

Actividad #7

Generación, Visualización y Almacenamiento de dinámico de Datasets CSV

- Nombre:
- Fecha:
- Reposiroty: <https://github.com/vasanza/SSE>
- Refrence: https://github.com/vasanza/Matlab_Code/tree/main

Table of Contents

Descripción:	1
Objetivos:	1
Copia la actividad en tu respaldo:	1
Desarrollo de la Actividad:	2
Paso 1: Borrar variables en el workspace y limpiar cmd:	2
Paso 2: Crear un codigo basico:	2
Paso 3: TimeStamp (dd-MMM-yyyy HH:mm:ss):	2
Paso 4: Mostrar resultados con plot:	3
Paso 5: crear dataset usando tablas:	4
Paso 6: Guardar los carchivos VSC en el computador:	5

Descripción:

Objetivos:

- **Comprender la generación de señales senoidales** a partir de parámetros ajustables como frecuencia, amplitud y fase.
- **Visualizar múltiples señales combinadas** (suma y resta) utilizando representaciones gráficas en función del tiempo.
- **Incorporar un sistema de TimeStamp** para alinear las señales con una referencia temporal precisa.
- **Construir datasets estructurados** en forma de tablas que integren señales y marcas de tiempo.
- **Exportar datasets generados en MATLAB** a archivos CSV utilizando funciones personalizadas.
- **Aplicar buenas prácticas de organización de proyectos** usando carpetas dedicadas (src, respaldo local) y funciones externas para modularizar el código.

Copia la actividad en tu respaldo

```
%Configuracion de carpeta ./src para librerias
addpath(genpath('./src'));

% Definir rutas
miRespaldo = 'C:\Desktop\SSE_vic'; %<=====
repositorio = 'C:\Desktop\SSE\2025';%<=====

if true
    % repositorio -> respaldo
    git_sse(miRespaldo)
```

```

else
    % Mombre de la carpeta de la Actividad en el repositorio
    nombreCarpeta = string(split(cd, filesep));
    nombreCarpeta = nombreCarpeta(end) % Nombre de la carpeta
    % Regresar al repositorio
    cd(fullfile(repositorio,nombreCarpeta))
end

```

Desarrollo de la Actividad

Paso 1: Borrar variables en el workspace y limpiar cmd

```

clear % Borrar variables en el workspace y libera memoria RAM
clc % Limpia el Command Window
addpath(genpath('./src'));

```

Paso 2: Crear un codigo basico

```

% Parámetros
f = 2;           % Frecuencia de la señal en Hz
A1 = 3;          % Amplitud1
A2 = 16;         % Amplitud1
fase1 = 0;       % Fase1
fase2 = 10;      % Fase2
T = 10;          % Duración en segundos
fsmin = f*2;     % Frecuencia de muestreo mínima (Nyquist-Shannon)
fs = 20;         % Frecuencia de muestreo en Hz

```

Documentacion de la senal senoidal

```
help senal_senoidal
```

```

--- SEÑAL SENOIDAL ---
function [y] = senal_senoidal(frecuencia,amplitud,fase,tiempo,fs)

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

```

```

% Usamos la funcion: senal_senoidal(frecuencia,amplitud,fase,tiempo,fs)
y1 = senal_senoidal(f,A1,fase1,T,fs)';
y2 = senal_senoidal(f,A2,fase2,T,fs)';
y3 = y1 + y2;
y4 = y1 - y2;

```

Paso 3: TimeStamp (dd-MMM-yyyy HH:mm:ss)

```
simulationTime = seconds(0:1/fs:T)
```

```
simulationTime = 1×201 duration
```

0 sec 0.05 sec 0.1 sec 0.15 sec 0.2 sec 0.25 sec 0. ...

```
startTime = datetime('now')
```

```
startTime = datetime  
13-Jun-2025 21:51:37
```

```
timeStamp = (startTime + simulationTime)'
```

```
timeStamp = 201x1 datetime  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
13-Jun-2025 21:51:37  
⋮
```

```
timeStamp.Format
```

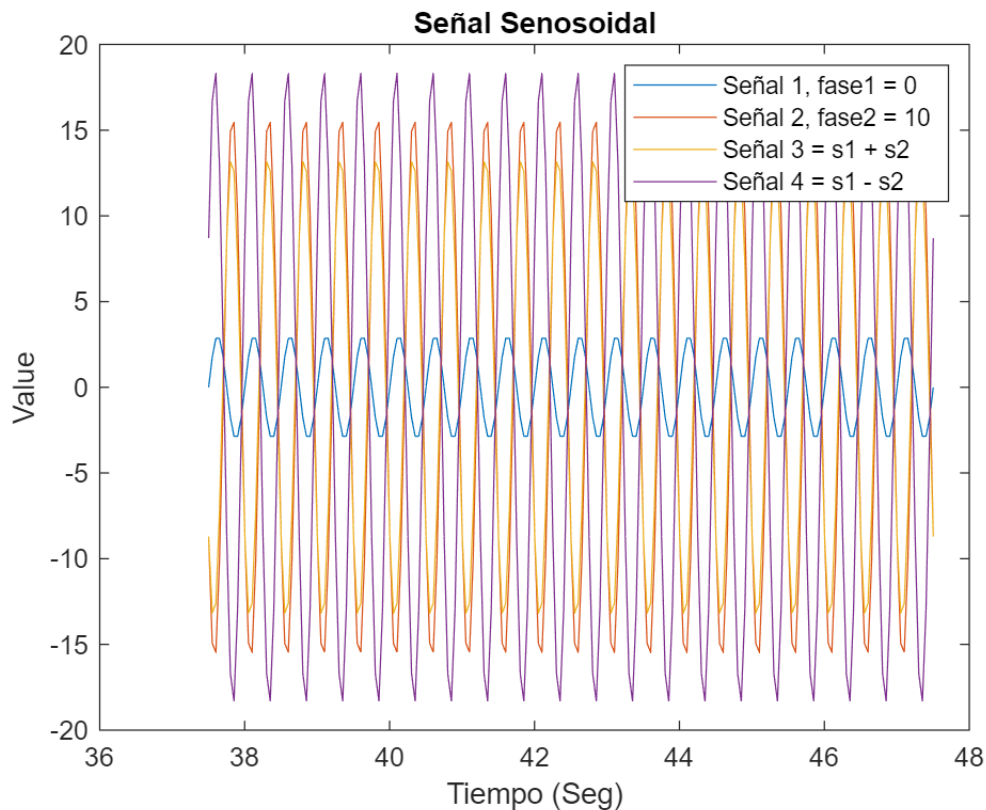
```
ans =  
'dd-MMM-uuuu HH:mm:ss'
```

```
size(timeStamp) %filas, columnas
```

```
ans = 1x2  
201      1
```

Paso 4: Mostrar resultados con plot

```
figure  
plot(timeStamp.Second,y1) %Señal 1  
hold on  
plot(timeStamp.Second,y2) %Señal 2  
plot(timeStamp.Second,y3) %Señal 3  
plot(timeStamp.Second,y4) %Señal 4  
title("Señal Senosoidal")  
xlabel("Tiempo (Seg)")  
ylabel("Value")  
legend("Señal 1, fase1 = 0", "Señal 2, fase2 = 10", "Señal 3 = s1 + s2", "Señal 4 = s1 - s2")  
hold off
```



Paso 5: crear dataset usando tablas

```
%cell array
VariableNames = {'TimeStamp', 'y1 = Sin1', 'y2 = Sin2', 'Y3 = y1 + y2'...
, 'Y4 = y1 - y2'}
```

```
VariableNames = 1x5 cell
'TimeStamp' 'y1 = Sin1' 'y2 = Sin2' 'Y3 = y1 + y2' 'Y4 = y1 - y2'
```

```
% Para la tabla el timeStamp debe ser una columna
dataset = table(timeStamp, y1, y2, y3, y4, 'VariableNames', VariableNames)
```

```
dataset = 201x5 table
```

	TimeStamp	y1 = Sin1	y2 = Sin2	Y3 = y1 + y2	Y4 = y1 - y2
1	13-Jun-2025 21:51:37	0	-8.7043	-8.7043	8.7043
2	13-Jun-2025 21:51:37	1.7634	-14.9331	-13.1697	16.6964
3	13-Jun-2025 21:51:37	2.8532	-15.4579	-12.6047	18.3110
4	13-Jun-2025 21:51:37	2.8532	-10.0783	-7.2251	12.9315
5	13-Jun-2025 21:51:37	1.7634	-0.8491	0.9142	2.6125
6	13-Jun-2025 21:51:37	0	8.7043	8.7043	-8.7043
7	13-Jun-2025 21:51:37	-1.7634	14.9331	13.1697	-16.6964
8	13-Jun-2025 21:51:37	-2.8532	15.4579	12.6047	-18.3110

	TimeStamp	y1 = Sin1	y2 = Sin2	Y3 = y1 + y2	Y4 = y1 - y2
9	13-Jun-2025 21:51:37	-2.8532	10.0783	7.2251	-12.9315
10	13-Jun-2025 21:51:37	-1.7634	0.8491	-0.9142	-2.6125
11	13-Jun-2025 21:51:38	-0	-8.7043	-8.7043	8.7043
12	13-Jun-2025 21:51:38	1.7634	-14.9331	-13.1697	16.6964
13	13-Jun-2025 21:51:38	2.8532	-15.4579	-12.6047	18.3110
14	13-Jun-2025 21:51:38	2.8532	-10.0783	-7.2251	12.9315

⋮

```
dataset.Properties.VariableNames
```

```
ans = 1x5 cell
'TimeStamp' 'y1 = Sin1' 'y2 = Sin2' 'Y3 = y1 + y2' 'Y4 = y1 - y2'
```

```
height(dataset)
```

```
ans = 201
```

Paso 6: Guardar los archivos VSC en el computador

```
filename = 'dataset.csv';
fSave_dataset(filename,dataset)
```