

Actividad #9

Carga Masiva (data folder) y Visualización de Datos desde Archivos CSV

- Nombre:
- Fecha:
- Reposirot: <https://github.com/vasanza/SSE>
- Refrence: https://github.com/vasanza/Matlab_Code/tree/main
- Dataset: Open Energy Data Initiative (OEDI)
- Photovoltaic Data Acquisition (PVDAQ) Public Datasets

Table of Contents

Descripción:	1
Objetivos:	1
Tipo de Dataset a utilizar:	1
Copia la actividad en tu respaldo	2
Desarrollo de la Actividad	2
Paso 1: Borrar variables en el workspace y limpiar cmd	2
Paso 2: Configuración de carpeta ./src para librerías	2
Paso 3: Configuración de carpeta ./data para datasets	2
Paso 4: Buscar nombres de archivos .csv en ./data	2
Paso 5: Cargar automaticamente todos los archivos csv desde ./data	3
Paso 6: Extraer y graficar las variables	3

Descripción:

Objetivos:

- **Configurar una estructura de carpetas eficiente** para el manejo de librerías y datasets externos en MATLAB.
- **Detectar y cargar automáticamente múltiples archivos CSV** desde un directorio de trabajo.
- **Unificar y consolidar datos reales de mediciones fotovoltaicas** en una sola tabla para su análisis.
- **Extraer dinámicamente las variables contenidas** en los datasets sin necesidad de codificación manual.
- **Visualizar múltiples señales medidas en función del tiempo**, facilitando la interpretación de fenómenos fotovoltaicos.

Tipo de Dataset a utilizar:

- Cada columna representa una variable y cada fila representa una muestra
- La primera columna es el TimeStamp. Ejemplo: 'dd-MMM-yyyy HH:mm:ss.SSS'
- El arhivo debe estar en formato **.csv**, si es otro formato de debera convertir a .csv. Ejemplo: si es **.parquet**, se lo convierte con esta pagina <https://table.studio/convert/parquet/to/csv>
- Si existen multiples archivos .csv, estos se deberan colocar en la carpeta data, siempre que esten relacionadas a un mismo sistema. **Ejemplo de regresion:** consumo_enero_2025.csv, consumo_febrero_2025.csv, etc. **Ejemplo de clasificacion:** consumo_casa_enero_2025.csv, sonsumo_escuela_enero_2025.csv, etc.

	measured_...	inv_string0...	inv_string0...	inv_string0...	inv_string0...	inv_string0...
	Datetime	Number	Number	Number	Number	Number
1	measured_on	inv_string01...	inv_string01...	inv_string01...	inv_string01...	inv_string01...
2	2018-12-29...	0.0	0.0	123.891	0.0	0.0
3	2018-12-29...	0.0	0.0	123.875	0.0	0.0
4	2018-12-29...	0.0	0.0	121.844	0.0	0.0
5	2018-12-30...	0.0	0.0	122.75	0.0	0.0
6	2018-12-30...	0.0	0.0	121.672	0.0	0.0
7	2018-12-31...	0.0	0.0	123.688	0.0	0.0

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
    % 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
```

Paso 2: Configuración de carpeta ./src para librerias

```
addpath(genpath('./src'));
```

Paso 3: Configuración de carpeta ./data para datasets

```
datapath = fullfile('./data/');
```

Paso 4: Buscar nombres de archivos .csv en ./data

```
filename = FindCSV(datapath)
```

```
filename = 5x1 struct
```

Fields	name	folder	date	bytes	isdir	datenum
1	'Copy_2_of_...	'C:\Users\v...	'14-Jun-202...	18700465	0	7.3978e+05

Fields	name	folder	date	bytes	isdir	datenum
2	'Copy_3_of_...	'C:\Users\lv...	'14-Jun-202...	18700465	0	7.3978e+05
3	'Copy_4_of_...	'C:\Users\lv...	'14-Jun-202...	18700465	0	7.3978e+05
4	'2105_inv01...	'C:\Users\lv...	'14-Jun-202...	18700465	0	7.3978e+05
5	'Copy_of_21...	'C:\Users\lv...	'14-Jun-202...	18700465	0	7.3978e+05

```
maxFiles = size(filename,1)
```

```
maxFiles = 5
```

Paso 5: Cargar automaticamente todos los archivos csv desde ./data

```
allData = [];
for i=1:maxFiles
    nameFile = filename(i).name;
    pathFile = strcat(datapath, nameFile)
    rawData = fLoad_dataset(pathFile);
    allData = [allData; rawData];
end
```

```
pathFile =
'..\data\Copy_2_of_2105_inv01_data.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.

```
pathFile =
'..\data\Copy_3_of_2105_inv01_data.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.

```
pathFile =
'..\data\Copy_4_of_2105_inv01_data.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.

```
pathFile =
'..\data\2105_inv01_data.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.

```
pathFile =
'..\data\Copy_of_2105_inv01_data.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.

```
clear rawData nameFile filename;
```

Paso 6: Extraer y graficar las variables

```
allData.Properties
```

```
ans =
    TableProperties with properties:
```

```
    Description: ''
```

```

        UserData: []
        DimensionNames: {'Row' 'Variables'}
        VariableNames: {1×6 cell}
        VariableDescriptions: {1×6 cell}
        VariableUnits: {}
        VariableContinuity: []
        RowNames: {}
        CustomProperties: No custom properties are set.
        Use addprop and rmprop to modify CustomProperties.

```

```

t = datetime(allData.measured_on);
VariableNames = allData.Properties.VariableNames

```

```

VariableNames = 1×6 cell
'measured_on' 'inv_string01_ac_output_kwh_inv_150164' 'inv_string01_ac_output__p' ...

```

```

MaxVariables = size(VariableNames,2);

% 2 por que el timeStamp ya fue leído
dataArray = [];
for i=2:MaxVariables
    variable = allData{:,VariableNames(i)};
    dataArray = [dataArray, variable];
end

figure;
maxSamples = size(dataArray,1);
samples = 121610

```

```

samples = 121610

```

```

plot(t(1:samples,:),dataArray(1:samples,:)); % las 1k primeras filas

legend(string(VariableNames(2:end)));
xlabel('Tiempo');
ylabel('Valores');
title('Señales en función del tiempo');
grid on;

```

