

Actividad 1:

Tema: “Muestreo y Cuantificación en Octave”.

Objetivos:

- El alumno entienda el muestreo en Octave
- El alumno entienda la cuantificación en Octave

Nota: El nombre del archivo a enviar debe tener el siguiente formato:

PrimerNombre_ApellidoPaterno_GX.pdf donde X=numero de su grupo

Cada alumno enviará su actividad grupal e indicara que alumno(s) de su grupo no ayudo o colaboro en realización de dicha actividad.

Alumno(s) que no ayudo: _____

Pasos a seguir:

Observar el siguiente video para el entendimiento de muestreo y cuantificación:


Video1:

<https://www.youtube.com/watch?v=9GxcNyGQsuk>

Investigar sobre las funciones de Octave en grupo

Funciones en Octave

Funcion Triplus() –Señal Triangular


Octave Forge

[Home](#)
[Packages](#)
[Developers](#)
[Support/Help](#)
[Documentation](#)

Function File: $y = \text{tripuls}(t)$

Function File: $y = \text{tripuls}(t, w)$

Function File: $y = \text{tripuls}(t, w, \text{skew})$

Generate a triangular pulse over the interval $[-w/2, w/2]$, sampled at times t . This is useful with the function `pulstran` for generating a series of pulses.

`skew` is a value between -1 and 1, indicating the relative placement of the peak within the width. -1 indicates that the peak should be at $-w/2$, and 1 indicates that the peak should be at $w/2$. The default value is 0.

Example:

```
fs = 11025; # arbitrary sample rate
f0 = 100; # pulse train sample rate
w = 0.3/f0; # pulse width 3/10th the distance between pulses
plot (pulstran (0:1/fs:4/f0, 0:1/f0:4/f0, "tripuls", w));
```

See also: `gauspuls`, `pulstran`, `rectpuls`.

Demonstration 1


The following code

```
fs = 11025; # arbitrary sample rate
f0 = 100; # pulse train sample rate
w = 0.5/f0; # pulse width 1/10th the distance between pulses
x = pulstran (0:1/fs:4/f0, 0:1/f0:4/f0, "tripuls", w);
plot ([0:length(x)-1]*1000/fs, x);
xlabel ("Time (ms)");
ylabel ("Amplitude");
title ("Triangular pulse train of 5 ms pulses at 10 ms intervals");
```

Produces the following figure

Referencia: <https://octave.sourceforge.io/signal/function/tripuls.html>

Funcion Pulstran()


Octave Forge

[Home](#)
[Packages](#)
[Developers](#)
[Support/Help](#)
[Documentation](#)

Function File: $y = \text{pulstran}(t, d, \text{func}, \dots)$

Function File: $y = \text{pulstran}(t, d, p)$

Function File: $y = \text{pulstran}(t, d, p, Fs)$

Function File: $y = \text{pulstran}(t, d, p, Fs, \text{method})$

Generate the signal $y = \sum(\text{func}(t+d, \dots))$ for each d . If d is a matrix of two columns, the first column is the delay d and the second column is the amplitude a , and $y = \sum(a * \text{func}(t+d))$ for each d, a . Clearly, `func` must be a function which accepts a vector of times. Any extra arguments needed for the function must be tagged on the end.

Example:

```
fs = 11025; # arbitrary sample rate
f0 = 100; # pulse train sample rate
w = 0.001; # pulse width of 1 millisecond
subplot (pulstran (0:1/fs:0.1, 0:1/f0:0.1, "rectpuls", w), fs);
```

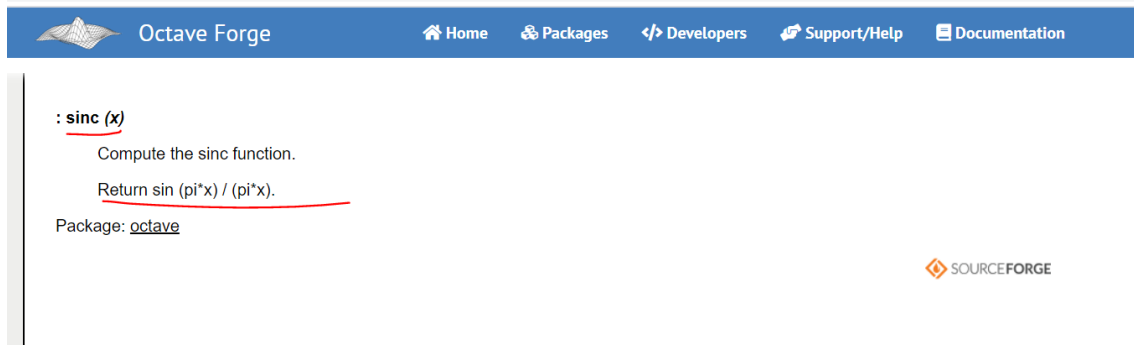
If instead of a function name you supply a pulse shape sampled at frequency F_s (default 1 Hz), an interpolated version of the pulse is added at each delay d . The interpolation stays within the time range of the delayed pulse. The interpolation method defaults to linear, but it can be any interpolation method accepted by the function `interp1`.

Example:

```
fs = 11025; # arbitrary sample rate
f0 = 100; # pulse train sample rate
w = boxcar(10); # pulse width of 1 millisecond at 10 kHz
subplot (pulstran (0:1/fs:0.1, 0:1/f0:0.1, w, 10000), fs);
```

Referencia: <https://octave.sourceforge.io/signal/function/pulstran.html>

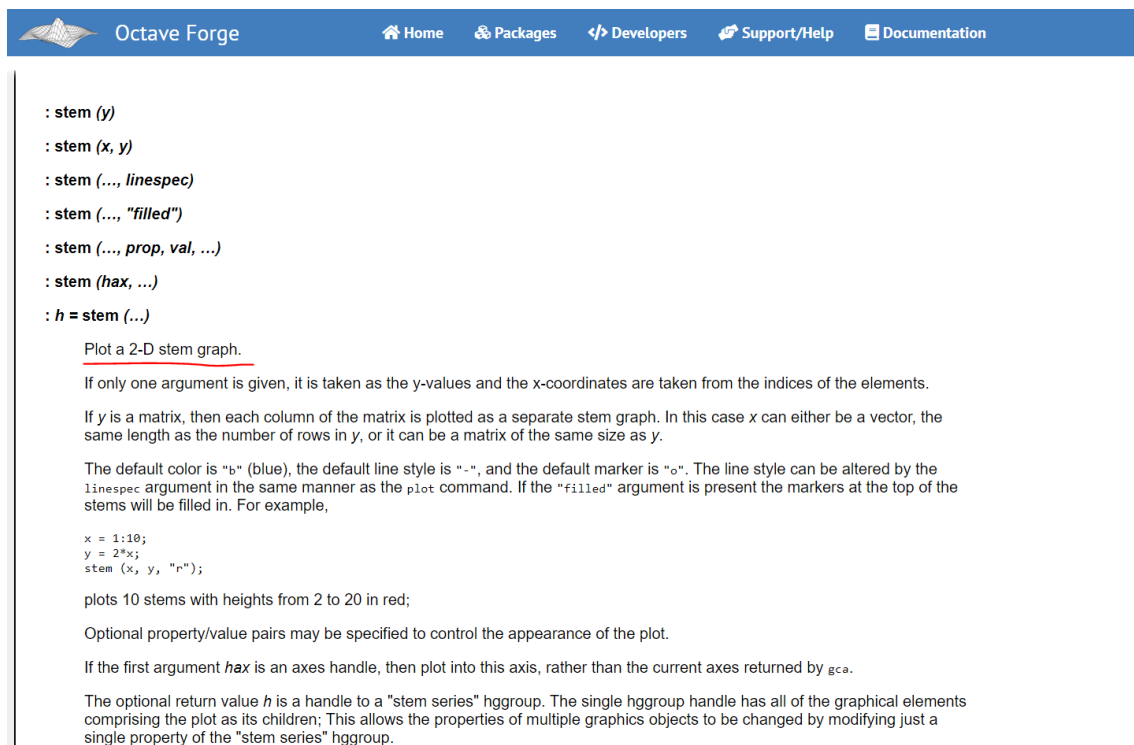
Funcion Sampling (Sinc)



The screenshot shows the Octave Forge website header with navigation links: Home, Packages, Developers, Support/Help, and Documentation. The main content area displays the documentation for the `sinc` function. It includes the function signature `: sinc (x)`, a description "Compute the sinc function.", the return value "Return $\sin(\pi x) / (\pi x)$ ", and the package name "Package: `octave`". The SourceForge logo is visible in the bottom right corner.

Referencia: <https://octave.sourceforge.io/octave/function/sinc.html>

Funcion Stem() –Plot Discrete Sequence



The screenshot shows the Octave Forge website header with navigation links: Home, Packages, Developers, Support/Help, and Documentation. The main content area displays the documentation for the `stem` function. It lists several function signatures: `: stem (y)`, `: stem (x, y)`, `: stem (... , linespec)`, `: stem (... , "filled")`, `: stem (... , prop, val, ...)`, `: stem (hax, ...)`, and `: h = stem (...)`. The description states "Plot a 2-D stem graph." and explains that if only one argument is given, it is taken as the y-values and the x-coordinates are taken from the indices of the elements. It also mentions that if `y` is a matrix, each column is plotted as a separate stem graph. The default color is "b" (blue), the default line style is "--", and the default marker is "o". The `linespec` argument can alter the line style and marker. If the "filled" argument is present, the markers at the top of the stems will be filled. An example code snippet is provided:

```
x = 1:10;  
y = 2*x;  
stem (x, y, "r");
```

 This plots 10 stems with heights from 2 to 20 in red. Optional property/value pairs can be specified to control the appearance of the plot. If the first argument `hax` is an axes handle, the plot is drawn into this axis. The optional return value `h` is a handle to a "stem series" hgroup, which contains all graphical elements of the plot.

Referencia: <https://octave.sourceforge.io/octave/function/stem.html>

Preguntas:

1-Defina que es el Muestreo y la Cuantificación. En que aplicaciones se usan

2-Defina que es el teorema de Nyquist y jitter

3- Dada una señal sinusoidal $y(t)$ de frecuencia 2Hz , de amplitud de G Voltios y de frecuencia de Muestreo $f_s=40\text{Hz}$. El intervalo del tiempo es desde 0 a 2.

G=Numero de su Grupo

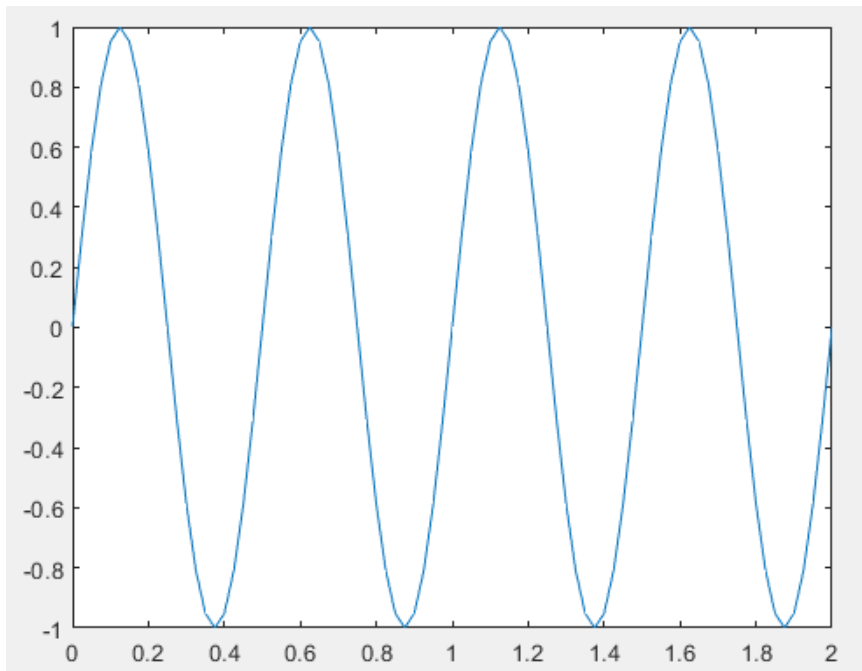
Graficar la señal $y(t)$ continuo en el tiempo

Comandos

```
%% SEÑAL SENOIDAL
clc, clear, close all;
fs = 40           % Frecuencia de muestreo
t = 0:1/(fs):2;   % Intervalo de t
f = 2             % frecuencia en Hz
G = 1;           % Amplitud
y = G*sin(2*pi*f*t) % Señal

axis([0 2 -G G])  % Ejes
plot(t,y)         % Grafica de la señal continua
```

Pantallazo de grafica



Graficar la cuantificacion de la señal $y(t)$ discreta en el tiempo

Comandos

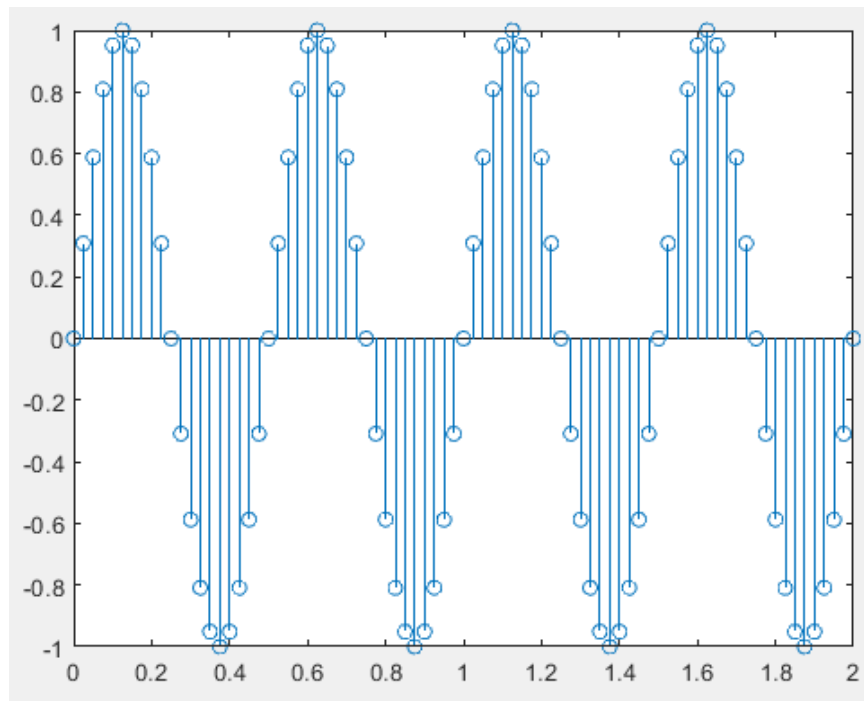
```
%% CUANTIZACION IDEAL (Ancho de pulso = 0)
clc, clear, close all;
fs = 40; % Frecuencia de muestreo
t = [0:1/(fs):2]; % Intervalo de t
f = 2; % frecuencia de la señal
G = 1; % Amplitud
axis([0 2 -G G]) % Ejes
y = G*sin(2*pi*f*t) % Señal en funcion del tiempo
stem(t,y) % Señal cuantizada con ancho de pulso 0

%% CUANTIZACION REAL (Ancho de pulso distinto a 0)
clc, clear;
fs = 40; % Frecuencia de muestreo
t = [0:0.0001:2]; % Intervalo de t
d = [0:1/(fs):2]; % Cuantizacion basado en frec de muestreo
f = 2; % frecuencia de la señal
G = 1; % Amplitud
axis([0 2 -G G]) % Ejes

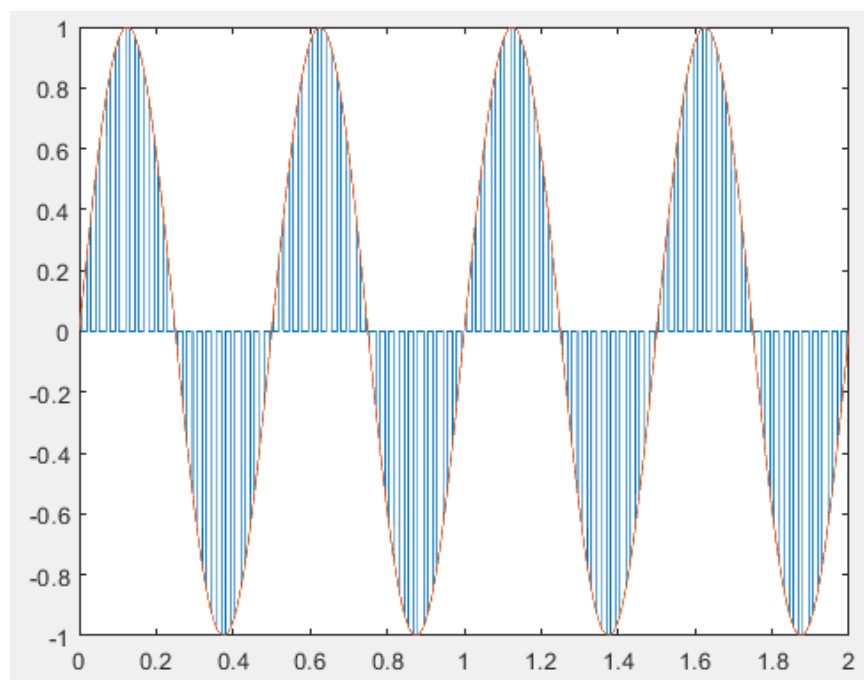
y = G*sin(2*pi*f*t) % Señal
h = pulstran(t,d,'rectpuls',0.01); % Función tren de pulsos
unitarios
sampling = h.*y; % Muestreo
plot(t,sampling,t,y) % Señal cuantizada con ancho de pulso ;0
```

Pantallazo de graficas

Cuantización ideal (Ancho de pulso igual a 0)



Cuantización real (Ancho de pulso distinto de 0, $f_s = 40$ Hz)



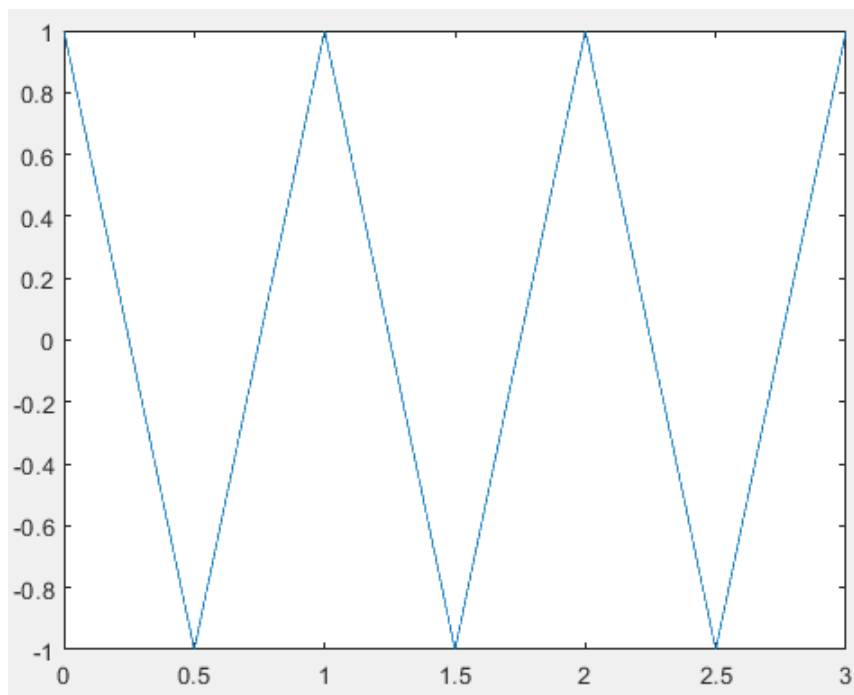
4-Dada la señal triangular $y(t)$ donde la frecuencia de Muestreo $f_s=20\text{Hz}$. La amplitud de esta señal es de G Voltios. $G=\text{Numero de su Grupo}$

Graficar la señal $y(t)$ continua

Comandos

```
%% SEÑAL TRIANGULAR CONTINUA
clear, clc, close all;
fs = 20;           % Frecuencia de muestreo
t = [0:0.001:3];   % Intervalo de t
d = [0:1:3];        % Cuantización basado en a frec de
muestreo
G = 1;             % Amplitud
y = 2*G*pulstran(t,d,'tripuls',1)-1; % Señal triangular
plot (t,y)         % Grafica de la señal continua
```

Pantallazo de grafica



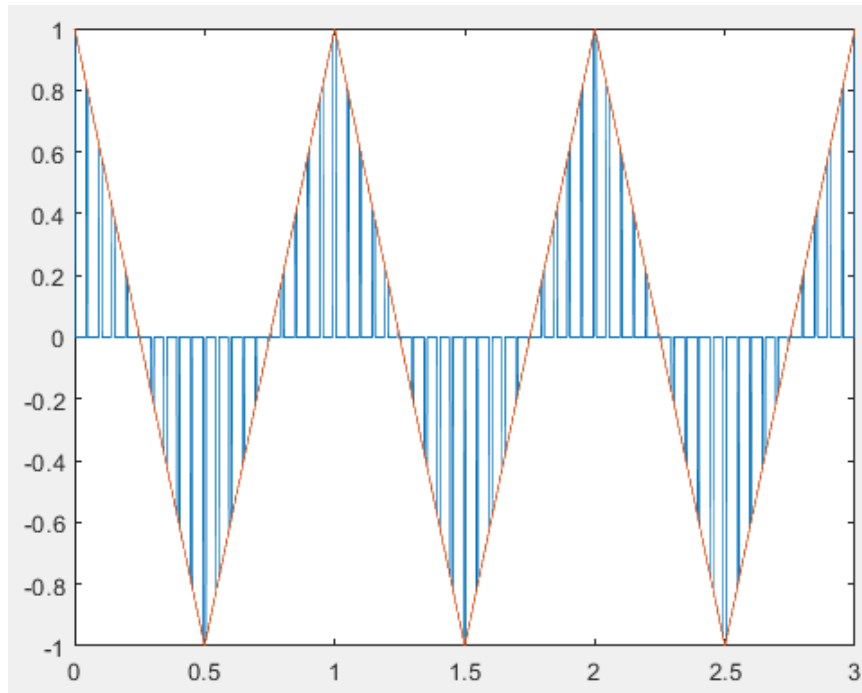
Graficar la cuantificación de la señal $y(t)$ discreta en el tiempo

Comandos

```
%% SEÑAL TRIANGULAR DISCRETA
clear, clc, close all;
fs = 20;           % Frecuencia de muestreo
t = [0:0.001:3];   % Intervalo de t
d = [0:1:3];        % frec de señal
G = 1;             % Amplitud
y = 2*G*pulstran(t,d,'tripuls',1)-1;
```

```
e = [0:1/(fs):3]; % Cuantizacion basado en frec de
muestreo
h = pulstran(t,e,'rectpuls',0.01); % Tren de pulsos unitarios
sampling = h.*y; % Muestreo
plot(t,sampling,t,y) % Grafica de Señal cuantizada
```

Pantallazo de graficas (fs = 20 Hz)



5- Graficar la señal periódica digital $y(t)$ donde la frecuencia de Muestreo $f_s=20\text{Hz}$

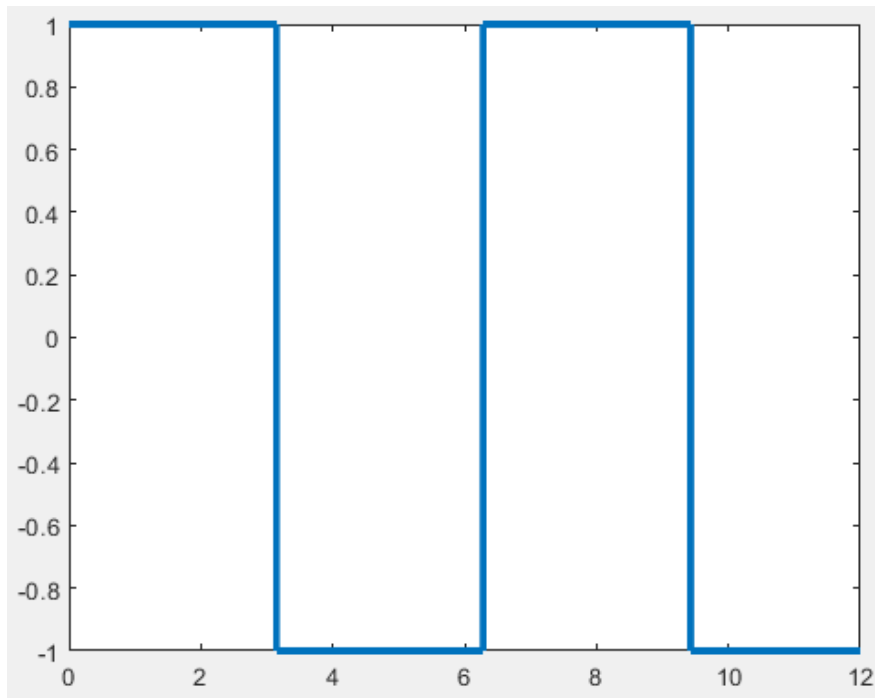
Graficar la señal $y(t)$

Comandos

```
%% SEÑAL DIGITAL
clear, clc, close all;
fs = 20; % Frecuencia de muestreo
t = [0:0.001:12]; % Intervalo de t
G = 1; % Amplitud

y = G*square(t); % Señal digital
plot(t,y,'LineWidth',3) % Grafica de señal continua
```


Pantallazo de graficas



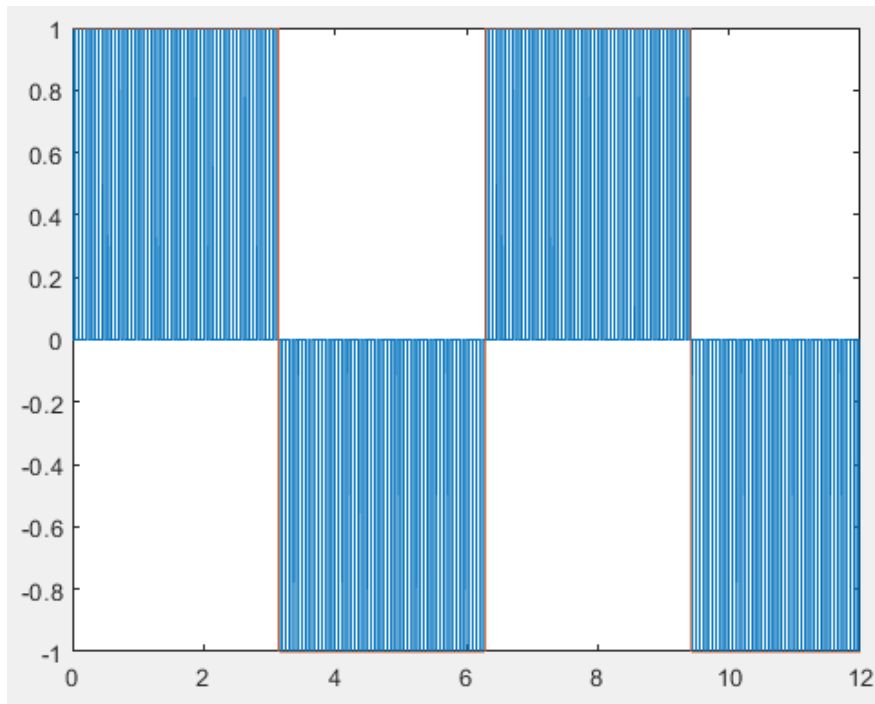
Graficar la cuantificación de la señal $y(t)$ discreta en el tiempo

Comandos

```
%% SEÑAL DIGITAL DISCRETA
clear, clc, close all;
fs = 20;           % Frecuencia de muestreo
t = [0:0.001:12];  % Intervalo de t
G = 1;             % Amplitud

y = G*square(t);   % Señal digital
e = [0:1/(fs):12]; % Cuantización basado en frec de muestreo
h = pulstran(t,e,'rectpuls',0.01); % Tren de pulsos unitarios
sampling = h.*y;    % Muestreo
plot(t,sampling,t,y) % Señal cuantizada
```

Pantallazo de grafica (fs = 20 Hz)



6-Se tiene la señal $y(t)$ que es la sumatoria de 2 señales sinusoidales que son las siguientes:

$X(t)=A*\sin(2*\pi*f*t);$ donde la $A=G$ y la $f=10\text{Hz}$ $G=\text{Numero de su Grupo}$

$Z(t)=A*\sin(2*\pi*f*t);$ donde la $A=G$ y la $f=20\text{Hz}$ $G=\text{Numero de su Grupo}$

Grafico de la señal $y(t)$ continua resultante

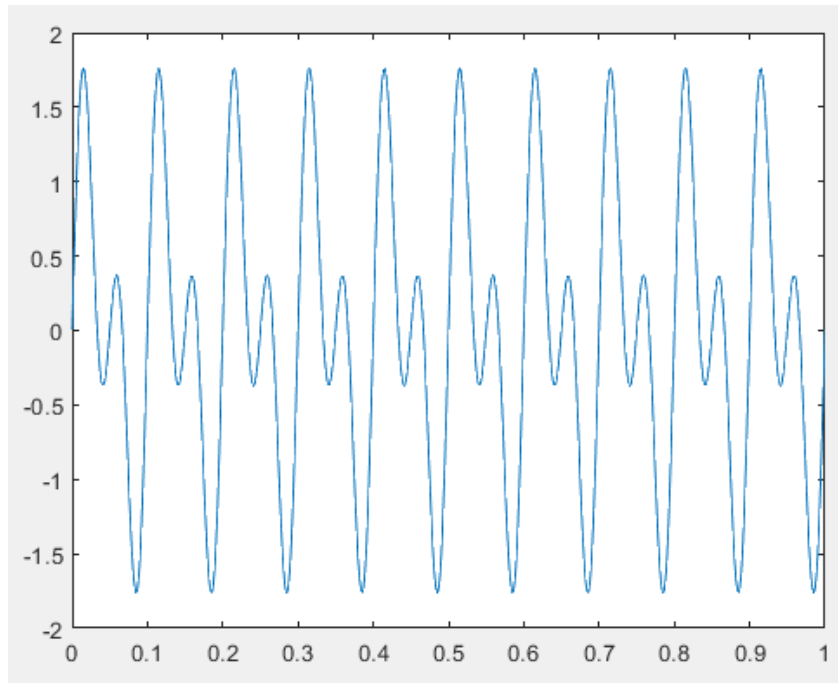
Comandos

```
% SUMA DE 2 SEÑALES SENOIDALES CON DISTINTA FRECUENCIA
clear, clc, close all;
t = [0:0.001:1];      % Intervalo de t
f1 = 10;               % Frecuencia de la señal 1
f2 = 20;               % Frecuencia de la señal 2
G = 1;                 % Amplitud de las señales

x = G*sin(2*pi*f1*t); % Señal 1
y = G*sin(2*pi*f2*t); % Señal 2
z = x+y;               % Resultante

plot(t,z)
```

Pantallazo de graficas



Graficar la cuantificación de la señal $y(t)$ discreta en el tiempo, con frecuencia de Muestreo $f_s=100\text{Hz}$.

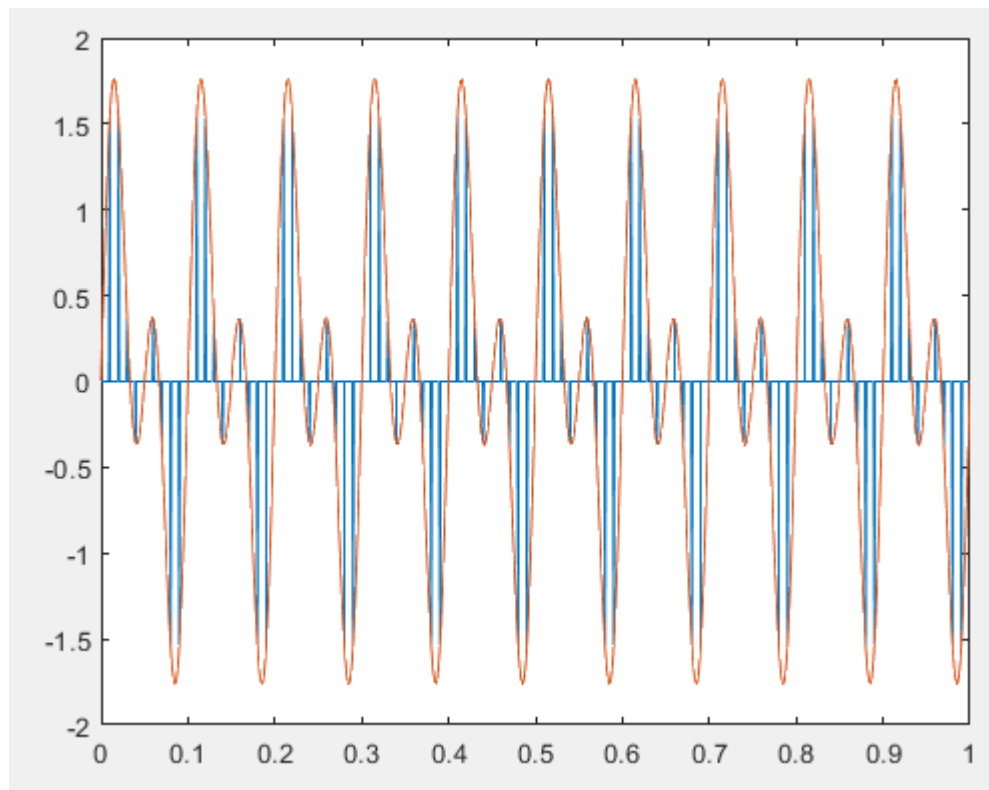
Comandos

```
% SUMA DE 2 SEÑALES SENOIDALES CON DISTINTA FRECUENCIA
% SEÑAL CUANTIZADA
clear, clc, close all;
fs = 100                % Frecuencia de muestreo
t = [0:0.001:1];       % Intervalo de t
f1 = 10;                % Frecuencia de la señal 1
f2 = 20;                % Frecuencia de la señal 2
G = 1;                 % Amplitud de las señales

x = G*sin(2*pi*f1*t); % Señal 1
y = G*sin(2*pi*f2*t); % Señal 2
z = x+y;               % Resultante

d = [0:1/(fs):1];      % Cuantización basado en frec de muestreo
h = pulstran(t,d,'rectpuls',0.001); % Tren de pulsos unitarios
sampling = h.*z;        % Muestreo
plot(t,sampling,t,z) % Señal cuantizada
```

Pantallazo de graficas (fs = 100 Hz)



**7-Graficar la señal sampling para un intervalo del tiempo de $-2 \cdot G$ a $2 \cdot G$ segundos donde:
 G =Numero de su Grupo**

De acuerdo con el documento entregado por el profesor, la Funcion Sampling en Octave esta representada por Sinc().

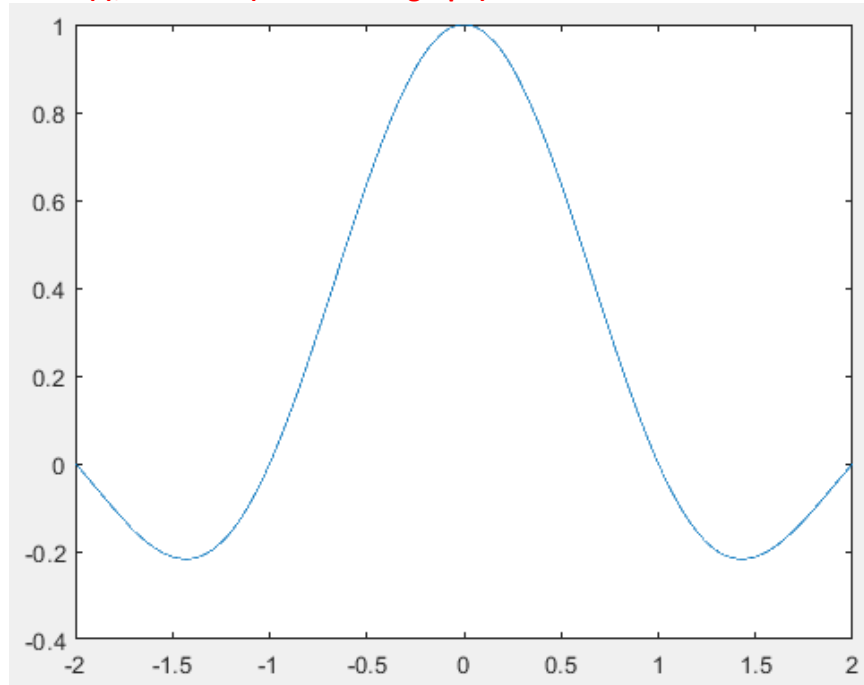
Comandos

```
% SEÑAL SAMPLING: Sinc()
G = 1;
t = [-2*G:0.001:2*G];      % Intervalo de t
y = sinc(t)                 % Señal sampling

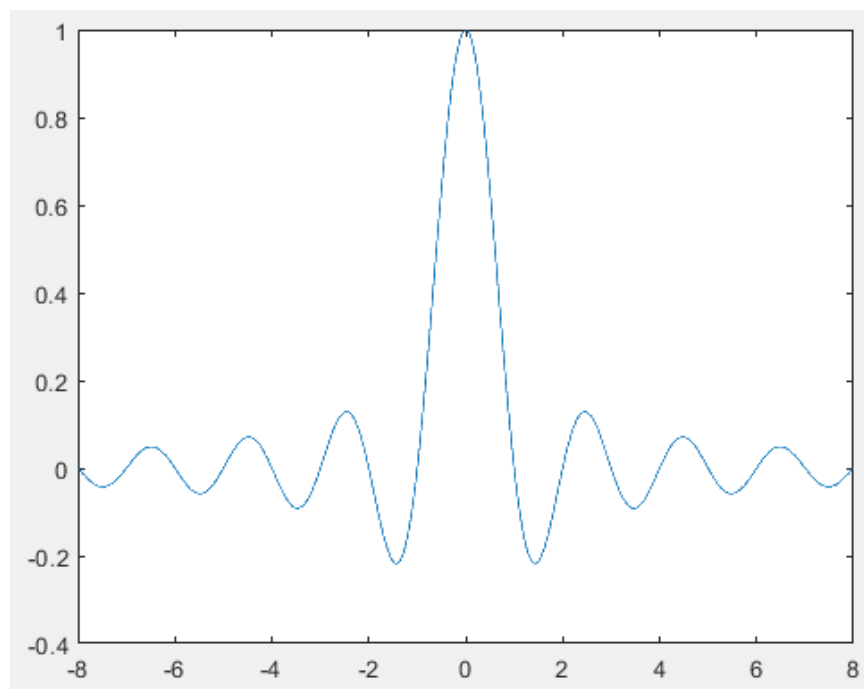
plot(t,y)
```

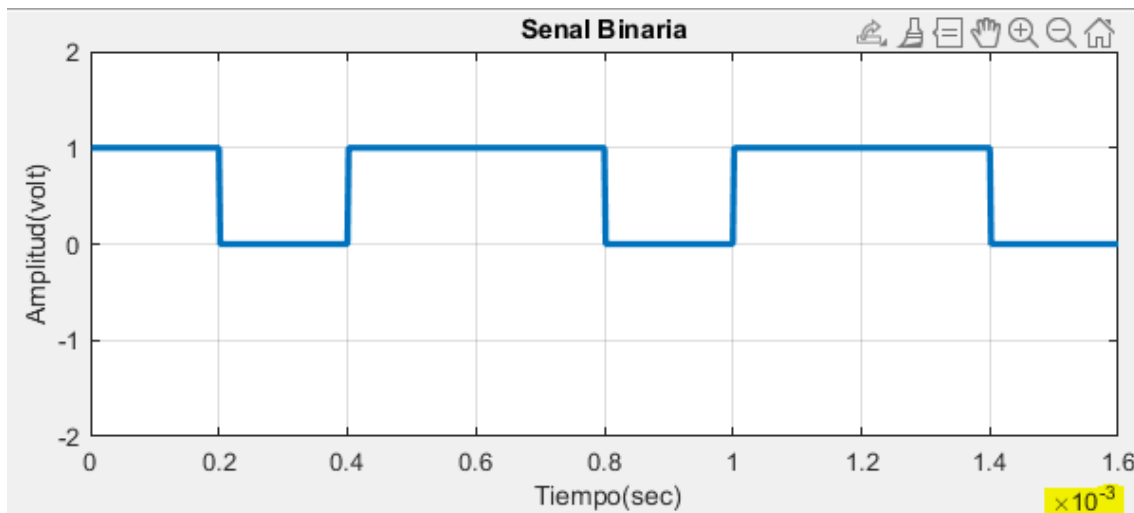
Pantallazo de grafica

Funcion sinc(t), con $G = 1$ (Numero del grupo)



Funcion sinc(t), con $G = 5$



8-Se tiene la siguiente señal Banda Base “10110110”

Si el ancho del pulso de cada bit es 0.2 milisegundos, calcular la Velocidad binaria y el Ancho de banda de esta señal.

Comandos

```
% GRAFICA DE LA SEÑAL BINARIA
clc, clear;

% Señal binaria en bits
x = [1 0 1 1 0 1 1 0];
% Ancho de pulso de cada bit
ap = 200e-6;

% Grafica de la señal binaria
bit=[];
for n=1:length(x)
    if x(n)==1;
        se=ones(1,100);
    else x(n)==0;
        se=zeros(1,100);
    end
    bit=[bit se];
end

t = ap /100: ap /100: 100*length(x)*(ap /100);
plot(t,bit, 'lineWidth',2.5);
grid on;
axis([ 0 ap *length(x) -2 2]);

ylabel('Amplitud(volt)');
xlabel('Tiempo(sec)');
title('Senal Binaria');
```

```
% CALCULO DE LA VELOCIDAD BINARIA
vb = 1/ap;
X = sprintf('La velocidad binaria es %d bps',vb);
disp(X)

% CALCULO DEL ANCHO DE BANDA DE LA SEÑAL
bw = 8*vb;
Y = sprintf('La señal tiene 8 bits, entonces el ancho de banda
sera %d bps',bw);
disp(Y)
```