

Actividad #8

Almacenamiento, Carga y Visualización Dinámica de Datasets en Formato CSV usando MATLAB

- Nombre:
- Fecha:
- Repositorio: <https://github.com/vasanza/SSE>
- Reference: 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 código básico:	2
Paso 3: TimeStamp (dd-MMM-yyyy HH:mm:ss.SSS):	2
Paso 4: Mostrar resultados con plot:	3
Paso 5: crear dataset usando tablas:	4
Paso 6: Guardar los archivos csv en el computador:	5
Paso 7: Leer archivos csv y retornar una tabla:	5
Paso 8: Extraer y graficar los datos cargados:	6

Descripción:

Objetivos:

- **Generar señales senoidales con parámetros personalizables** de frecuencia, amplitud y fase.
- **Asignar marcas de tiempo con precisión milisegundos** para simular un entorno de adquisición de datos.
- **Crear y guardar datasets en formato CSV** utilizando tablas en MATLAB.
- **Leer archivos CSV y convertirlos en tablas manipulables**, permitiendo la reutilización de datos.
- **Visualizar y analizar las señales cargadas** para validar el proceso completo de exportación e importación.

Copia la actividad en tu respaldo

```
%Configuración de carpeta ./src para librerías
addpath(genpath('./src'));

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

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

```

% Nombre 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 código básico

```

% 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

```

Documentación de la señal 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 función: 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.SSS)

```
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. ...

```

```
%setear formto de tiempo en milisegundos
```

```
startTime = datetime  
14-Jun-2025 08:22:38.197
```

```
startTime = datetime('now', 'Format', 'dd-MMM-yyyy HH:mm:ss.SSS')  
timeStamp = (startTime + simulationTime)'
```

```
timeStamp = 201x1 datetime  
14-Jun-2025 08:22:38.197  
14-Jun-2025 08:22:38.247  
14-Jun-2025 08:22:38.297  
14-Jun-2025 08:22:38.347  
14-Jun-2025 08:22:38.397  
14-Jun-2025 08:22:38.447  
14-Jun-2025 08:22:38.497  
14-Jun-2025 08:22:38.547  
14-Jun-2025 08:22:38.597  
14-Jun-2025 08:22:38.647  
⋮
```

```
timeStamp.Format
```

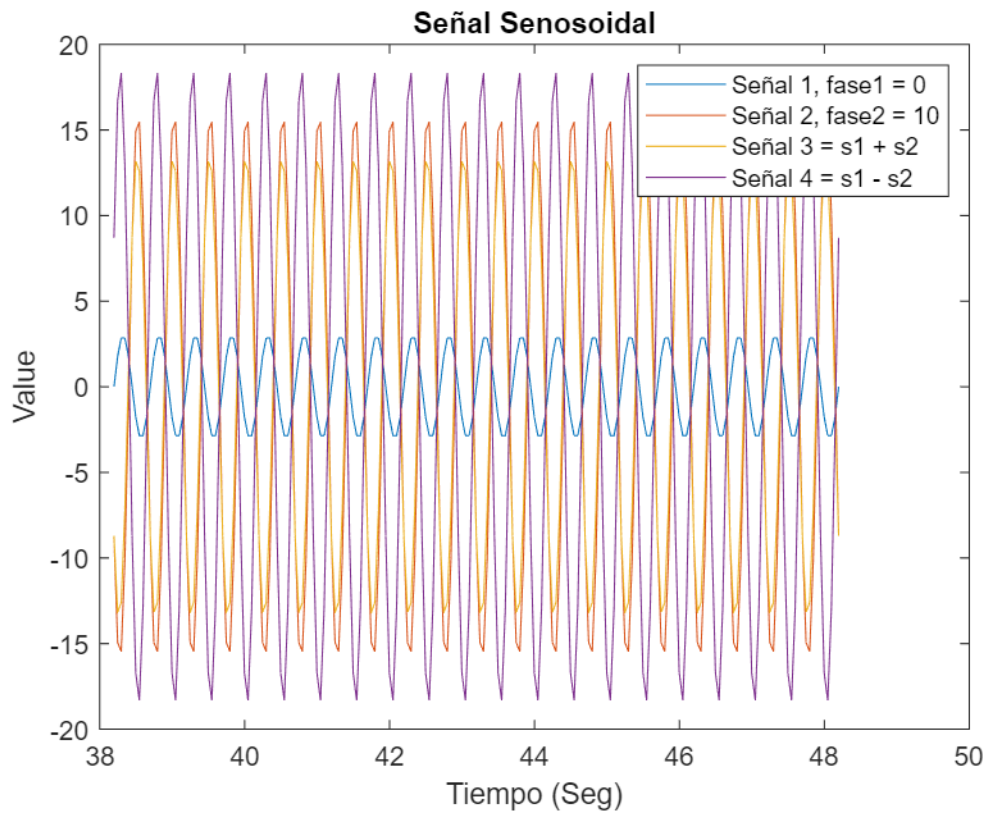
```
ans =  
'dd-MMM-yyyy HH:mm:ss.SSS'
```

```
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	14-Jun-2025 08:22:38.197	0	-8.7043	-8.7043	8.7043
2	14-Jun-2025 08:22:38.247	1.7634	-14.9331	-13.1697	16.6964
3	14-Jun-2025 08:22:38.297	2.8532	-15.4579	-12.6047	18.3110
4	14-Jun-2025 08:22:38.347	2.8532	-10.0783	-7.2251	12.9315
5	14-Jun-2025 08:22:38.397	1.7634	-0.8491	0.9142	2.6125
6	14-Jun-2025 08:22:38.447	0	8.7043	8.7043	-8.7043
7	14-Jun-2025 08:22:38.497	-1.7634	14.9331	13.1697	-16.6964
8	14-Jun-2025 08:22:38.547	-2.8532	15.4579	12.6047	-18.3110

	TimeStamp	y1 = Sin1	y2 = Sin2	Y3 = y1 + y2	Y4 = y1 - y2
9	14-Jun-2025 08:22:38.597	-2.8532	10.0783	7.2251	-12.9315
10	14-Jun-2025 08:22:38.647	-1.7634	0.8491	-0.9142	-2.6125
11	14-Jun-2025 08:22:38.697	0	-8.7043	-8.7043	8.7043
12	14-Jun-2025 08:22:38.747	1.7634	-14.9331	-13.1697	16.6964
13	14-Jun-2025 08:22:38.797	2.8532	-15.4579	-12.6047	18.3110
14	14-Jun-2025 08:22:38.847	2.8532	-10.0783	-7.2251	12.9315

⋮

```
%Nombres de las variables
dataset.Properties.VariableNames
```

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

```
%cuantas columnas tiene el dataset
length(dataset.Properties.VariableNames)
```

```
ans = 5
```

```
%cuantas filas tiene el dataset
height(dataset) % pata Table no funcion el length
```

```
ans = 201
```

```
%cuantas filas tiene el dataset
size(dataset,1) % 1 para filas
```

```
ans = 201
```

```
%cuantas columnas tiene el dataset
size(dataset,2) % 2 para columnas
```

```
ans = 5
```

Paso 6: Guardar los archivos csv en el computador

```
filename = 'dataset.csv';
fSave_dataset(filename,dataset)
%Limpiar workspace
clear
```

Paso 7: Leer archivos csv y retornar una tabla

```
rawData = fLoad_dataset('dataset.csv');
```

Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating variable names for the table. The original column headers are saved in the VariableDescriptions property. Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.

```
%muestra las x primera columnas
disp(rawData(1:10,:));
```

TimeStamp	y1_Sin1	y2_Sin2	Y3_Y1_Y2	Y4_Y1_Y2
14-Jun-2025 08:22:38.197	0	-8.7043	-8.7043	8.7043
14-Jun-2025 08:22:38.247	1.7634	-14.933	-13.17	16.696
14-Jun-2025 08:22:38.297	2.8532	-15.458	-12.605	18.311
14-Jun-2025 08:22:38.347	2.8532	-10.078	-7.2251	12.931
14-Jun-2025 08:22:38.397	1.7634	-0.84915	0.91421	2.6125
14-Jun-2025 08:22:38.447	0	8.7043	8.7043	-8.7043
14-Jun-2025 08:22:38.497	-1.7634	14.933	13.17	-16.696
14-Jun-2025 08:22:38.547	-2.8532	15.458	12.605	-18.311
14-Jun-2025 08:22:38.597	-2.8532	10.078	7.2251	-12.931
14-Jun-2025 08:22:38.647	-1.7634	0.84915	-0.91421	-2.6125

Paso 8: Extraer y graficar los datos cargados

```
rawData.Properties
```

```
ans =
    TableProperties with properties:

        Description: ''
        UserData: []
        DimensionNames: {'Row' 'Variables'}
        VariableNames: {'TimeStamp' 'y1_Sin1' 'y2_Sin2' 'Y3_Y1_Y2' 'Y4_Y1_Y2'}
        VariableDescriptions: {'TimeStamp' 'y1 = Sin1' 'y2 = Sin2' 'Y3 = y1 + y2' 'Y4 = y1 - y2'}
        VariableUnits: {}
        VariableContinuity: []
        RowNames: {}
        CustomProperties: No custom properties are set.
        Use addprop and rmprop to modify CustomProperties.
```

```
t = datetime(rawData.TimeStamp);
y1 = rawData.y1_Sin1;
y2 = rawData.y2_Sin2;
y3 = rawData.Y3_Y1_Y2;
y4 = rawData.Y4_Y1_Y2;

figure;
plot(t,y1); hold on;
plot(t,y2);
plot(t,y3);
plot(t,y4);

legend('y1 = Sin1', 'y2 = Sin2', 'Y3 = y1 + y2', 'Y4 = y1 - y2');
xlabel('Tiempo');
ylabel('Valores');
title('Señales en función del tiempo');
grid on;
```

