

---

# TRABAJO PRÁCTICO N°1 DE ANÁLISIS Y PROCESAMIENTO DE SEÑALES

## Table of Contents

|  |   |
|--|---|
| <i>INTEGRANTES:</i> .....              | 1 |
| EJERCICIO A .....                      | 1 |
| <i>Definición De Variables</i> .....   | 1 |
| <i>Graficación De Resultados</i> ..... | 2 |
| EJERCICIO B .....                      | 2 |
| <i>Definición De Variables</i> .....   | 2 |
| <i>Graficación De Resultados</i> ..... | 3 |
| EJERCICIO C .....                      | 4 |
| <i>Definición De Variables</i> .....   | 4 |
| <i>Graficación De Resultados</i> ..... | 4 |
| <i>Graficación De Resultados</i> ..... | 5 |
| CONCLUSIONES .....                     | 6 |

## ***INTEGRANTES:***

- Di Leo Catherine
- Scicchitano Román
- Schejtman Román (Responsable)
- Sosa Iribarren Nicolás
- Zantleifer Olivia

## **EJERCICIO A**

### ***Definición De Variables***

```
clc
clear all
close all
fs = 150;           % Frecuencia de muestreo
Ts = 1/fs;          % Tiempo de muestreo
t1 = 0.4 : Ts : 0.5; % Variable de tiempo para el primer intervalo
t2 = 0.5 : Ts : 0.8; % Variable de tiempo para el segundo intervalo
% Cambiando los valores de inicio y final de cada intervalo, se controla
cómo se grafican
```

TRABAJO PRÁCTICO N°1 DE  
ANÁLISIS Y PROCESAMIENTO  
DE SEÑALES

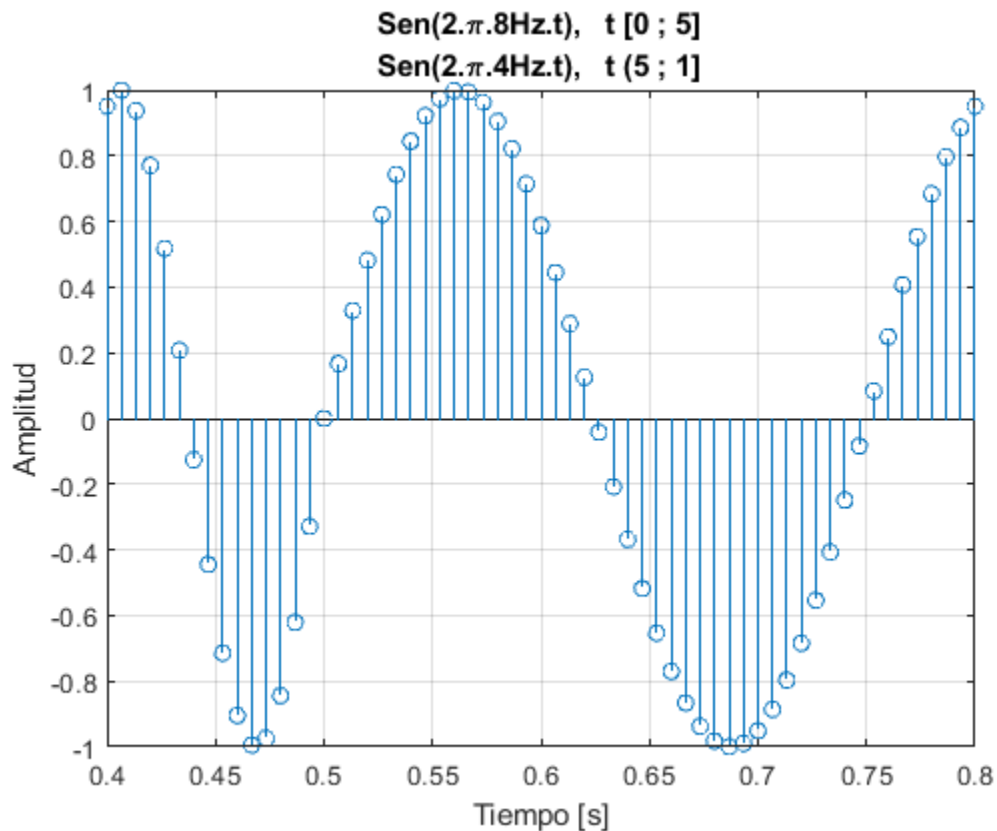
---

```
f1 = 8;           % Frecuencia 1
f2 = 4;           % Frecuencia 2

y1 = sin(2 * pi * f1 * t1);      % Muestreo del primer intervalo
y2 = sin(2 * pi * f2 * t2);      % Muestreo del segundo intervalo
Y = [y1 , y2];
T = [t1 , t2];
```

## Graficación De Resultados

```
figure
stem(T , Y)      % Resultado del muestreo
xlabel('Tiempo [s]');
ylabel('Amplitud');
title('Sen(2.\pi.8Hz.t), t [0 ; 5]' , 'Sen(2.\pi.4Hz.t), t (5 ; 1]',
'FontWeight','bold')
grid on
```



## EJERCICIO B

### Definición De Variables

```
A1 = 5;           % Amplitud de la primera señal
A2 = 2;           % Amplitud de la segunda señal
```

TRABAJO PRÁCTICO N°1 DE  
ANÁLISIS Y PROCESAMIENTO  
DE SEÑALES

---

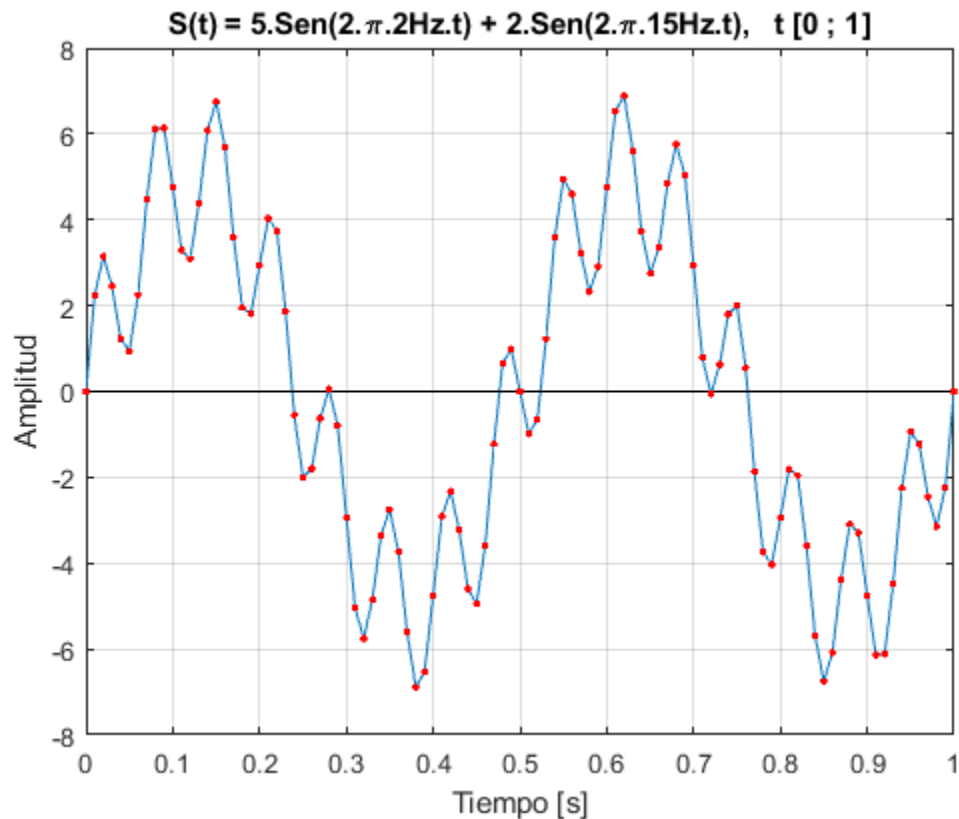
```
f1 = 2;           % Frecuencia 1
f2 = 15;          % Frecuencia 2
fs = 100;         % Frecuencia de muestreo
t = 0:1/fs:1;     % Intervalo de tiempo
```

```
Suma_De_Ondas = A1 * sin(2 * pi * f1 * t) + A2 * sin(2 * pi * f2 * t);
```

## ***Graficación De Resultados***

```
figure()
plot(t, Suma_De_Ondas);
xlabel('Tiempo [s]');
ylabel('Amplitud');
title('S(t) = 5.Sen(2.\pi.2Hz.t) + 2.Sen(2.\pi.15Hz.t), t [0 ; 1]')
grid on;

hold on;
xLimits = xlim;
plot(xLimits, [0 0], 'k-');
plot(t, Suma_De_Ondas, 'r.');
hold off;
```



## EJERCICIO C

### *Definición De Variables*

```
f0 = 3;          % Frecuencia fundamental = 3Hz
f = [-3, -2, -1, 0, 1, 2, 3]*f0;          % Vector de frecuencias de la señal
X = [0.2, 0.7, 0.5, 1, 0.5, 0.7, 0.2];    % Coeficientes de la Serie de
fourier correspondientes a cada frecuencia
fs = 100;        % Definimos la frecuencia de muestreo.
Ts = 1/fs;        % Tiempo de muestreo.
t = 0 : Ts : 2;   % Variable de tiempo
```

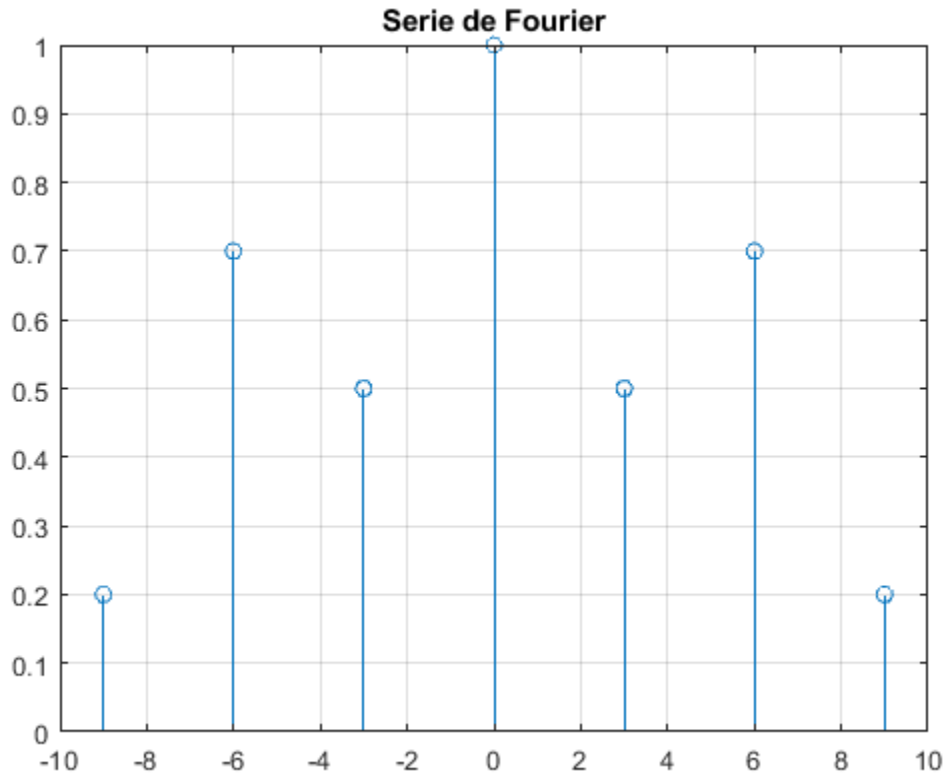
### *Graficación De Resultados*

```
figure()
stem(f , X), title('Serie de Fourier');    %Gráfico de la serie de fourier
discreta
grid on;
```

%Como la transformada de Fourier de un coseno es  $1/2 * [\delta(f-f_0) + \delta(f+f_0)]$ ,  
%hay que agregar por fuera de cada parte de mi señal que sea un coseno, un 2\*  
%para corregir ese 1/2 que se "agrega" al ser transformado, y que el valor  
%de los coeficientes para cada frecuencia no sea la mitad del que debería.  
%Además es necesario agregarle multiplicando también el valor que uno  
quisiera  
%que tenga el coeficiente de Fourier para ese coseno.

%Visto de otra forma un poco más clara, si el coseno es escrito como forma  
exponencial compleja:  
 $\cos(2\pi f_0 t) = 1/2 * e^{j2\pi f_0 t} + 1/2 * e^{-j2\pi f_0 t}$   
%a simple vista se puede ver que es igual su desarrollo en serie de  
%Fourier, y que  $f_0$  y  $-f_0$  tienen los dos un coeficiente de 1/2. Si a esto se  
lo  
%multiplica por 2 y por el coeficiente que yo quiero, en este caso 0.7 por  
%ejemplo:  $2 * 0.7 * \cos(2\pi f_0 t) = 2 * 0.7 * (1/2 * e^{j2\pi f_0 t} + 1/2 * e^{-j2\pi f_0 t})$   
%entonces:  $2 * 0.7 * \cos(2\pi f_0 t) = 0.7 * e^{j2\pi f_0 t} + 0.7 * e^{-j2\pi f_0 t}$ )  
%la frecuencia queda con coeficiente 0.7, que es lo que se quería obtener.

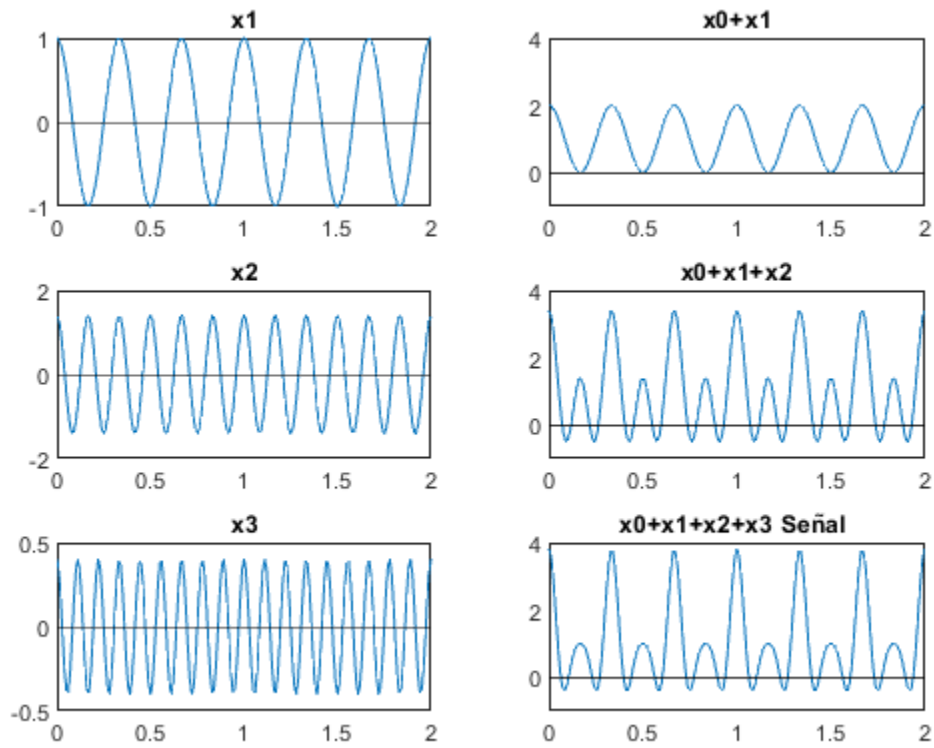
```
% Definimos los distintos "componentes" de la señal para después sumarlos
x0 = X(4);          % El DC offset, frecuencia = 0, correspond a una corriente
continua
x1 = 2 * X(5) * cos(2 * pi * f(5) * t);    % Componente con la frecuencia
fundamental
x2 = 2 * X(6) * cos(2 * pi * f(6) * t);    % Componente con el primer
armónico
x3 = 2 * X(7) * cos(2 * pi * f(7) * t);    % Componente con el segundo
armónico
```



## Graficación De Resultados

Graficamos los componentes de la señal, y se muestra la suma paso a paso

```
VECTOR_DE_VALORES_EN_X = 0: 1 : 2;           %Sirve para graficar el eje x
Aux_1 = 0 : 2;                               %Sirve para graficar el eje x
Aux_2 = 0 * Aux_1;                           %Sirve para graficar el eje x
figure()
subplot(3, 2, 1)
    plot(t, x1, Aux_1, Aux_2, 'k'), title('x1');
subplot(3, 2, 3)
    plot(t, x2, Aux_1, Aux_2, 'k'), title('x2');
subplot(3, 2, 5)
    plot(t, x3, Aux_1, Aux_2, 'k'), title('x3');
    subplot(3, 2, 2)
        plot(t, x0+x1, Aux_1, Aux_2, 'k'), title('x0+x1'), axis([0 2 -1 4]);
subplot(3, 2, 4)
    plot(t, x0+x1+x2, Aux_1, Aux_2, 'k'), title('x0+x1+x2'), axis([0 2 -1 4]);
subplot(3, 2, 6)
    plot(t, x0+x1+x2+x3, Aux_1, Aux_2, 'k'), title('x0+x1+x2+x3 Señal '),
axis([0 2 -1 4]);
```



## CONCLUSIONES

*Published with MATLAB® R2024a*