S); % x_dis es la señal pseudo-analógica muestreada.
Sólo obtengo como ejemplo las primeras 4 muestras. Generalicen para muestrear
toda la señal
% x_dis(2)=x_a(1+1*Res/S);
% x_dis(3)=x_a(1+2*Res/S);
% x_dis(4)=x_a(1+3*Res/S);
%etc...

t_dis=0:Ts:2*T0;
% t_dis=t_dis(1:4); % t_dis(1:4) sirve para limitar el vector de tiempo para
sólo las primeras cuatro muestras que les puse de ejemplo. Generalizar para
todo el tiempo que dura x_a(t)

for i=1 : 1 : size(t_dis, 2)
   x_dis(i)=x_a(1+(i-1)*Res/S);
end
```

```
% Cuantizar la señal muestreada (esta es su tarea)
x_dis_max = max(x_dis);
x_dis_min =  min(x_dis);
af=x_dis_max-x_dis_min;
N2=16;
lvl=af/N2;
aux=0;
n_bits=log2(N2);
for i = 1 : 1 : size(t_dis,2)
    for n = 0 : 1 : N2
        if x_dis(i)>= x_dis_min+lvl*n
            if n==N2
                x_cuant(i)=x_dis_min+lvl*(n-1);
                aux=n-1;
                break;
            else
                x_cuant(i) = x_dis_min+lvl*n;
            end
        else
            aux=n-1;
            break;
        end
    end
    aux=de2bi(aux, 'left-msb');
    while length(aux)<n_bits
        aux=[0,aux];
    end
    if i==1
        x_bits=aux;
    else
        x_bits=[x_bits, aux];
    end
end
% Codificar la señal cuantizada (esta es su tarea)
t_bits=0:Ts/n_bits:(length(x_bits)-1)*Ts/n_bits;

disp(x_bits);
disp(t_bits);
% Graficas
if FLAG_PLOT == 1
    figure
    subplot(2,1,1);
    plot(t,x_a);                % Gráfica de la señal x_a(t)
    grid on; xlabel('Tiempo [s]','fontsize',20);
ylabel('x_a(t)','fontsize',20);
    hold on                     % Sirve para graficar dos curvas en la misma
gráfica. Las señales cuantizadas de tarea las quiero así traslapadas con la
señal original
    stem(t_dis,x_dis);% Gráfica de la señal muestread
    stem(t_dis, x_cuant);
    subplot(2,1,2);
    stairs(t_bits,x_bits);
    grid on; xlabel('Tiempo [s]','fontsize',20);
ylabel('x_b(t)','fontsize',20);
    axis([0 n_bits*Ts 0 1.2]);
```

```
    hold off
end
```