

# Actividad #2

## Generación y Visualización de Señales Senoidales en MATLAB

- Nombre:
- Fecha:
- Repositorio: <https://github.com/vasanza/SSE>

### Table of Contents

Descripción:	1
Objetivos:	1
Desarrollo de la Actividad:	1
Paso 1: Limpiar variables y línea de comandos:	1
Paso 2: Crear un código básico:	1
Paso 3: Mostrar resultados con plot:	2

### Descripción:

Esta actividad consiste en generar señales senoidales sintéticas con diferentes fases utilizando MATLAB. Se analiza cómo varía la señal con la fase y se visualizan los resultados mediante gráficas. La actividad refuerza conceptos clave como frecuencia, fase, muestreo y visualización de señales.

### Objetivos:

- Generar señales senoidales sintéticas modificando sus parámetros.
- Comprender el efecto de la fase en la forma de una señal periódica.
- Utilizar funciones de graficación para representar señales en el tiempo.
- Personalizar gráficas con títulos, etiquetas y leyendas.

## Desarrollo de la Actividad

### Paso 1: Limpiar variables y línea de comandos

```
clear % Borrar variables en el workspace y libera memoria RAM
clc % Limpia el Command Window
raiz = cd; % Linux command (address current folder)
```

### Paso 2: Crear un código básico

```
% Parámetros
f = 10;           % Frecuencia en Hz
A = 1;           % Amplitud
fase1 = 0; fase2 = 10; % Fase
T = 1;           % Duración en segundos
fs = 1000;       % Frecuencia de muestreo en Hz

% Vector de tiempo
t = 0:1/fs:T;

% Señal senoidal
```

```
y1 = A * sin(2 * pi * f * t + fase1);  
y2 = A * sin(2 * pi * f * t + fase2);
```

### Paso 3: Mostrar resultados con plot

```
figure  
plot(t,y1) %Señal 1  
hold on  
plot(t,y2) %Señal 2  
title("Señal Senosoidal")  
xlabel("Tiempo (Seg)")  
ylabel("Value")  
legend("Señal 1, fase1 = 0", "Señal 1, fase2 = 10")
```

