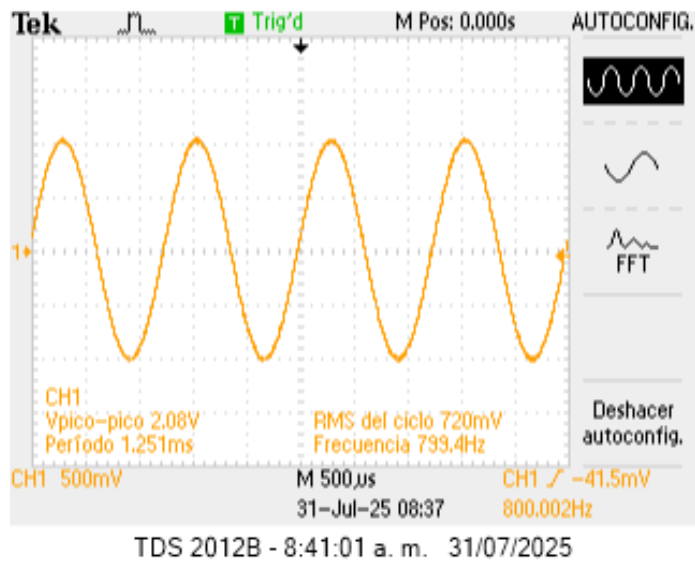


## SENO 1

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz
Vp	1

SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A		
				d.B	Vrms	VP(v)
Sen 1	0	1	800	-2,99	0,70876131	1,00233985



### GRAFICA MATLAB CON CODIGO

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/100000:2*T;
```

```
A1=1;
```

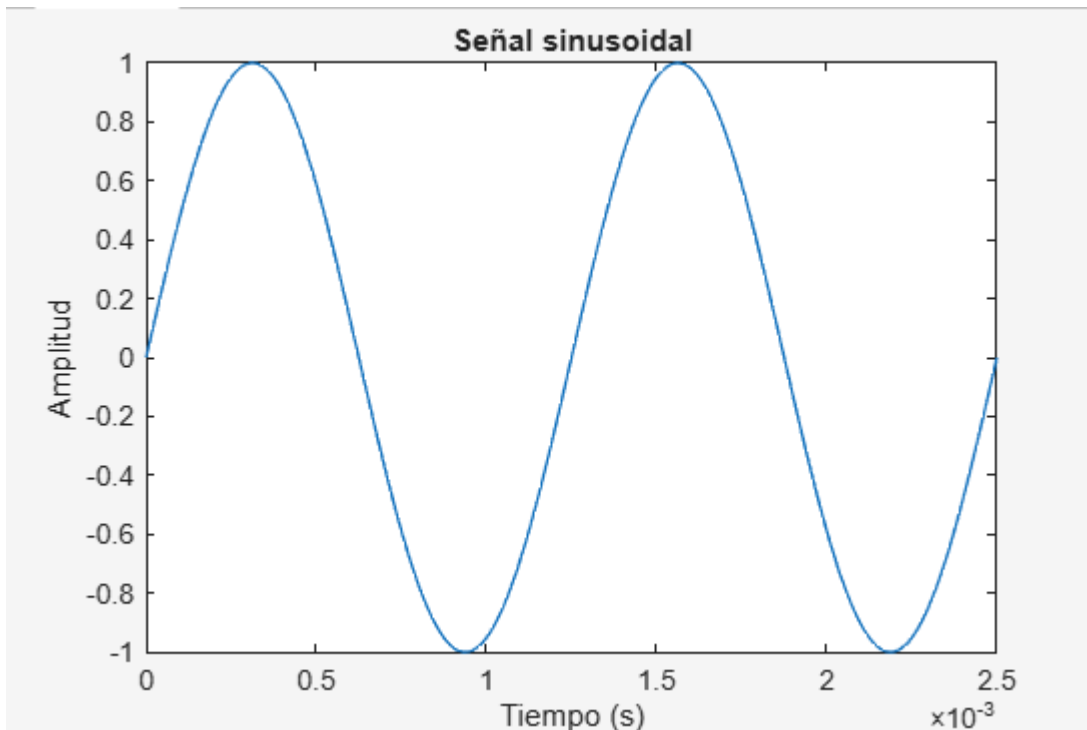
```
v1=A1*sin(2*pi*f*t);
```

```
plot(t,v1);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal sinusoidal');
```



### GRAFICA DE MEDIDOS VP

### GRAFICA MATLAB CON DATOS EXPORTADO

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOTIEMPO1.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOTIEMPO1.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos(:,4); % Columna 4 = Tiempo
```

```
v = datos(:,5); % Columna 5 = Voltaje
```

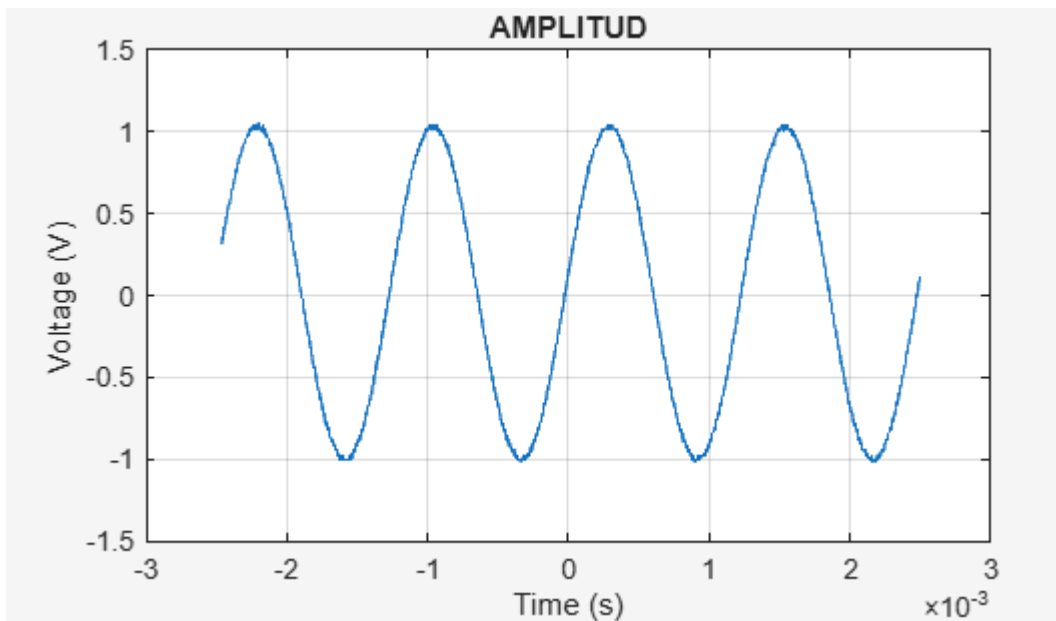
% Graficar

```
plot(t, v);
```

```

xlabel('Time (s)');
ylabel('Voltage (V)');
title('AMPLITUD');
grid on;

```



### **DOMINIO FRECUENCIA**

```

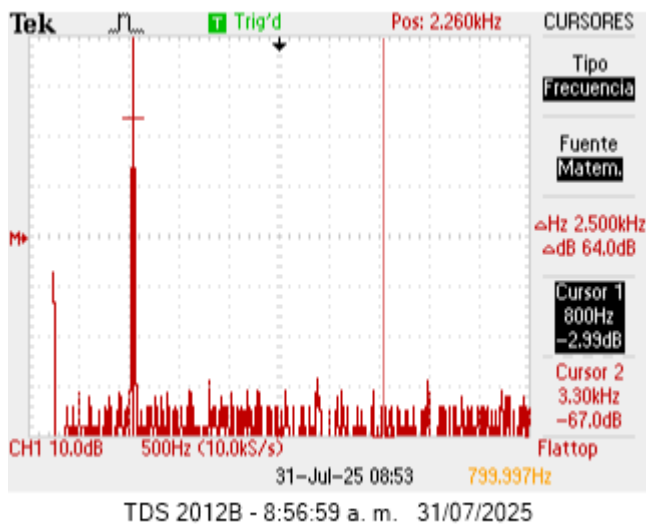
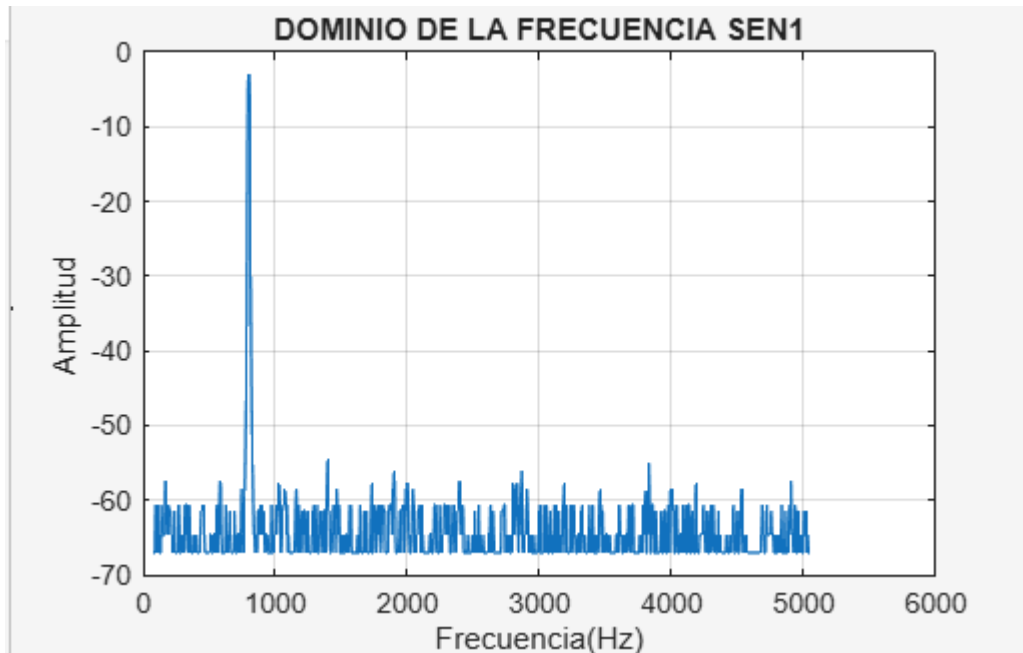
% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOFRECUENCIAS1.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos
datos = readtable('DOMINIOFRECUENCIAS1.csv', opts);
% Convertir las columnas correctas (4 y 5) a vectores
t = datos(:,4); % Columna 4 = Tiempo
v = datos(:,5); % Columna 5 = Voltaje
% Graficar
plot(t, v);

```

```

xlabel('Frecuenciav(Hz)');
ylabel('Amplitud');
title('DOMINIO DE LA FRECUENCIA SEN1');
grid on;

```



**GRAFICA FFT (TEORICA) TRAFSNORMADS DE FOURIER QUE SE TOMA CON LOS VALORES DEL TIEMPO Y SE TRANSFORMA A FRECUENCIA**

```

% Parámetros de la señal

T = 0.00125;      % Periodo (s)

f = 1/T;          % Frecuencia (Hz) → 800 Hz

Fs = 1e5;         % Frecuencia de muestreo (100 kHz)

t = 0:1/Fs:2*T;    % Tiempo total de 2 ciclos (0 a 0.0025 s)

A1 = 1;           % Amplitud de la señal

% Generar señal senoidal

v1 = A1 * sin(2*pi*f*t);

% Cálculo de la FFT

N = length(v1);    % Número total de muestras

Y = fft(v1);        % Transformada rápida de Fourier

Y_mag = abs(Y)/N;   % Magnitud normalizada

Y_mag(2:end-1) = 2*Y_mag(2:end-1); % Duplicar componentes (menos DC y Nyquist)

% Escala de frecuencia

f_axis = (0:N-1)*(Fs/N); % Vector de frecuencias asociado a la FFT

% Convertir magnitud a decibelios

Y_dB = 20*log10(Y_mag + eps); % eps evita log(0) y errores numéricos

% Graficar el espectro (solo la mitad positiva)

plot(f_axis(1:N/2), Y_dB(1:N/2), 'r', 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dBV)');

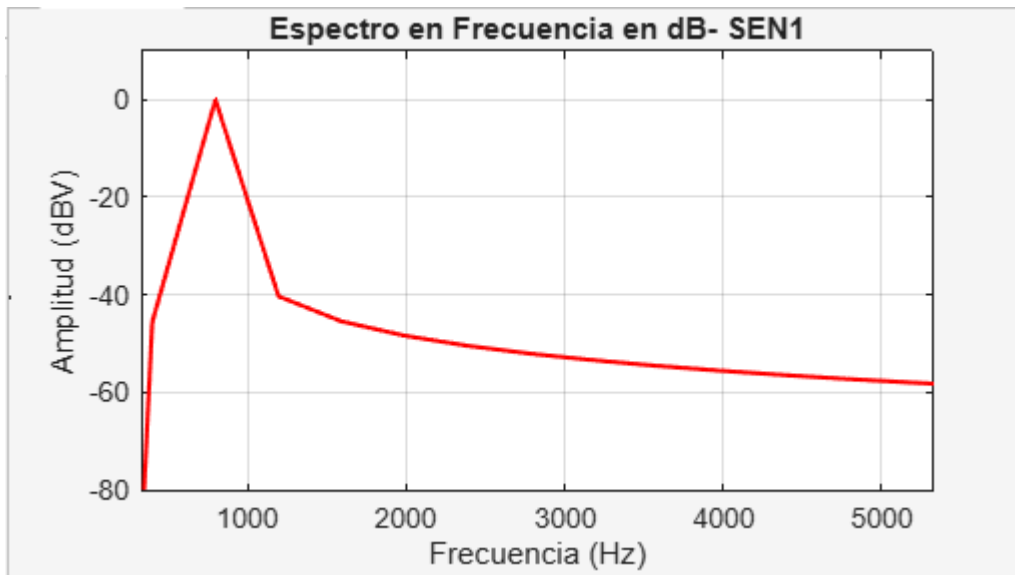
title('Espectro en Frecuencia en dB- SEN1');

grid on;

xlim([0 5000]);     % Rango de frecuencias (visualización)

ylim([-80 10]);     % Rango de amplitud (escala dBV)

```



### GRAFICA IFFT

% Parámetros de la señal SEN1

T = 0.00125; % Periodo de la señal (s)

f = 1/T; % Frecuencia de la señal (Hz) → 800 Hz

Fs = 1e5; % Frecuencia de muestreo (100 kHz)

t = 0:1/Fs:2\*T; % Tiempo total de 2 ciclos (0 a 0.0025 s)

A1 = 1; % Amplitud de la señal

% Señal original (onda seno)

v1 = A1 \* sin(2\*pi\*f\*t);

% FFT de la señal

N = length(v1); % Número de muestras

Y = fft(v1); % FFT de la señal

% Reconstrucción por IFFT

v1\_rec = ifft(Y, 'symmetric'); % Reconstruir señal aplicando IFFT

% Gráfica final de la señal reconstruida

t\_ms = t \* 1000; % Convertir a milisegundos para graficar

figure;

plot(t\_ms, v1\_rec, 'r', 'LineWidth', 1.5);

```

xlabel('Tiempo (ms)');
ylabel('Amplitud (V)');
title('SEÑAL RECONSTRUIDA CON IFFT (DOMINIO DEL TIEMPO)');
grid on;

```



## PHYTON

```

import pandas as pd

import matplotlib.pyplot as plt

# Ruta al archivo

archivo = 'DOMINIOTIEMPOSEN1.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés

tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar

```

```
plt.plot(tiempo, voltaje, color='red')

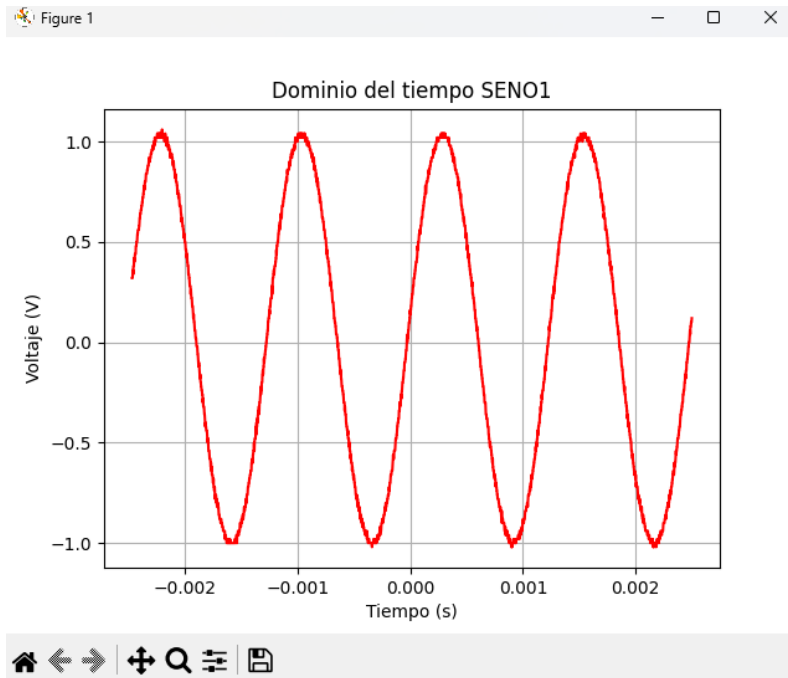
plt.title('Dominio del tiempo SEN01')

plt.xlabel('Tiempo (s)')

plt.ylabel('Voltaje (V)')

plt.grid(True)

plt.show()
```



## FFT

```
import pandas as pd

import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIASEN1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```



```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='red')
```

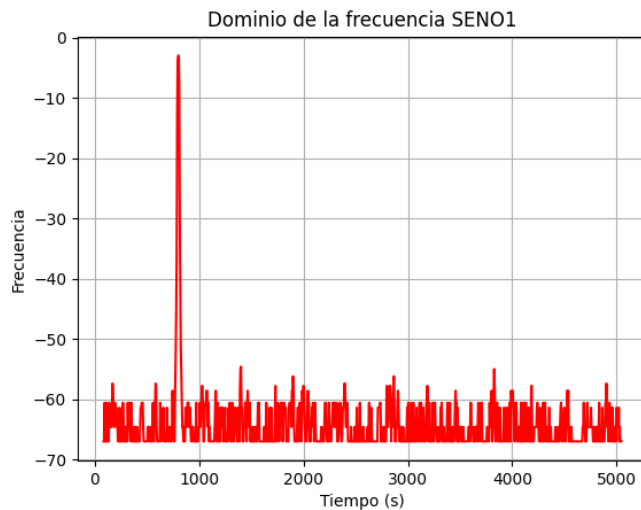
```
plt.title('Dominio de la frecuencia SENO1')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Frecuencia')
```

```
plt.grid(True)
```

```
plt.show()
```

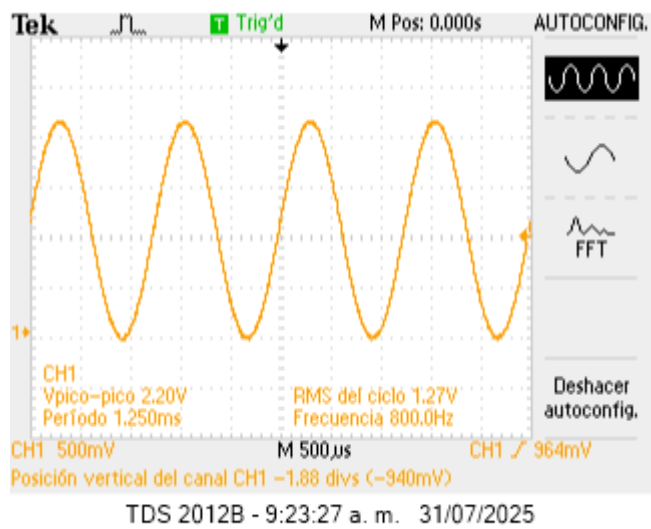


## SENO2

### TOMA DATOS EXPERIMENTALES

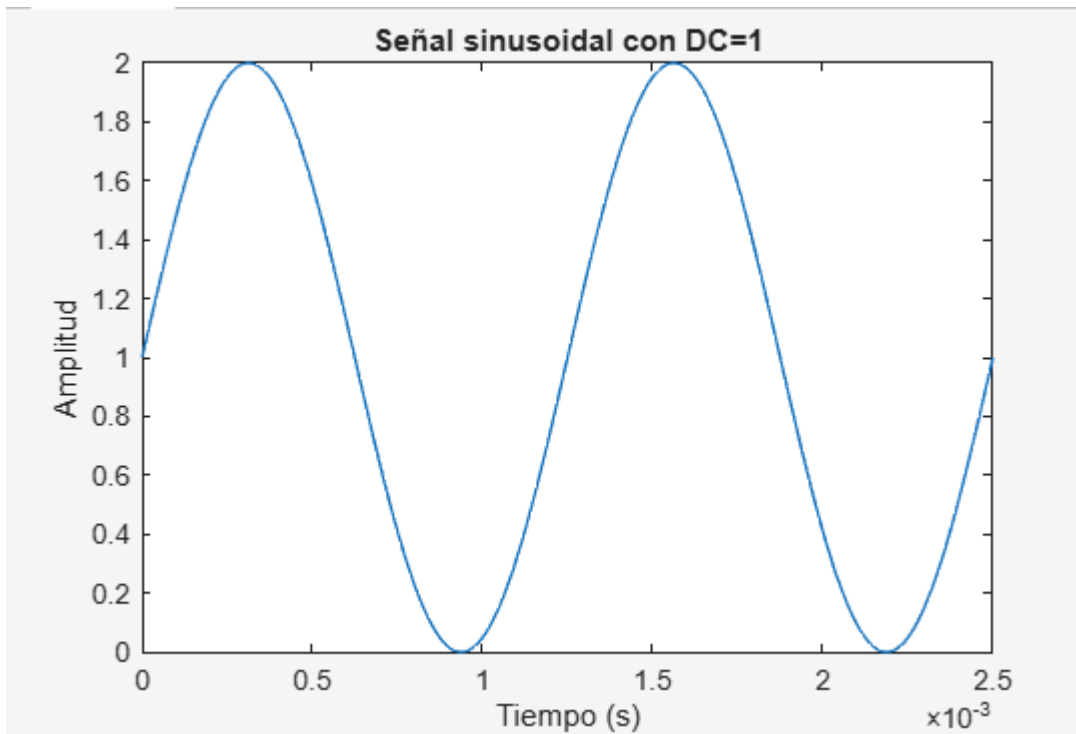
FRECUENCIA	800 Hz
Vp	1

SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A		
				d.B	Vrms	VP(v)
Sen 1	1	1	800	-2,19	0,77714132	1,09904379



## GRAFICA MATLAB CON CODIGO

```
T=0.00125;
f=1/T;
t=0:1/100000:2*T;
A1=1;
v1=1+A1*sin(2*pi*f*t);
plot(t,v1)
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal sinusoidal con DC=1');
```



### GRAFICA MATLAB CON DATOS EXPORTADO

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOTIEMPO2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOTIEMPO2.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

% Graficar

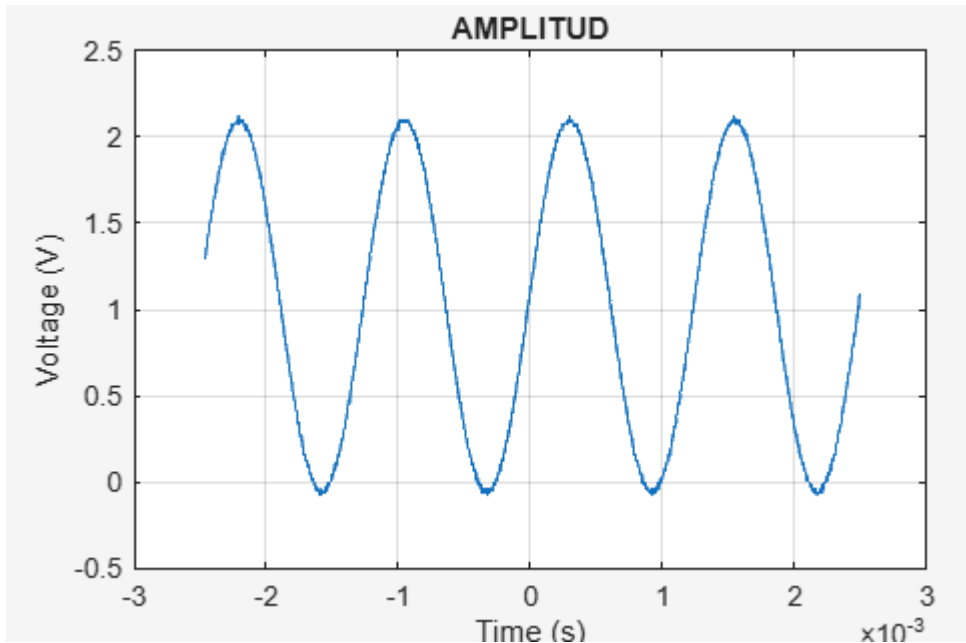
```
plot(t, v);
```

```
xlabel('Time (s)');
```

```
ylabel('Voltage (V)');
```

```
title('AMPLITUD');
```

grid on;



### DOMINIO FRECUENCIA

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOFRECUENCIAS2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOFRECUENCIAS2.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos(:,4); % Columna 4 = Tiempo
```

```
v = datos(:,5); % Columna 5 = Voltaje
```

% Graficar

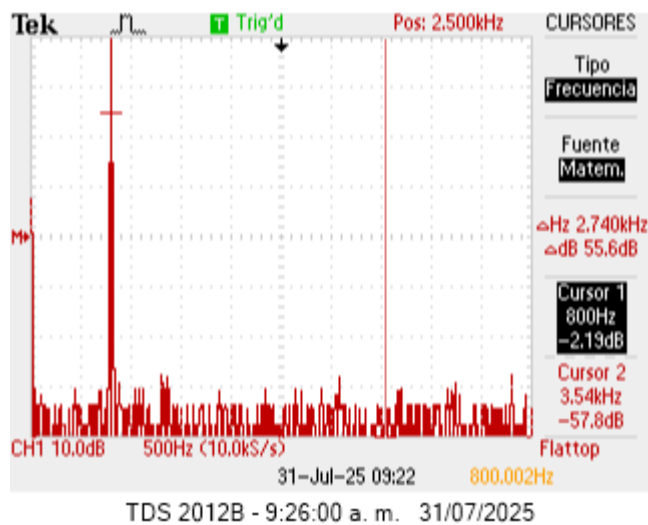
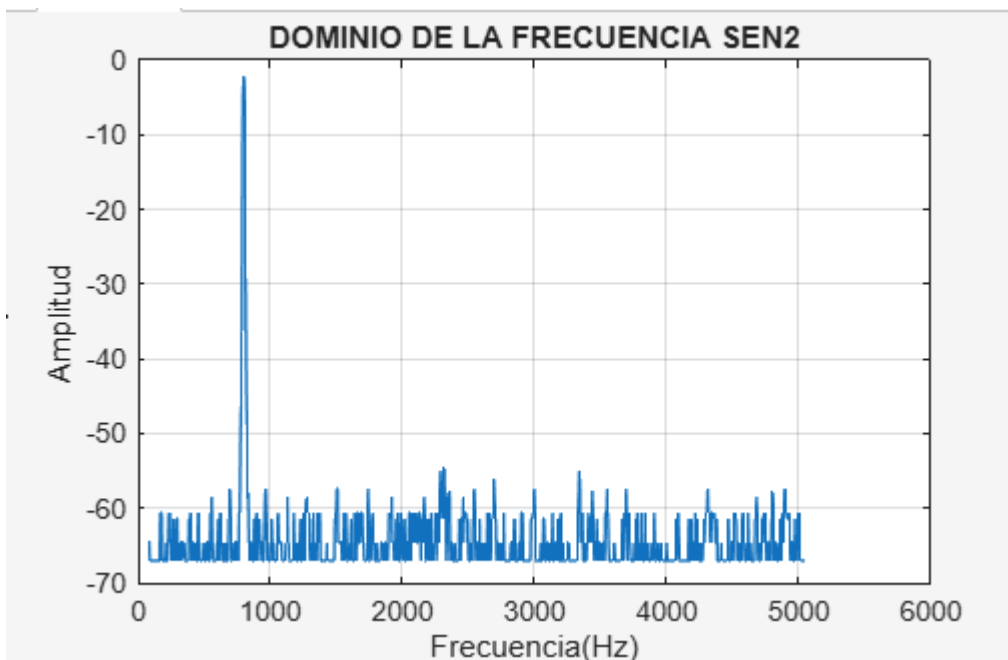
```
plot(t, v);
```

```
xlabel('Frecuencia(Hz)');
```

```
ylabel('Amplitud');
```

```
title('DOMINIO DE LA FRECUENCIA SEN2');
```

grid on;



### GRAFICA FFT (TEORICA)

% Señal SEN2

% Parámetros de la señal

T = 0.00125;      % Periodo (s)

f = 800;            % Frecuencia (Hz)

Fs = 1e5;           % Frecuencia de muestreo (100 kHz)

```

t = 0:1/Fs:2*T;    % Tiempo total de 2 ciclos

A1 = 1.09904379;    % Amplitud pico real desde la tabla (VP)

% Generar señal senoidal

v1 = A1 * sin(2*pi*f*t);

% Cálculo de la FFT

N = length(v1);    % Número total de muestras

Y = fft(v1);    % Transformada rápida de Fourier

Y_mag = abs(Y)/N;    % Magnitud normalizada

Y_mag(2:end-1) = 2*Y_mag(2:end-1); % Duplicar componentes (menos DC y Nyquist)

% Escala de frecuencia

f_axis = (0:N-1)*(Fs/N);    % Vector de frecuencias asociado a la FFT

% Convertir magnitud a decibelios

Y_dB = 20*log10(Y_mag + eps); % eps evita log(0) y errores numéricos

% Graficar el espectro (solo la mitad positiva)

plot(f_axis(1:N/2), Y_dB(1:N/2), 'g', 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dBV)');

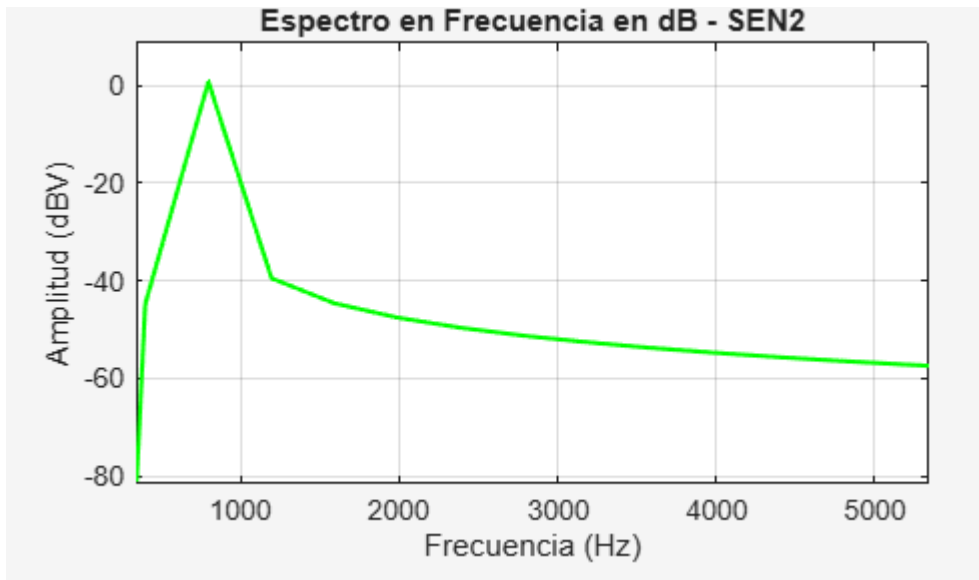
title('Espectro en Frecuencia en dB - SEN2');

grid on;

xlim([0 5000]);    % Rango de frecuencias (visualización)

ylim([-80 10]);    % Rango de amplitud (escala dBV)

```



### GRAFICA IFFT

% Señal SEN2

% === Parámetros de la señal ===

T = 0.00125; % Periodo (s)

f = 800; % Frecuencia (Hz)

Fs = 1e5; % Frecuencia de muestreo (100 kHz)

t = 0:1/Fs:2\*T; % Tiempo total de 2 ciclos

A1 = 1.09904379; % Amplitud pico real desde la tabla (VP)

% === Generar señal senoidal ===

v1 = A1 \* sin(2\*pi\*f\*t);

% === Cálculo de la FFT ===

N = length(v1); % Número total de muestras

Y = fft(v1); % Transformada rápida de Fourier

Y\_mag = abs(Y)/N; % Magnitud normalizada

Y\_mag(2:end-1) = 2\*Y\_mag(2:end-1); % Duplicar componentes (menos DC y Nyquist)

% === Escala de frecuencia ===

```
f_axis = (0:N-1)*(Fs/N);    % Vector de frecuencias asociado a la FFT
```

```
% === Convertir magnitud a decibelios ===
```

```
Y_dB = 20*log10(Y_mag + eps); % eps evita log(0) y errores numéricos
```

```
% === Gráfico del espectro de frecuencia (mitad positiva) ===
```

```
figure;
```

```
plot(f_axis(1:N/2), Y_dB(1:N/2), 'g', 'LineWidth', 1.5);
```

```
xlabel('Frecuencia (Hz)');
```

```
ylabel('Amplitud (dBV)');
```

```
title('Espectro en Frecuencia en dB - SEN2');
```

```
grid on;
```

```
xlim([0 5000]);
```

```
ylim([-80 10]);
```

```
% === RECONSTRUCCIÓN CON IFFT ===
```

```
v1_rec = ifft(Y, 'symmetric'); % IFFT simétrica
```

```
t_ms = t * 1000;          % Convertir tiempo a milisegundos
```

```
% === Gráfico de la señal reconstruida ===
```

```
figure;
```

```
plot(t_ms, v1_rec, 'g', 'LineWidth', 1.5);
```

```
xlabel('Tiempo (ms)');
```

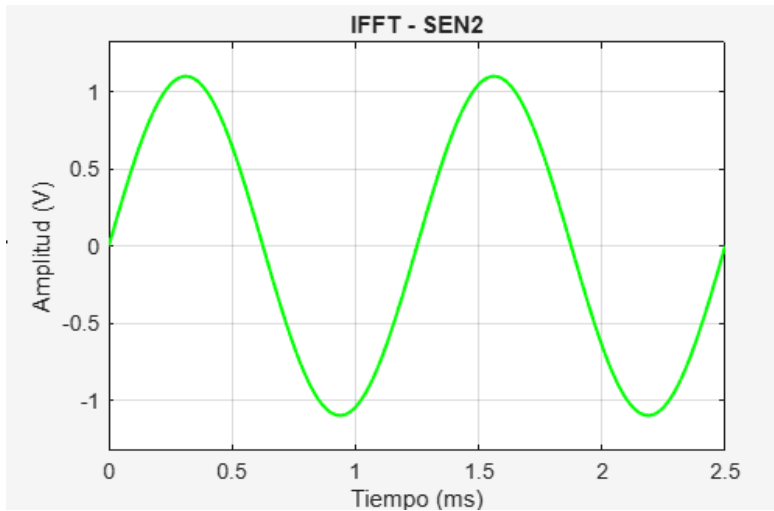
```
ylabel('Amplitud (V)');
```

```
title('IFFT - SEN2');
```

```
grid on;
```

```
ylim([-1.2*A1 1.2*A1]);    % Límite vertical ajustado a la amplitud real
```





## PHYTON

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOTIEMPO2.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#FFA500')
```

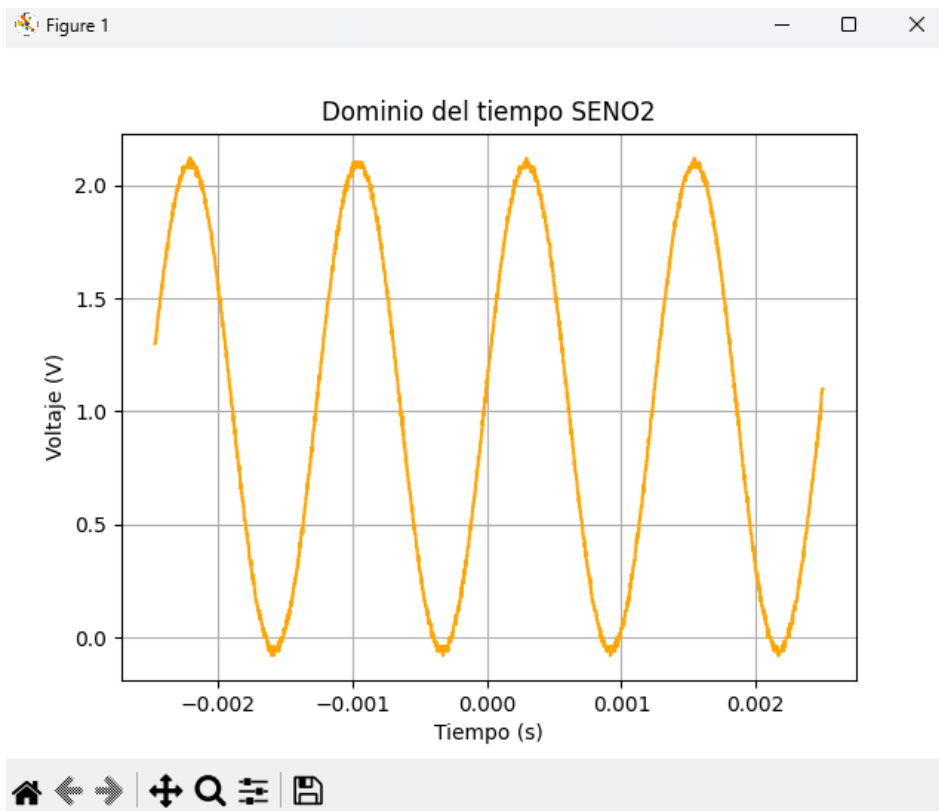
```
plt.title('Dominio del tiempo SENO2')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Voltaje (V)')
```

```
plt.grid(True)
```

```
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIAS2.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#FFA500')
```

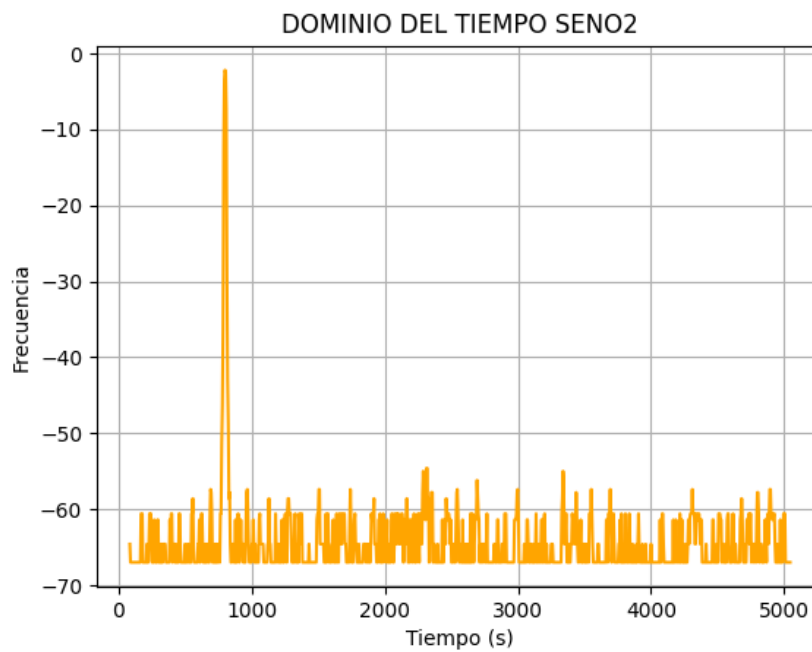
```
plt.title('DOMINIO DEL TIEMPO SENO2')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Frecuencia')
```

```
plt.grid(True)
```

```
plt.show()
```



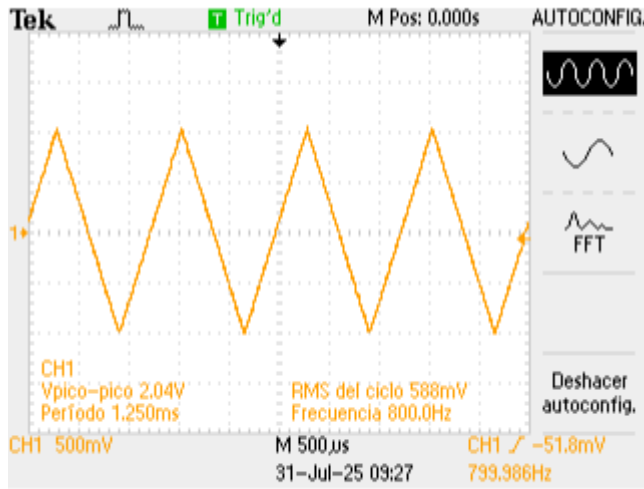
## TRIAN1

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz
Vp	1

SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A			
				d.B	Vrms	VP(v)	TEORICO
TRIANGULAR1	0	1	800	-4,59	0,5895	0,8337	0,81056947
		3	2400	-23,8	0,0646	0,0913	0,09006327
		5	4000	-32,6	0,0234	0,0332	0,03242278

		7	5600	-38,6	0,0117	0,0166	0,01654223
		9	7200	-43	0,0071	0,0100	0,01000703
		11	8800	-47	0,0045	0,0063	0,00669892



TDS 2012B - 9:30:22 a. m. 31/07/2025

## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000000:2\*T;

A1=0.810;

A3=0.090;

A5=0.032;

A7=0.017;

A9=0.010;

A11=0.0066;

v1=A1\*cos(2\*pi\*f\*t);

v2=A3\*cos(2\*pi\*3\*f\*t);

v3=A5\*cos(2\*pi\*5\*f\*t);

v4=A7\*cos(2\*pi\*7\*f\*t);

v5=A9\*cos(2\*pi\*9\*f\*t);

```
v6=A11*cos(2*pi*11*f*t);
```

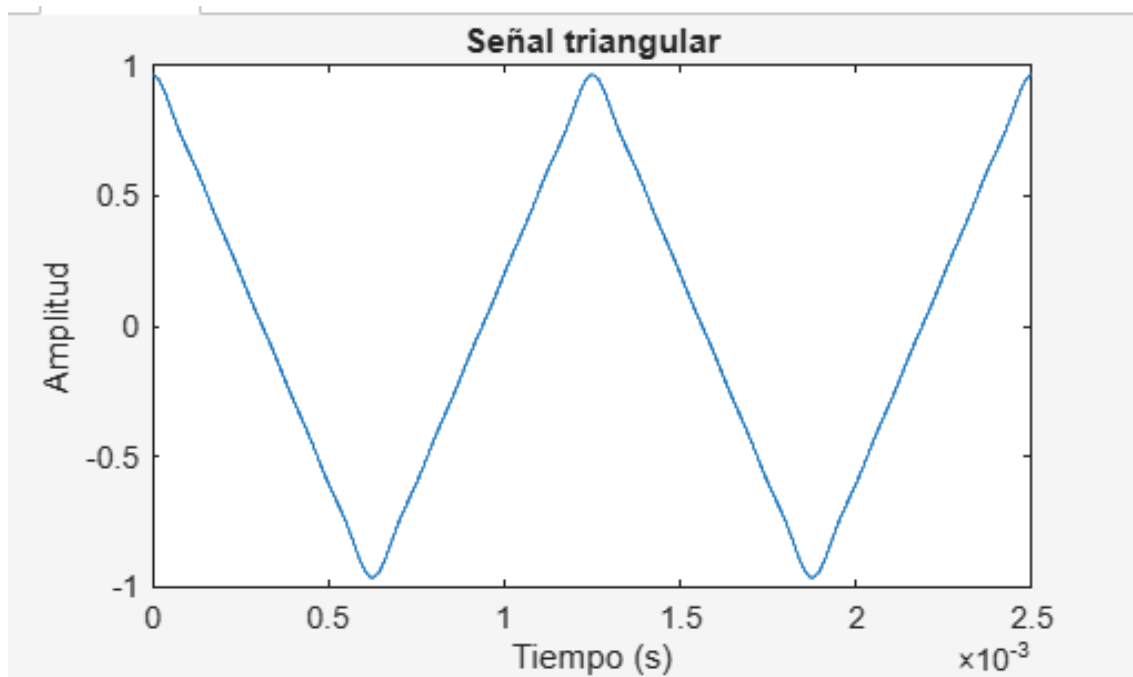
```
VT=v1+v2+v3+v4+v5+v6;
```

```
plot(t,VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal triangular');
```



#### GRAFICA CON MEDIDAS VP

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/1000000000:2*T;
```

```
A1=0.833;
```

```
A3=0.091;
```

```
A5=0.033;
```

```
A7=0.016;
```

```
A9=0.010;
```

```

A11=0.0063;

v1=A1*cos(2*pi*f*t);
v2=A3*cos(2*pi*3*f*t);
v3=A5*cos(2*pi*5*f*t);
v4=A7*cos(2*pi*7*f*t);
v5=A9*cos(2*pi*9*f*t);
v6=A11*cos(2*pi*11*f*t);

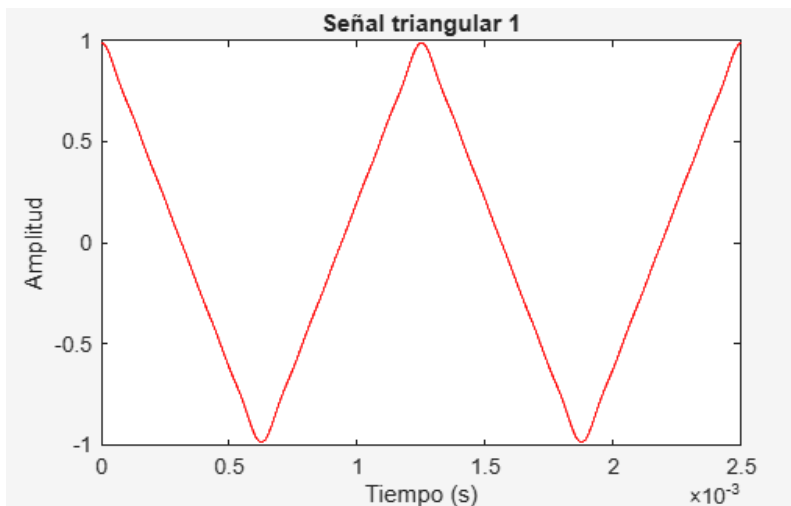
```

```

VT=v1+v2+v3+v4+v5+v6;

plot(t,VT, 'r');
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal triangular 1');

```



## GRAFICA MATLAB CON DATOS EXPORTADO

```

% Leer el archivo como tabla, empezando desde la fila 18

opts = detectImportOptions('DOMINIOTIEMPOTRI1.csv');

opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante

opts.Delimiter = ','; % Delimitador por coma

opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

```

```

% Leer los datos

datos = readtable('DOMINIOTIEMPOTRI1.csv', opts);

% Convertir las columnas correctas (4 y 5) a vectores

t = datos{:,4}; % Columna 4 = Tiempo

v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);

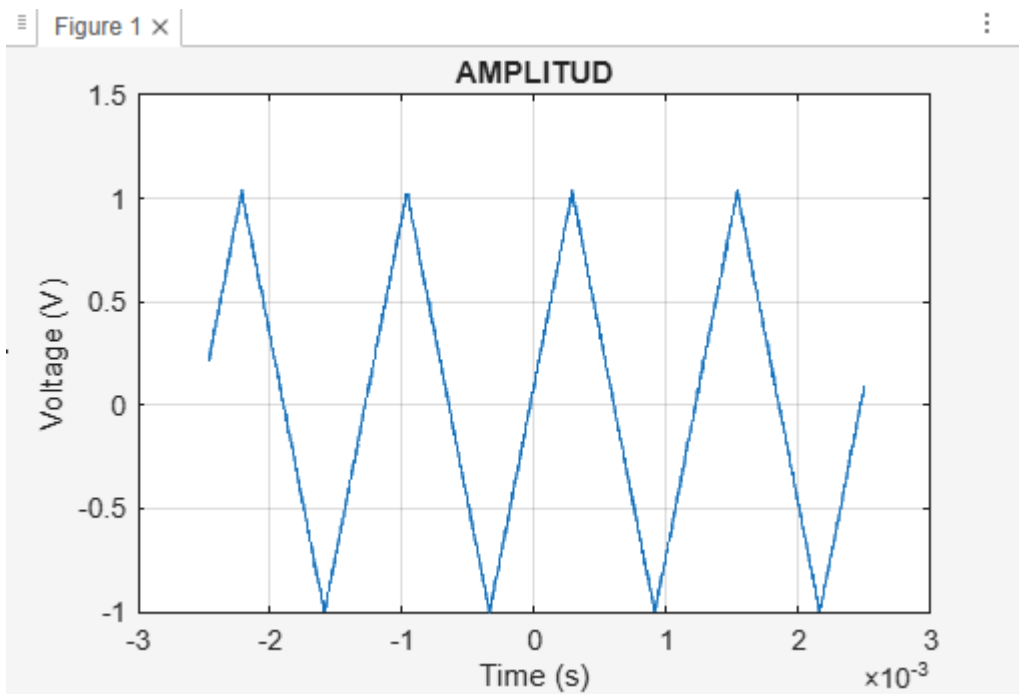
xlabel('Time (s)');

ylabel('Voltage (V)');

title('AMPLITUD');

grid on;

```



## DOMINIO FRECUENCIA

```

% Leer el archivo como tabla, empezando desde la fila 18

opts = detectImportOptions('DOMINIOFRECUENCIATRI1.csv');

opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante

opts.Delimiter = ','; % Delimitador por coma

opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

```

```

% Leer los datos

datos = readtable('DOMINIOFRECUENCIA TRI1.csv', opts);

% Convertir las columnas correctas (4 y 5) a vectores

t = datos{:,4}; % Columna 4 = Tiempo

v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);

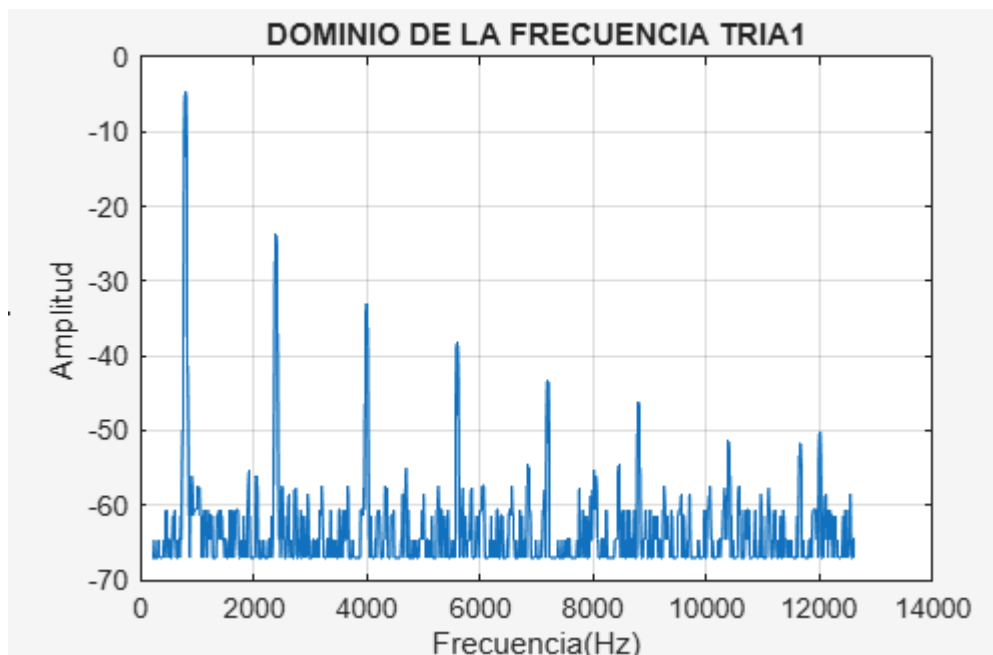
xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA TRIA1');

grid on;

```







```
v3 = A5*cos(2*pi*5*f*t);  
v4 = A7*cos(2*pi*7*f*t);  
v5 = A9*cos(2*pi*9*f*t);  
v6 = A11*cos(2*pi*11*f*t);  
VT = v1 + v2 + v3 + v4 + v5 + v6;
```

```
% === Gráfica en el dominio del tiempo ===
```

```
figure;  
plot(t, VT);  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
title('Señal Triangular en el tiempo');  
grid on;
```

```
% === FFT de la señal ===
```

```
N = length(VT);  
Y = fft(VT);  
Y_mag = abs(Y)/N;  
Y_mag(2:end-1) = 2*Y_mag(2:end-1); % Duplicar amplitudes excepto DC y Nyquist
```

```
% === Cálculo en decibelios ===
```

```
Y_dB = 20*log10(Y_mag + eps);
```

```
% === Eje de frecuencias ===
```

```
f_axis = (0:N-1)*(Fs/N);
```

```
% === Gráfica en el dominio de la frecuencia (dB) ===
```

```
figure;  
plot(f_axis(1:N/2), Y_dB(1:N/2), 'b', 'LineWidth', 1.5);
```

```

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

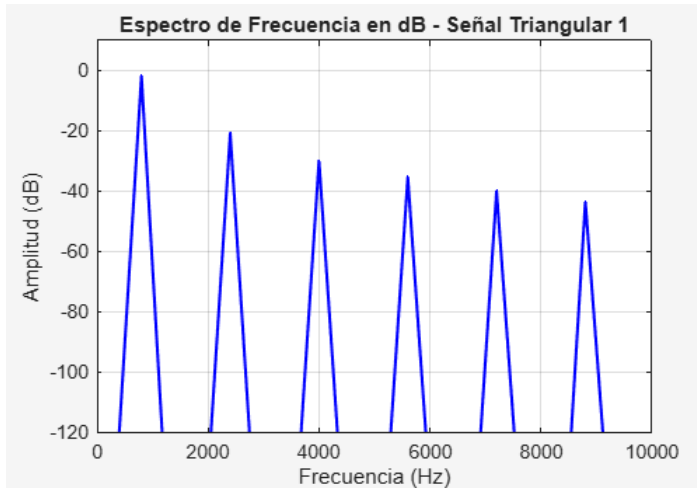
title('Espectro de Frecuencia en dB - Señal Triangular');

grid on;

xlim([0 10000]); % Visualizar hasta 10 kHz

ylim([-120 10]); % Rango típico en dB

```



## GRAFICA IFFT

```

% === Parámetros de la señal ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz) = 800

Fs = 1e9; % Frecuencia de muestreo (1 GHz)

t = 0:1/Fs:2*T; % Tiempo para 2 ciclos

% === Componentes armónicas ===

A1 = 0.810;

A3 = 0.090;

A5 = 0.032;

A7 = 0.017;

A9 = 0.010;

```

```
A11 = 0.0066;
```

```
% === Construcción de la señal compuesta ===
```

```
v1 = A1*cos(2*pi*f*t);
```

```
v2 = A3*cos(2*pi*3*f*t);
```

```
v3 = A5*cos(2*pi*5*f*t);
```

```
v4 = A7*cos(2*pi*7*f*t);
```

```
v5 = A9*cos(2*pi*9*f*t);
```

```
v6 = A11*cos(2*pi*11*f*t);
```

```
VT = v1 + v2 + v3 + v4 + v5 + v6;
```

```
% === Gráfica en el dominio del tiempo ===
```

```
figure;
```

```
plot(t, VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal Triangular en el tiempo');
```

```
grid on;
```

```
% === FFT de la señal ===
```

```
N = length(VT);
```

```
Y = fft(VT);
```

```
Y_mag = abs(Y)/N;
```

```
Y_mag(2:end-1) = 2*Y_mag(2:end-1); % Duplicar amplitudes excepto DC y Nyquist
```

```
% === Cálculo en decibelios ===
```

```
Y_dB = 20*log10(Y_mag + eps);
```

```
% === Eje de frecuencias ===
```

```
f_axis = (0:N-1)*(Fs/N);
```

```
% === Gráfica en el dominio de la frecuencia (dB) ===
```

```
figure;
```

```
plot(f_axis(1:N/2), Y_dB(1:N/2), 'b', 'LineWidth', 1.5);
```

```
xlabel('Frecuencia (Hz)');
```

```
ylabel('Amplitud (dB)');
```

```
title('Espectro de Frecuencia en dB - Señal Triangular');
```

```
grid on;
```

```
xlim([0 10000]); % Visualizar hasta 10 kHz
```

```
ylim([-120 10]); % Rango típico en dB
```

```
% === Reconstrucción con IFFT ===
```

```
VT_rec = ifft(Y, 'symmetric');
```

```
% === Gráfica de la señal reconstruida con IFFT ===
```

```
figure;
```

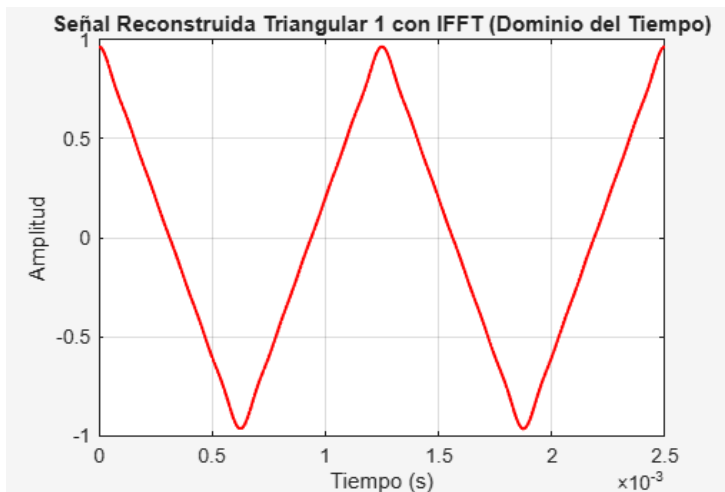
```
plot(t, VT_rec, 'r', 'LineWidth', 1.5);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal Reconstruida Triangular 1 con IFFT (Dominio del Tiempo)');
```

```
grid on;
```



## PHYTON

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOTIEMPOTRI1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#40E0D0')
```

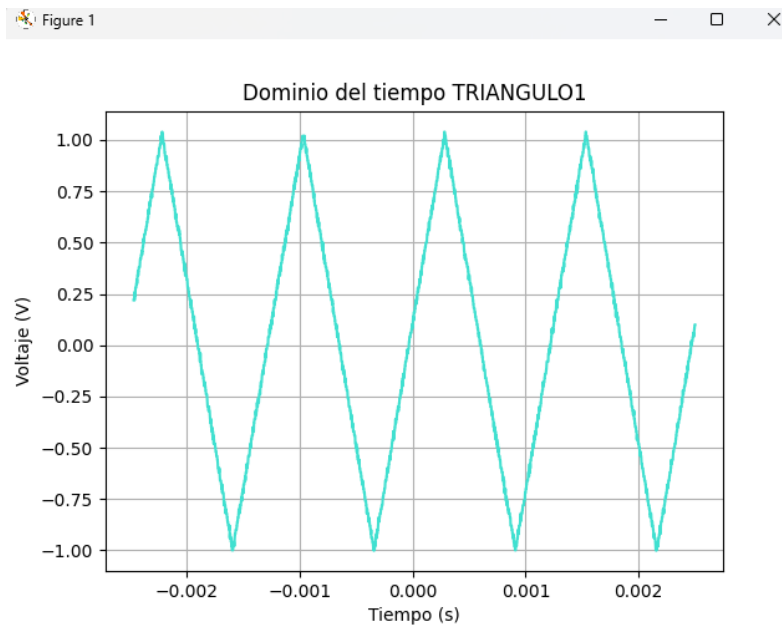
```
plt.title('Dominio del tiempo TRIANGULO1')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Voltaje (V)')
```

```
plt.grid(True)
```

```
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIATRI1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#40E0D0')
```

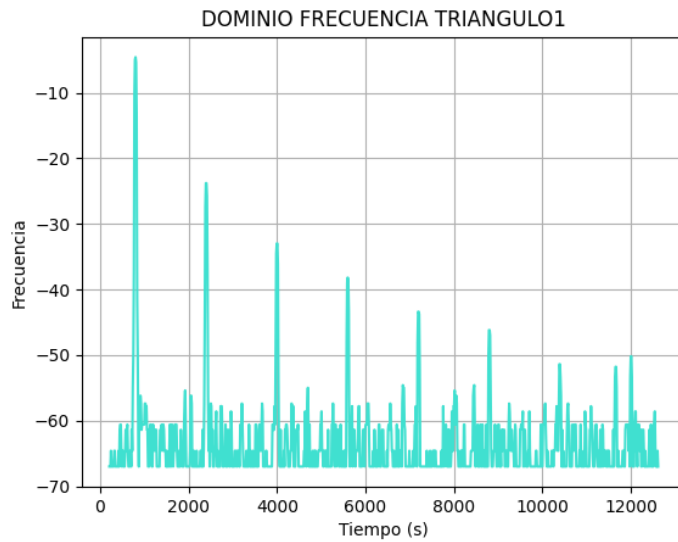
```
plt.title('DOMINIO FRECUENCIA TRIANGULO1')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Frecuencia')
```

```
plt.grid(True)
```

```
plt.show()
```

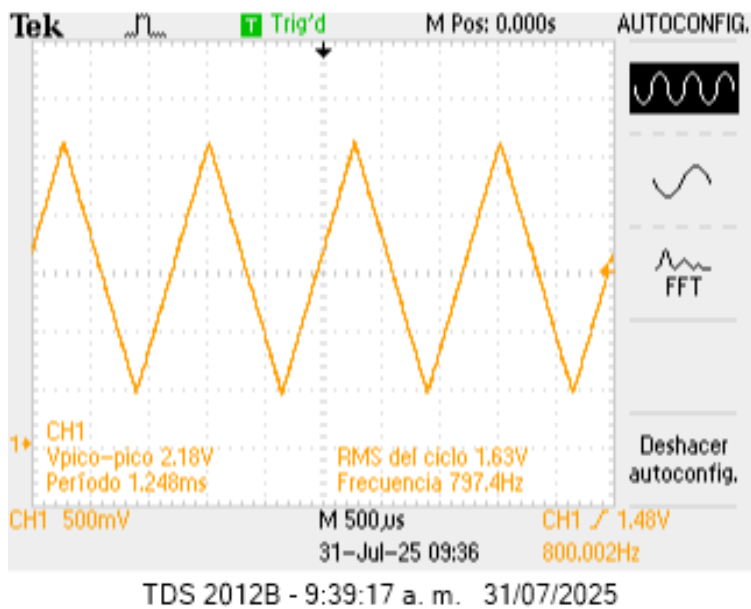


## TRIA2

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz						
Vp	1						
SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A			
				d.B	Vrms	VP(v)	TEORICO
TRIANGULAR2	1,5	1	800	-4,19	0,6173	0,8730	0,8106
		3	2400	-23,4	0,0676	0,0956	0,0901
		5	4000	-32,2	0,0245	0,0347	0,0324
		7	5600	-38,6	0,0117	0,0166	0,0165
		9	7200	-43,4	0,0068	0,0096	0,0100
		11	8800	-			
				45,08	0,0056	0,0079	0,0067





## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000:2\*T;

A1=0.810;

A3=0.090;

A5=0.032;

A7=0.017;

A9=0.010;

A11=0.0066;

v1=A1\*cos(2\*pi\*f\*t);

v2=A3\*cos(2\*pi\*3\*f\*t);

v3=A5\*cos(2\*pi\*5\*f\*t);

v4=A7\*cos(2\*pi\*7\*f\*t);

v5=A9\*cos(2\*pi\*9\*f\*t);

v6=A11\*cos(2\*pi\*11\*f\*t);

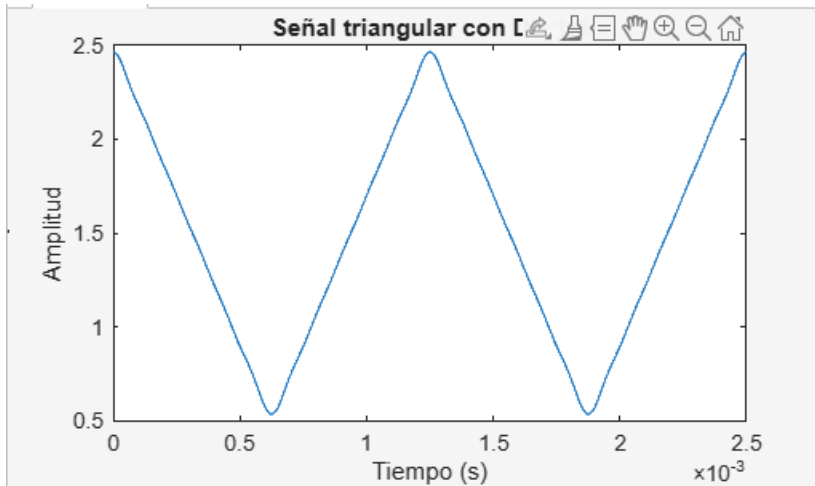
```
VT=1.5+(v1+v2+v3+v4+v5+v6);
```

```
plot(t,VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal triangular con DC 1.5');
```



### GRAFICA CON MEDIDAS VP

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/1000000:2*T;
```

```
A1=0.873;
```

```
A3=0.095;
```

```
A5=0.034;
```

```
A7=0.016;
```

```
A9=0.009;
```

```
A11=0.0079;
```

```
v1=A1*cos(2*pi*f*t);
```

```
v2=A3*cos(2*pi*3*f*t);
```

```
v3=A5*cos(2*pi*5*f*t);
```

```
v4=A7*cos(2*pi*7*f*t);
```

```
v5=A9*cos(2*pi*9*f*t);
```

```
v6=A11*cos(2*pi*11*f*t);
```

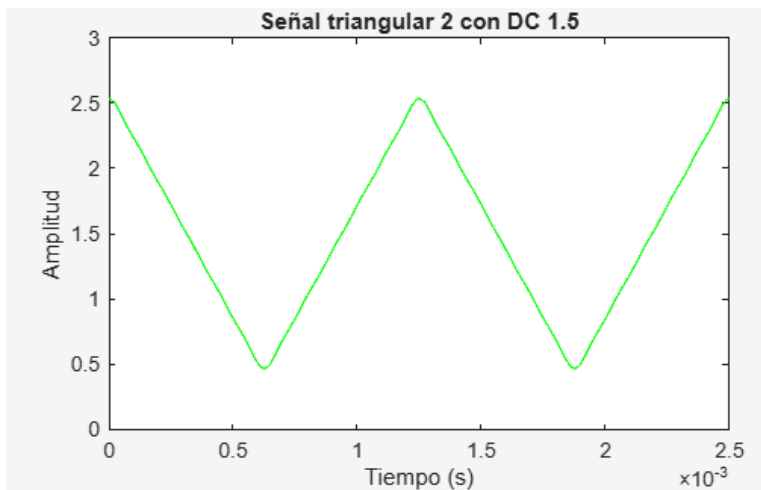
```
VT=1.5+(v1+v2+v3+v4+v5+v6);
```

```
plot(t,VT, 'g');
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal triangular 2 con DC 1.5');
```



## GRAFICA MATLAB CON DATOS EXPORTADO

```
% Leer el archivo como tabla, empezando desde la fila 18
```

```
opts = detectImportOptions('DOMINIOTIEMPOTRI2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

```
% Leer los datos
```

```
datos = readtable('DOMINIOTIEMPOTRI2.csv', opts);
```

```
% Convertir las columnas correctas (4 y 5) a vectores
```

```
t = datos(:,4); % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

```
% Graficar
```

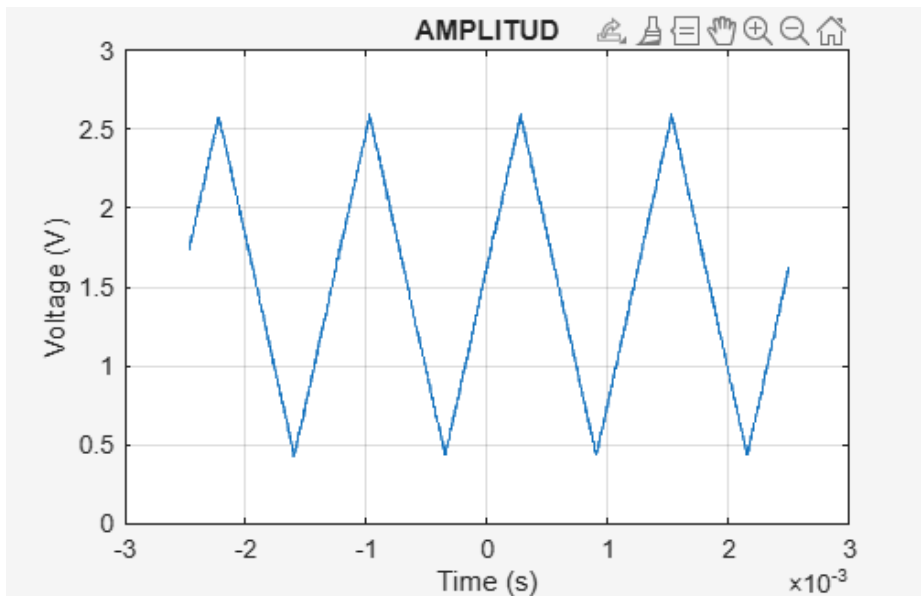
```
plot(t, v);
```

```
xlabel('Time (s)');
```

```
ylabel('Voltage (V)');
```

```
title('AMPLITUD');
```

```
grid on;
```



## DOMINIO FRECUENCIA

```
% Leer el archivo como tabla, empezando desde la fila 18
```

```
opts = detectImportOptions('DOMINIOFRECUENBCIATRI2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

```
% Leer los datos
```

```
datos = readtable('DOMINIOFRECUENBCIATRI2.csv', opts);
```

```
% Convertir las columnas correctas (4 y 5) a vectores
```

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

```
% Graficar
```

```

plot(t, v);

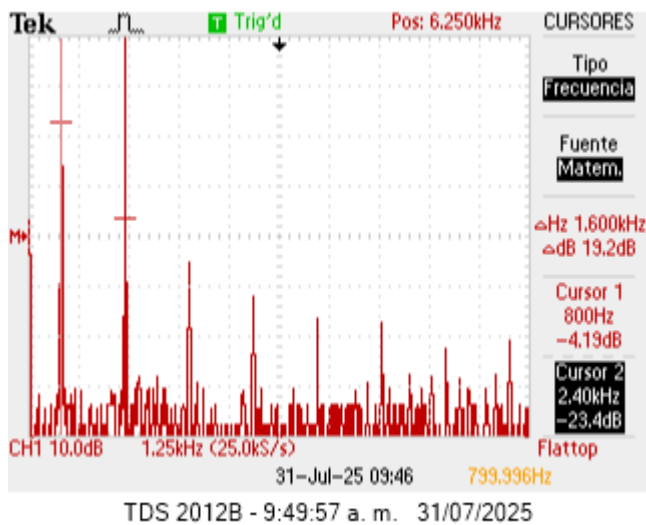
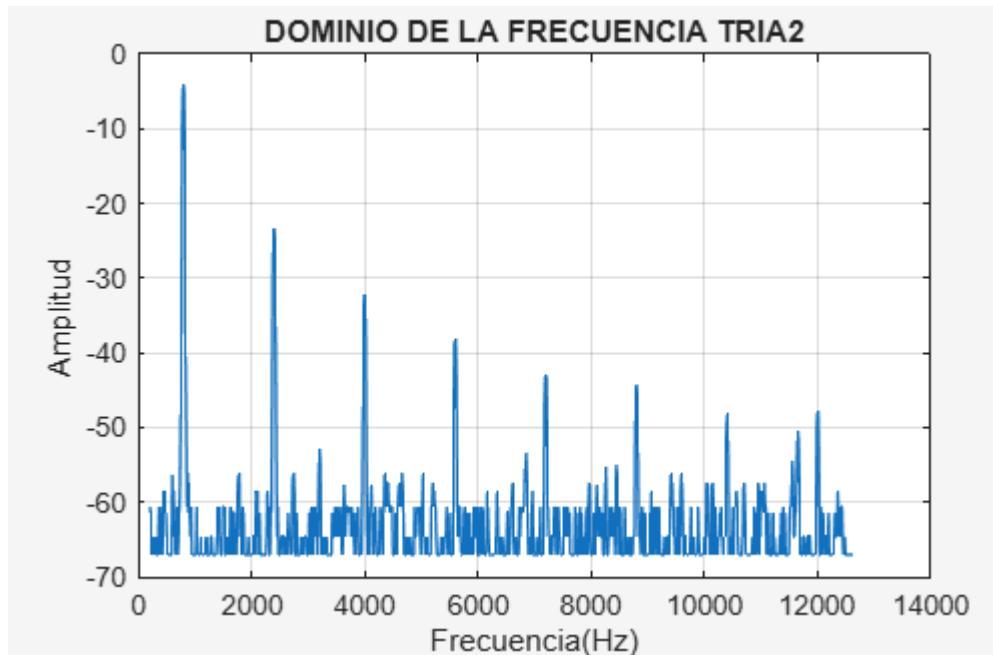
xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA TRIA2');

grid on;

```



**GRAFICA FFT (TEORICA)**

```
% Señal TRIANGULAR2
```

```
% === Parámetros de la señal ===
```

```
T = 0.00125;      % Periodo (s)
```

```
f = 1/T;          % Frecuencia fundamental (Hz) = 800
```

```
Fs = 1e6;         % Frecuencia de muestreo (1 MHz)
```

```
t = 0:1/Fs:2*T;    % Tiempo para 2 ciclos
```

```
% === Componentes armónicas ===
```

```
A1 = 0.810;
```

```
A3 = 0.090;
```

```
A5 = 0.032;
```

```
A7 = 0.017;
```

```
A9 = 0.010;
```

```
A11 = 0.0066;
```

```
% === Construcción de la señal con componente DC ===
```

```
v1 = A1*cos(2*pi*f*t);
```

```
v2 = A3*cos(2*pi*3*f*t);
```

```
v3 = A5*cos(2*pi*5*f*t);
```

```
v4 = A7*cos(2*pi*7*f*t);
```

```
v5 = A9*cos(2*pi*9*f*t);
```

```
v6 = A11*cos(2*pi*11*f*t);
```

```
VT = 1.5 + v1 + v2 + v3 + v4 + v5 + v6;
```

```
% === Gráfica en el dominio del tiempo ===
```

```
figure;
```

```
plot(t, VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```

title('Señal Triangular con DC = 1.5 V');

grid on;

% === FFT de la señal ===

N = length(VT);

Y = fft(VT);

Y_mag = abs(Y)/N;

Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

% === Cálculo en decibelios ===

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===

f_axis = (0:N-1)*(Fs/N);

% === Gráfica en el dominio de la frecuencia (dB) ===

figure;

plot(f_axis(1:N/2), Y_dB(1:N/2), 'r', 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

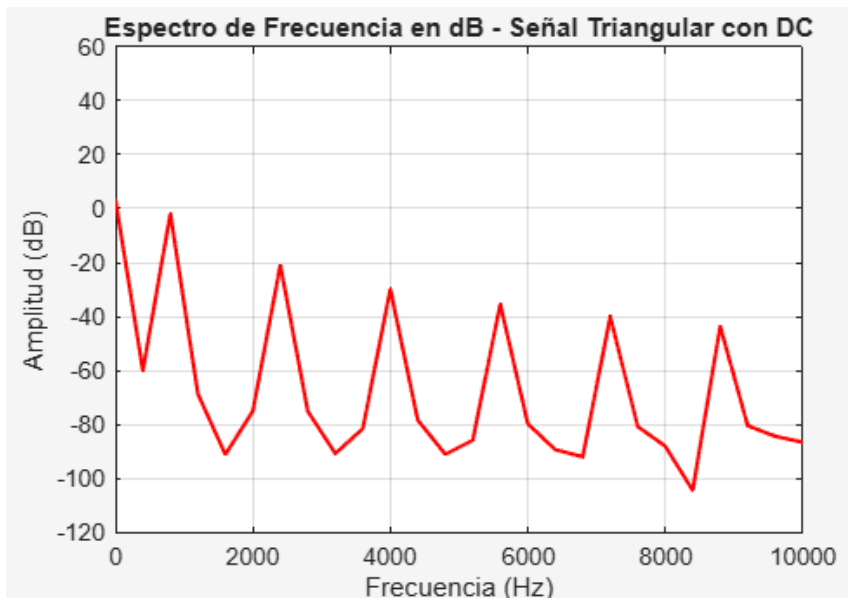
title('Espectro de Frecuencia en dB - Señal Triangular con DC');

grid on;

xlim([0 10000]); % Limitar visualización hasta 10 kHz

ylim([-120 60]); % Rango común en espectros dB

```



## GRAFICA IFFT

% Señal TRIANGULAR2

% === Parámetros de la señal ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz) = 800

Fs = 1e6; % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T; % Tiempo para 2 ciclos

% === Componentes armónicas ===

A1 = 0.810;

A3 = 0.090;

A5 = 0.032;

A7 = 0.017;

A9 = 0.010;

A11 = 0.0066;

% === Construcción de la señal con componente DC ===

v1 = A1\*cos(2\*pi\*f\*t);



```

v2 = A3*cos(2*pi*3*f*t);
v3 = A5*cos(2*pi*5*f*t);
v4 = A7*cos(2*pi*7*f*t);
v5 = A9*cos(2*pi*9*f*t);
v6 = A11*cos(2*pi*11*f*t);
VT = 1.5 + v1 + v2 + v3 + v4 + v5 + v6;

```

```

% === Gráfica en el dominio del tiempo (original) ===

```

```

figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal Triangular con DC = 1.5 V');
grid on;

```

```

% === FFT de la señal ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

```

```

% === Cálculo en decibelios ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```

% === Eje de frecuencias ===

```

```

f_axis = (0:N-1)*(Fs/N);

```

```

% === Gráfica en el dominio de la frecuencia (dB) ===

```

```

figure;

```

```

plot(f_axis(1:N/2), Y_dB(1:N/2), 'r', 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

title('Espectro de Frecuencia en dB - Señal Triangular con DC');

grid on;

xlim([0 10000]); % Limitar visualización hasta 10 kHz

ylim([-120 60]); % Rango común en espectros dB

% === Reconstrucción con IFFT ===

VT_rec = ifft(Y, 'symmetric');

% === Gráfica de la señal reconstruida (en rojo) ===

figure;

plot(t, VT_rec, 'r', 'LineWidth', 1.5);

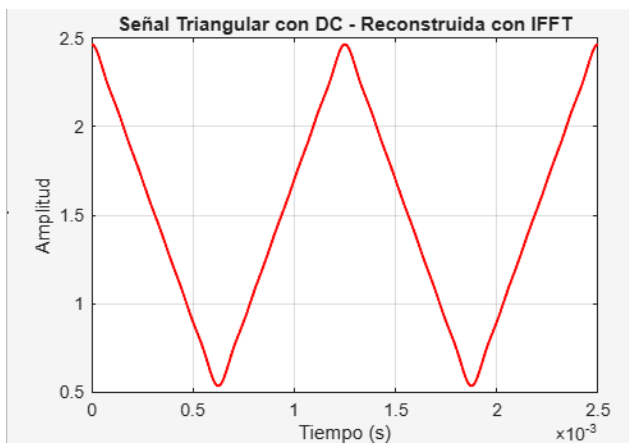
xlabel('Tiempo (s)');

ylabel('Amplitud');

title('Señal Triangular con DC - Reconstruida con IFFT');

grid on;

```



**PHYTON**

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOTIEMPOTRI2.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='b')
```

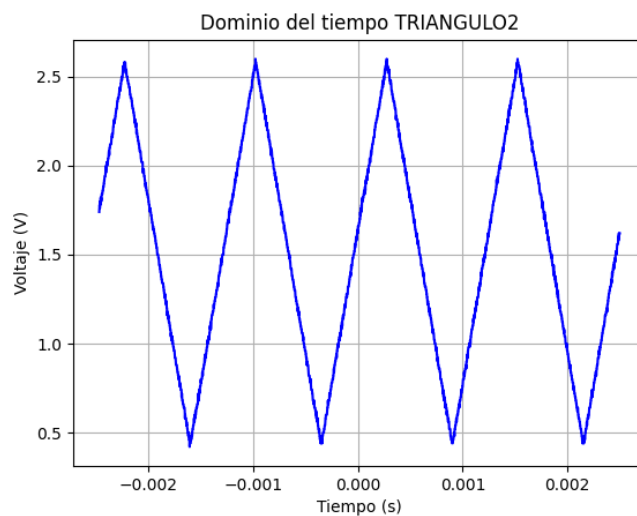
```
plt.title('Dominio del tiempo TRIANGULO2')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Voltaje (V)')
```

```
plt.grid(True)
```

```
plt.show()
```



## **FRECUENCIA**

```
import pandas as pd

import matplotlib.pyplot as plt

# Ruta al archivo

archivo = 'DOMINIOFRECUENBCIATRI2.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1

datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés

tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)

voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar

plt.plot(tiempo, voltaje, color='b')

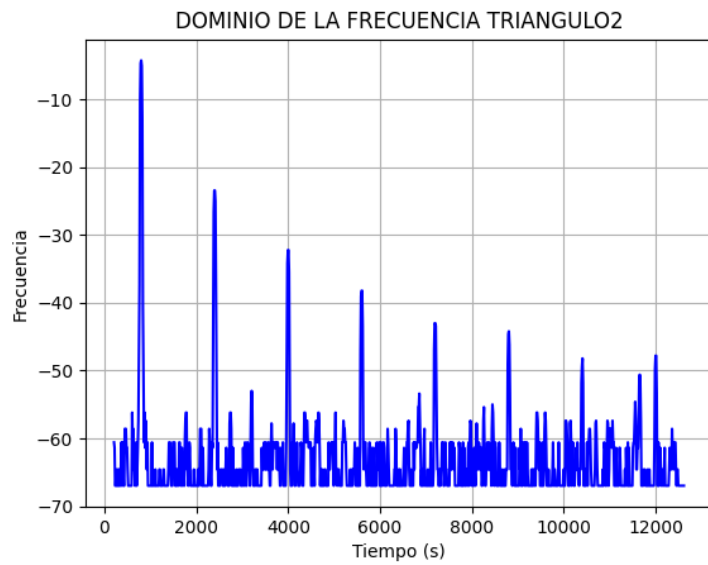
plt.title('DOMINIO DE LA FRECUENCIA TRIANGULO2')

plt.xlabel('Tiempo (s)')

plt.ylabel('Frecuencia')

plt.grid(True)

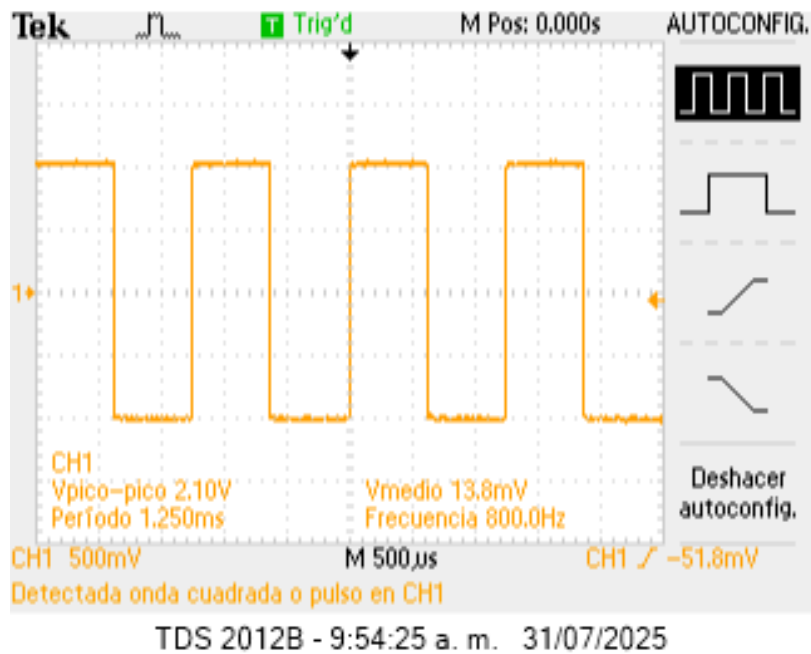
plt.show()
```



## CUAD1

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz						
Vp	1						
SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A			
				d.B	Vrms	VP(v)	TEORICO
CUADRADA1	0	1	800	-			
				0,999	0,8914	1,2606	1,2732
		3	2400	-10,2	0,3090	0,4370	0,4244
		5	4000	-15	0,1778	0,2515	0,2546
		7	5600	-17,8	0,1288	0,1822	0,1819
		9	7200	-19,8	0,1023	0,1447	0,1415
		11	8800	-21,8	0,0813	0,1150	0,1157



## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000:2\*T;

A1=1.273;

A3=0.424;

A5=0.254;

A7=0.1819;

A9=0.1415;

A11=0.1157;

v1=A1\*sin(2\*pi\*f\*t);

v2=A3\*sin(2\*pi\*3\*f\*t);

v3=A5\*sin(2\*pi\*5\*f\*t);

v4=A7\*sin(2\*pi\*7\*f\*t);

v5=A9\*sin(2\*pi\*9\*f\*t);

```
v6=A11*sin(2*pi*11*f*t);
```

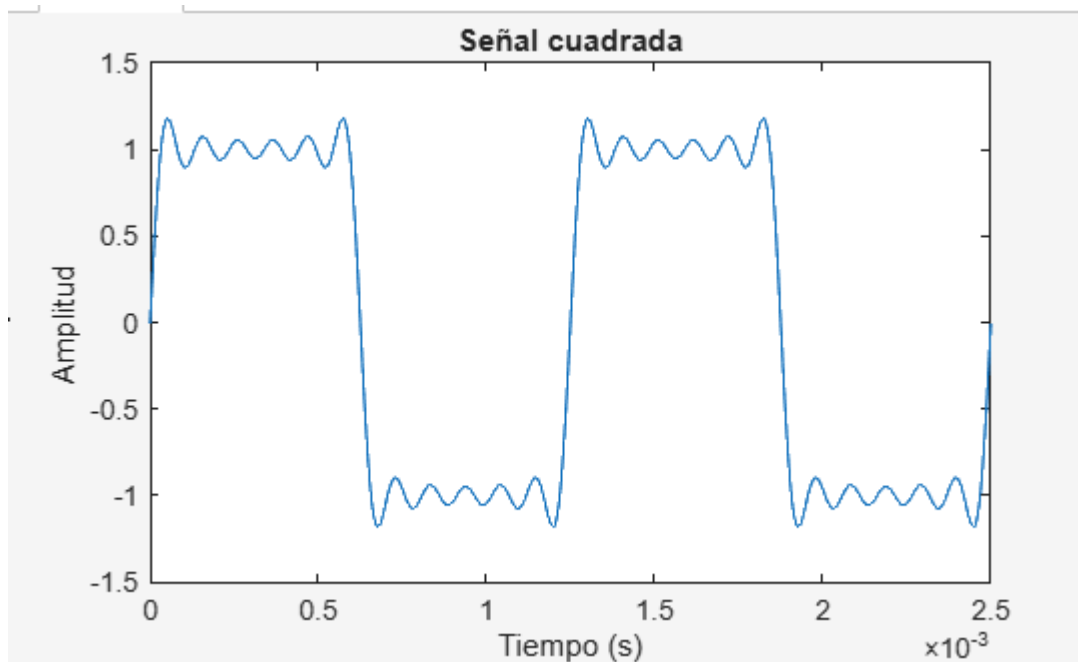
```
VT=(v1+v2+v3+v4+v5+v6);
```

```
plot(t,VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal cuadrada');
```



### MEDIDA CON VP

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/1000000:2*T;
```

```
A1=1.260;
```

```
A3=0.437;
```

```
A5=0.251;
```

```
A7=0.1822;
```

```
A9=0.1447;
```

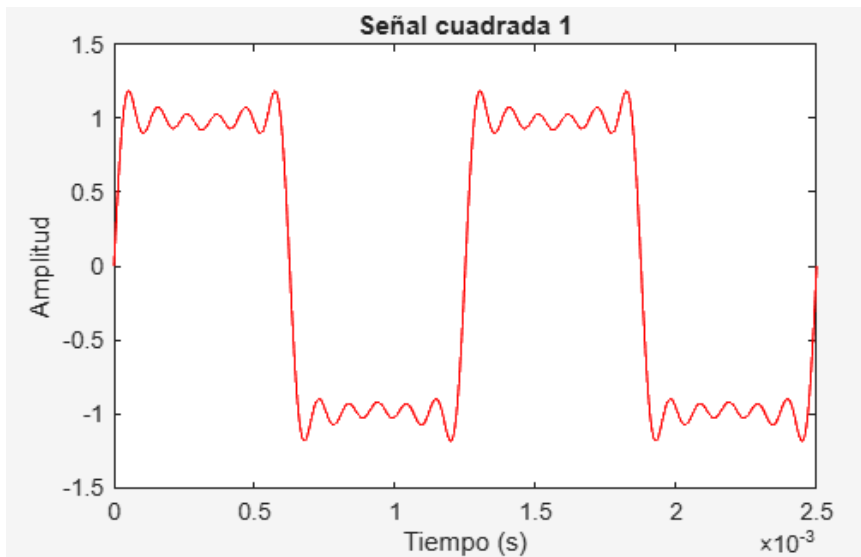
```
A11=0.1150;
```

```

v1=A1*sin(2*pi*f*t);
v2=A3*sin(2*pi*3*f*t);
v3=A5*sin(2*pi*5*f*t);
v4=A7*sin(2*pi*7*f*t);
v5=A9*sin(2*pi*9*f*t);
v6=A11*sin(2*pi*11*f*t);

VT=(v1+v2+v3+v4+v5+v6);
plot(t,VT, 'r');
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal cuadrada 1');

```



### GRAFICA MATLAB CON DATOS EXPORTADO

```

% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOTIEMPOCUA1.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos

```



```

datos = readtable('DOMINIOTIEMPOCUA1.csv', opts);

% Convertir las columnas correctas (4 y 5) a vectores

t = datos{:,4}; % Columna 4 = Tiempo

v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);

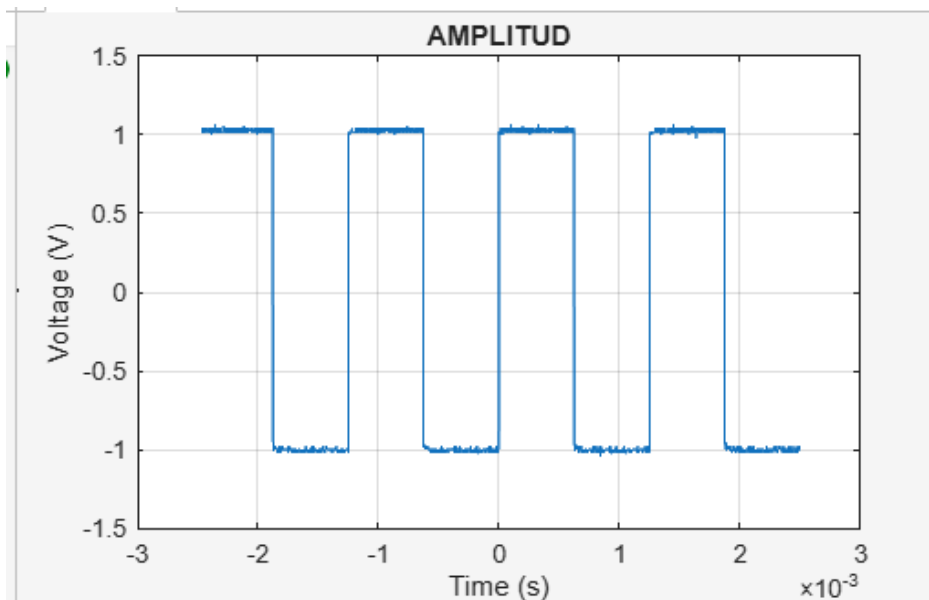
xlabel('Time (s)');

ylabel('Voltage (V)');

title('AMPLITUD');

grid on;

```



## DOMINIO FRECUENCIA

```

% Leer el archivo como tabla, empezando desde la fila 18

opts = detectImportOptions('DOMINIOFRECUENCIACUA1.csv');

opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante

opts.Delimiter = ','; % Delimitador por coma

opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

% Leer los datos

datos = readtable('DOMINIOFRECUENCIACUA1.csv', opts);

```

```
% Convertir las columnas correctas (4 y 5) a vectores
```

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

```
% Graficar
```

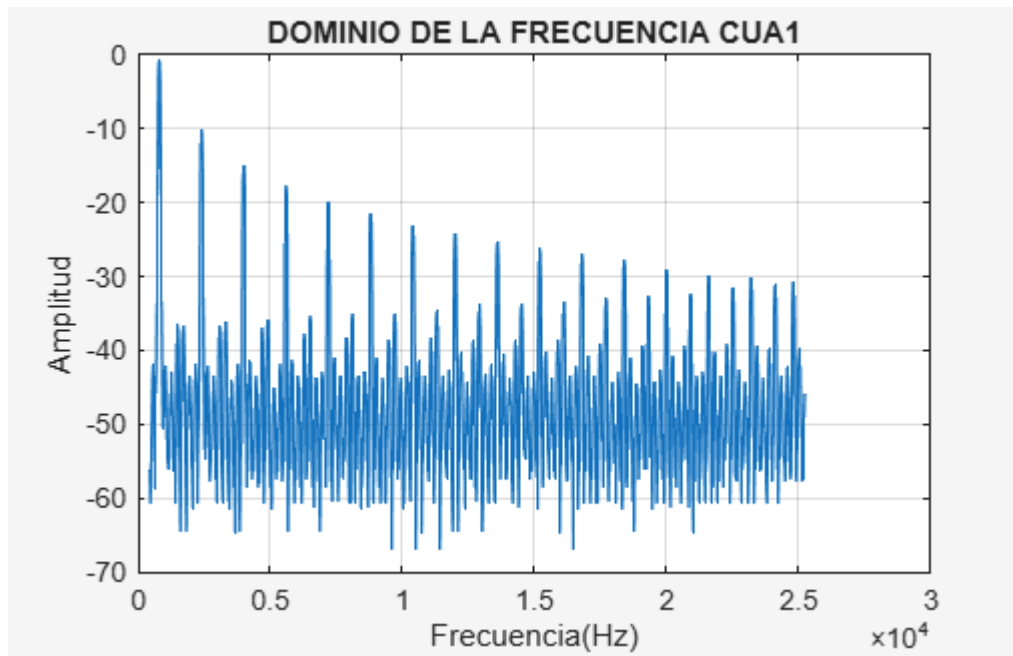
```
plot(t, v);
```

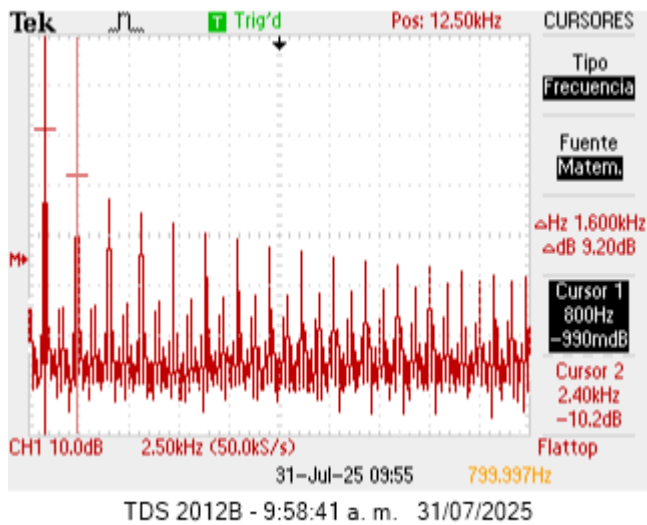
```
xlabel('Frecuencia(Hz)');
```

```
ylabel('Amplitud');
```

```
title('DOMINIO DE LA FRECUENCIA CUA1');
```

```
grid on;
```





### GRAFICA FFT (TEORICA)

% === Señal CUADRADA1 ===

T = 0.00125;            % Periodo (s)

f = 1/T;                % Frecuencia fundamental (Hz)

Fs = 1e6;              % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T;        % Tiempo de 2 ciclos

% === Componentes armónicas ===

A1 = 1.273;

A3 = 0.424;

A5 = 0.254;

A7 = 0.1819;

A9 = 0.1415;

A11 = 0.1157;

% === Construcción de la señal cuadrada ===

v1 = A1\*sin(2\*pi\*f\*t);

v2 = A3\*sin(2\*pi\*3\*f\*t);

```
v3 = A5*sin(2*pi*5*f*t);  
v4 = A7*sin(2*pi*7*f*t);  
v5 = A9*sin(2*pi*9*f*t);  
v6 = A11*sin(2*pi*11*f*t);  
VT = v1 + v2 + v3 + v4 + v5 + v6;
```

```
% === Gráfica de la señal en el tiempo ===
```

```
figure;  
plot(t, VT);  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
title('Señal CUADRADA1 en el dominio del tiempo');  
grid on;
```

```
% === FFT de la señal ===
```

```
N = length(VT);  
Y = fft(VT);  
Y_mag = abs(Y)/N;  
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)
```

```
% === Conversión a decibelios ===
```

```
Y_dB = 20 * log10(Y_mag + eps); % Agregar eps para evitar log(0)
```

```
% === Eje de frecuencias ===
```

```
f_axis = (0:N-1)*(Fs/N);
```

```
% === Gráfica del espectro en dB ===
```

```
figure;  
plot(f_axis(1:N/2), Y_dB(1:N/2), 'm', 'LineWidth', 1.5);
```

```

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

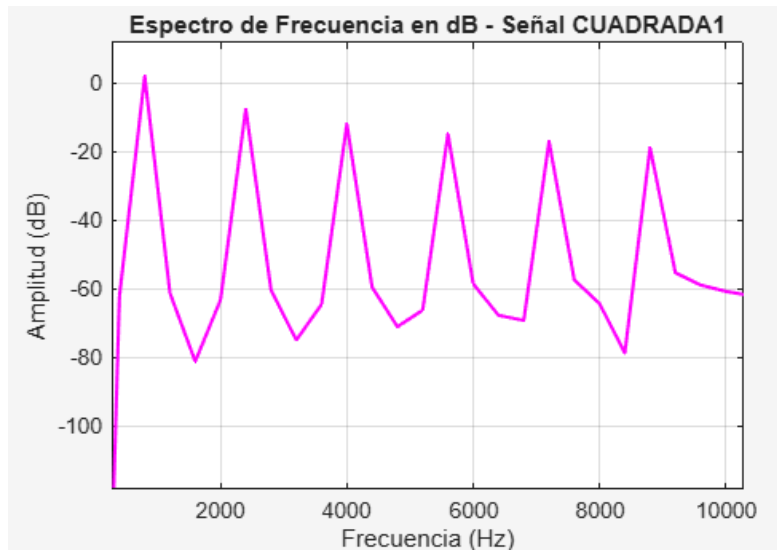
title('Espectro de Frecuencia en dB - Señal CUADRADA1');

grid on;

xlim([0 10000]); % Hasta 10 kHz para mejor visualización

ylim([-120 10]); % Rango de dB típico

```



### GRAFICA IFFT

```

% === Señal CUADRADA1 ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz)

Fs = 1e6; % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2*T; % Tiempo de 2 ciclos

% === Componentes armónicas ===

A1 = 1.273;

A3 = 0.424;

A5 = 0.254;

A7 = 0.1819;

```

```
A9 = 0.1415;
```

```
A11 = 0.1157;
```

```
% === Construcción de la señal cuadrada ===
```

```
v1 = A1*sin(2*pi*f*t);
```

```
v2 = A3*sin(2*pi*3*f*t);
```

```
v3 = A5*sin(2*pi*5*f*t);
```

```
v4 = A7*sin(2*pi*7*f*t);
```

```
v5 = A9*sin(2*pi*9*f*t);
```

```
v6 = A11*sin(2*pi*11*f*t);
```

```
VT = v1 + v2 + v3 + v4 + v5 + v6;
```

```
% === Gráfica de la señal en el tiempo ===
```

```
figure;
```

```
plot(t, VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal CUADRADA1 en el dominio del tiempo');
```

```
grid on;
```

```
% === FFT de la señal ===
```

```
N = length(VT);
```

```
Y = fft(VT);
```

```
Y_mag = abs(Y)/N;
```

```
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)
```

```
% === Conversión a decibelios ===
```

```
Y_dB = 20 * log10(Y_mag + eps); % Agregar eps para evitar log(0)
```

```
% === Eje de frecuencias ===
```

```
f_axis = (0:N-1)*(Fs/N);
```

```
% === Gráfica del espectro en dB ===
```

```
figure;
```

```
plot(f_axis(1:N/2), Y_dB(1:N/2), 'm', 'LineWidth', 1.5);
```

```
xlabel('Frecuencia (Hz)');
```

```
ylabel('Amplitud (dB)');
```

```
title('Espectro de Frecuencia en dB - Señal CUADRADA1');
```

```
grid on;
```

```
xlim([0 10000]);
```

```
ylim([-120 10]);
```

```
% === Reconstrucción con IFFT ===
```

```
VT_rec = ifft(Y, 'symmetric'); % Reconstrucción garantizando señal real
```

```
% === Gráfica de la señal reconstruida con IFFT ===
```

```
figure;
```

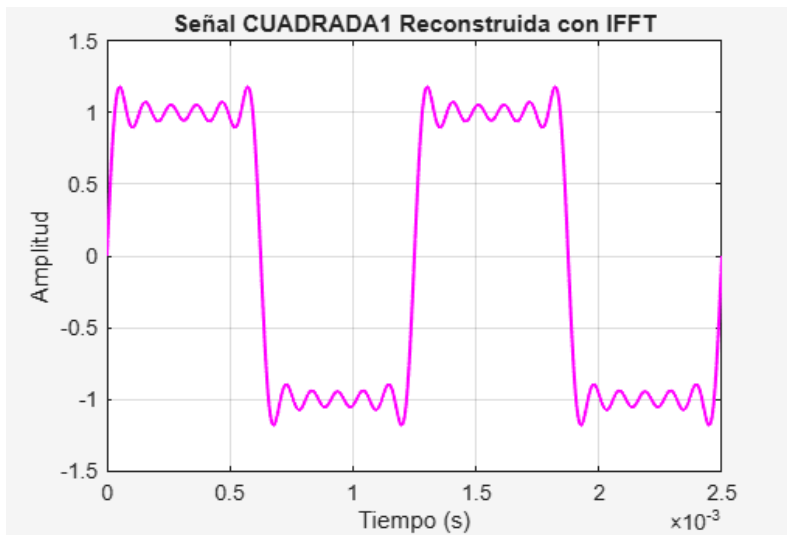
```
plot(t, VT_rec, 'm', 'LineWidth', 1.5);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal CUADRADA1 Reconstruida con IFFT');
```

```
grid on;
```



## PHYTON

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOTIEMPOCUA1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#333333')
```

```
plt.title('Dominio del tiempo CUADRADO1')
```

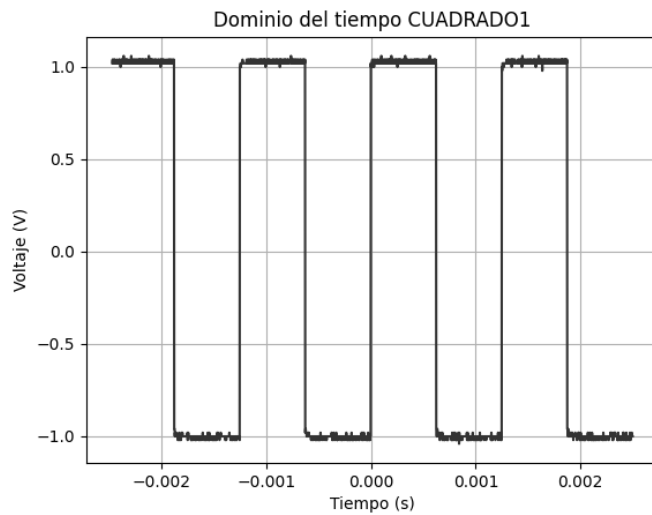
```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Voltaje (V)')
```

```
plt.grid(True)
```



```
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIACUA1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#333333')
```

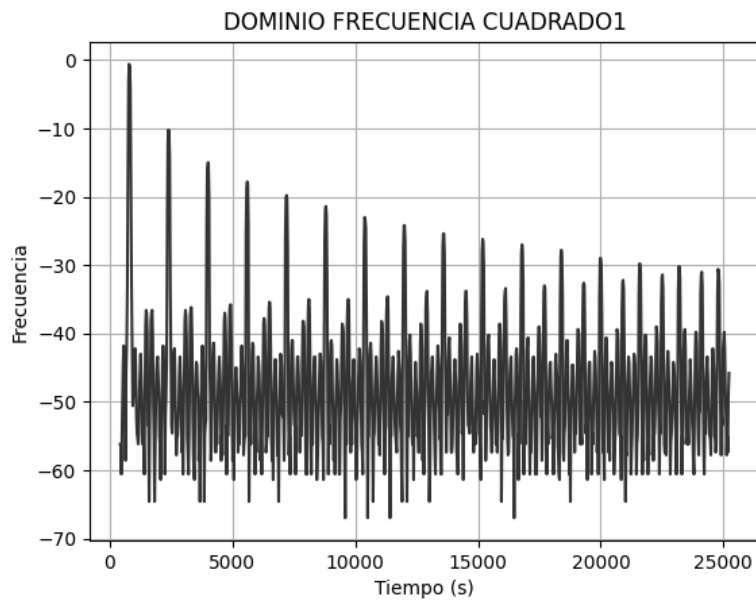
```
plt.title('DOMINIO FRECUENCIA CUADRADO1')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Frecuencia')
```

```
plt.grid(True)
```

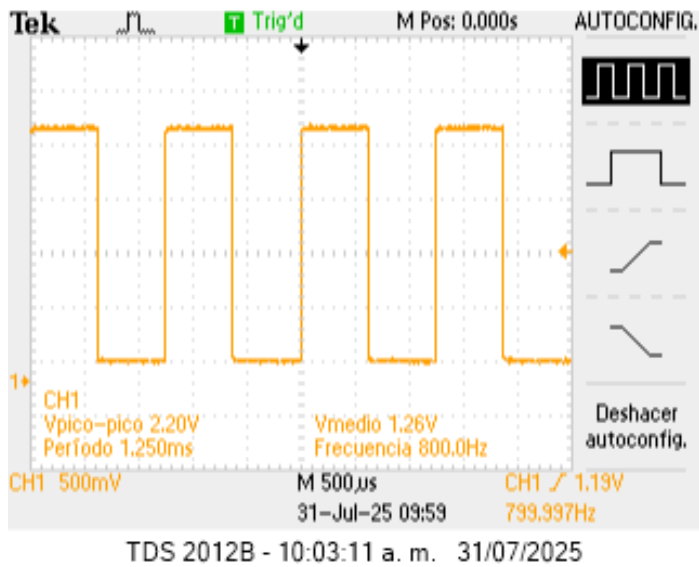
```
plt.show()
```



## CUAD2

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz						
Vp	1						
SEÑAL	DC	ARMONICA	FRECUENCIA(HZ)	A			
				d.B	Vrms	VP(v) exp	TEORICO
CUADRADA2	1,25	1	800	- 0,19	0,9784	1,3836	1,2732
		3	2400	- 9,79	0,3240	0,4582	0,4244
		5	4000	- 14,6	0,1862	0,2633	0,2546
		7	5600	- 17,4	0,1349	0,1908	0,1819
		9	7200	- 19,4	0,1072	0,1515	0,1415
		11	8800	-21	0,0891	0,1260	0,1157



## GRAFICA MATLAB CON CODIGO

```
%CUADRADA2
```

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/1000000:2*T;
```

```
A1=1.273;
```

```
A3=0.424;
```

```
A5=0.254;
```

```
A7=0.1819;
```

```
A9=0.1415;
```

```
A11=0.1157;
```

```
v1=A1*sin(2*pi*f*t);
```

```
v2=A3*sin(2*pi*3*f*t);
```

```
v3=A5*sin(2*pi*5*f*t);
```

```
v4=A7*sin(2*pi*7*f*t);
```

```
v5=A9*sin(2*pi*9*f*t);
```

```
v6=A11*sin(2*pi*11*f*t);
```

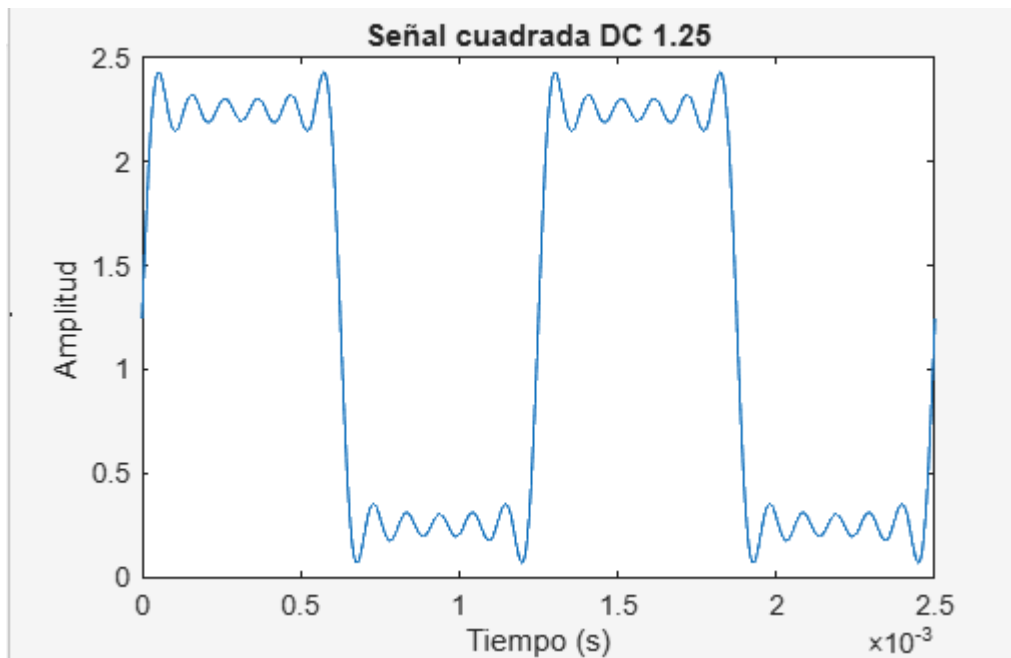
```
VT=1.25+(v1+v2+v3+v4+v5+v6);
```

```
plot(t,VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal cuadrada DC 1.25');
```



### MEDIDOS VP

```
T=0.00125;
```

```
f=1/T;
```

```
t=0:1/1000000:2*T;
```

```
A1=1.383;
```

```
A3=0.458;
```

```
A5=0.263;
```

```
A7=0.1908;
```

```
A9=0.1515;
```

```
A11=0.1260;
```

```
v1=A1*sin(2*pi*f*t);
```

```
v2=A3*sin(2*pi*3*f*t);
```

```

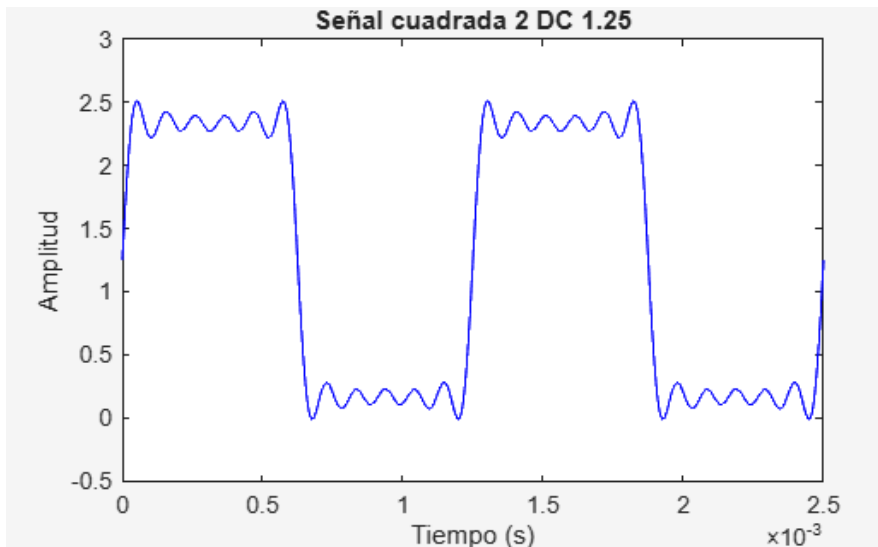
v3=A5*sin(2*pi*5*f*t);
v4=A7*sin(2*pi*7*f*t);
v5=A9*sin(2*pi*9*f*t);
v6=A11*sin(2*pi*11*f*t);

```

```

VT=1.25+(v1+v2+v3+v4+v5+v6);
plot(t,VT,'b');
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal cuadrada 2 DC 1.25');

```



#### GRAFICA MATLAB CON DATOS EXPORTADO

```

% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOTIEMPOCUA2.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos
datos = readtable('DOMINIOTIEMPOCUA2.csv', opts);
% Convertir las columnas correctas (4 y 5) a vectores

```

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

```
% Graficar
```

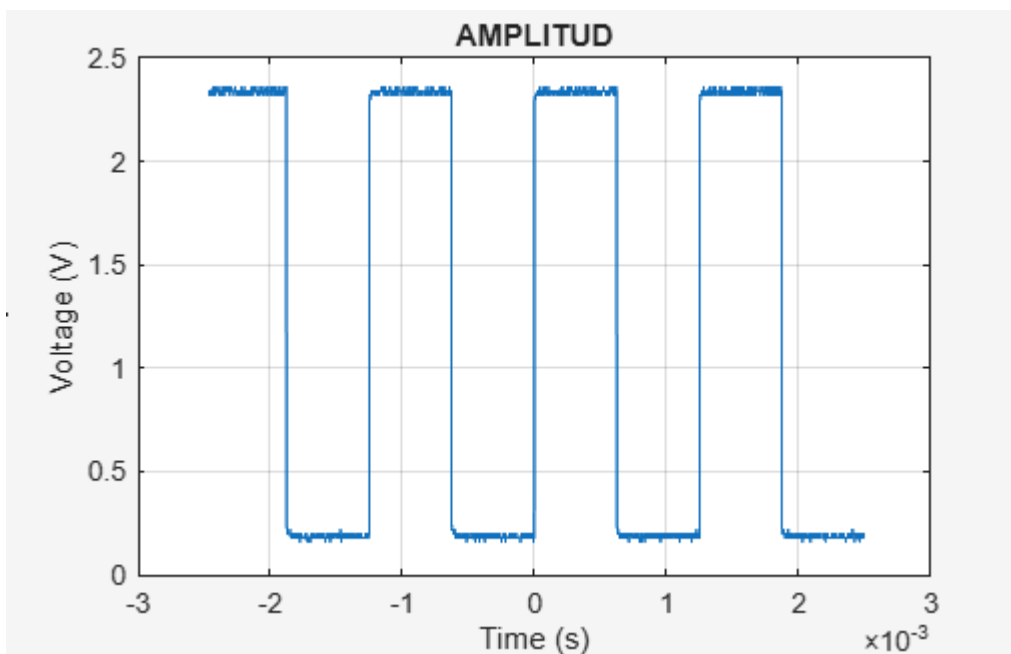
```
plot(t, v);
```

```
xlabel('Time (s)');
```

```
ylabel('Voltage (V)');
```

```
title('AMPLITUD');
```

```
grid on;
```



## DOMINIO FRECUENCIA

```
% Leer el archivo como tabla, empezando desde la fila 18
```

```
opts = detectImportOptions('DOMINIOFRECUENCIACUA2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

```
% Leer los datos
```

```
datos = readtable('DOMINIOFRECUENCIACUA2.csv', opts);
```

```
% Convertir las columnas correctas (4 y 5) a vectores
```

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```

v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);

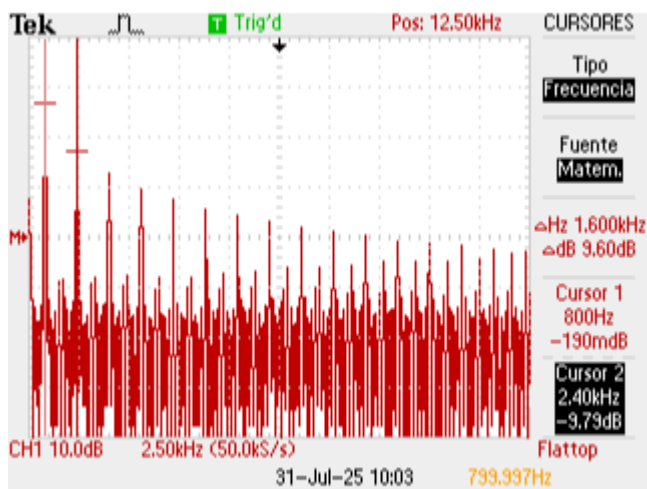
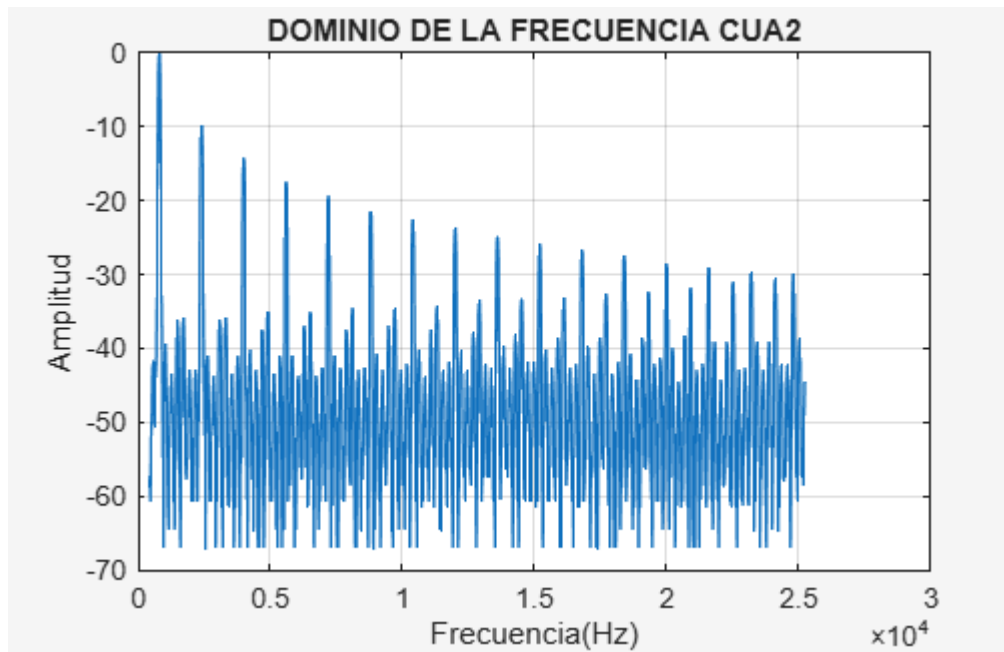
xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA CUA2');

grid on;

```



TDS 2012B - 10:06:20 a. m. 31/07/2025

## GRAFICA FFT (TEORICA)

% === Señal CUADRADA2 con DC ===

T = 0.00125;            % Periodo (s)

f = 1/T;                % Frecuencia fundamental (Hz)

Fs = 1e6;              % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T;        % Tiempo de 2 ciclos

% === Componentes armónicas ===

A1 = 1.273;

A3 = 0.424;

A5 = 0.254;

A7 = 0.1819;

A9 = 0.1415;

A11 = 0.1157;

% === Construcción de la señal con componente DC ===

v1 = A1\*sin(2\*pi\*f\*t);

v2 = A3\*sin(2\*pi\*3\*f\*t);

v3 = A5\*sin(2\*pi\*5\*f\*t);

v4 = A7\*sin(2\*pi\*7\*f\*t);

v5 = A9\*sin(2\*pi\*9\*f\*t);

v6 = A11\*sin(2\*pi\*11\*f\*t);

VT = 1.25 + v1 + v2 + v3 + v4 + v5 + v6; % Señal cuadrada con DC

% === Gráfica en el dominio del tiempo ===

figure;

plot(t, VT);

xlabel('Tiempo (s)');

ylabel('Amplitud');



```

title('Señal CUADRADA2 con DC = 1.25 V');

grid on;

% === FFT de la señal ===

N = length(VT);

Y = fft(VT);

Y_mag = abs(Y)/N;

Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

% === Conversión a decibelios ===

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===

f_axis = (0:N-1)*(Fs/N);

% === Gráfica del espectro en dB ===

figure;

plot(f_axis(1:N/2), Y_dB(1:N/2), 'k', 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

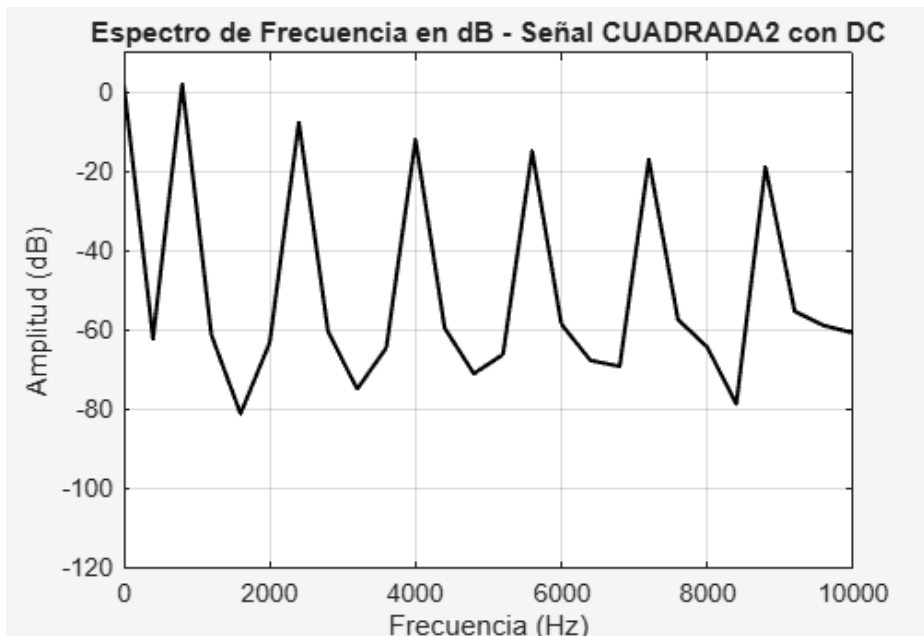
title('Espectro de Frecuencia en dB - Señal CUADRADA2 con DC');

grid on;

xlim([0 10000]); % Visualizar hasta 10 kHz

ylim([-120 10]); % Rango de amplitudes en dB

```



### GRAFICA IFFT

% === Señal CUADRADA2 con DC ===

T = 0.00125;      % Periodo (s)

f = 1/T;      % Frecuencia fundamental (Hz)

Fs = 1e6;      % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T;      % Tiempo de 2 ciclos

% === Componentes armónicas ===

A1 = 1.273;

A3 = 0.424;

A5 = 0.254;

A7 = 0.1819;

A9 = 0.1415;

A11 = 0.1157;

% === Construcción de la señal con componente DC ===

```

v1 = A1*sin(2*pi*f*t);
v2 = A3*sin(2*pi*3*f*t);
v3 = A5*sin(2*pi*5*f*t);
v4 = A7*sin(2*pi*7*f*t);
v5 = A9*sin(2*pi*9*f*t);
v6 = A11*sin(2*pi*11*f*t);
VT = 1.25 + v1 + v2 + v3 + v4 + v5 + v6; % Señal cuadrada con DC

```

```

% === Gráfica en el dominio del tiempo ===

```

```

figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal CUADRADA2 con DC = 1.25 V');
grid on;

```

```

% === FFT de la señal ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

```

```

% === Conversión a decibelios ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```

% === Eje de frecuencias ===

```

```

f_axis = (0:N-1)*(Fs/N);

```

```

% === Gráfica del espectro en dB ===

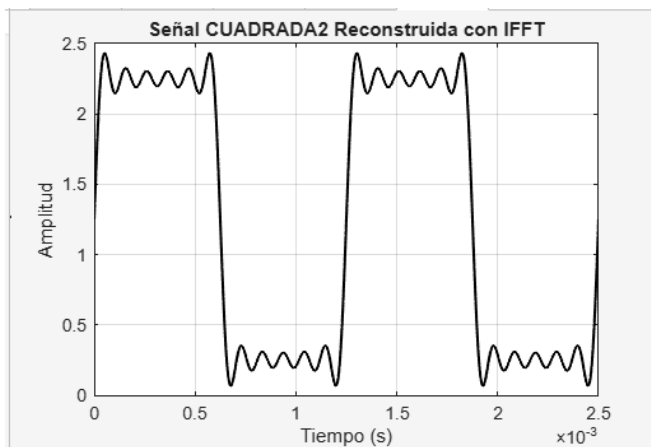
```

```
figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'k', 'LineWidth', 1.5);
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
title('Espectro de Frecuencia en dB - Señal CUADRADA2 con DC');
grid on;
xlim([0 10000]); % Visualizar hasta 10 kHz
ylim([-120 10]); % Rango de amplitudes en dB
```

```
% === Reconstrucción con IFFT ===
VT_rec = ifft(Y, 'symmetric'); % Señal reconstruida real
```

```
% === Gráfica de la señal reconstruida con IFFT ===
```

```
figure;
plot(t, VT_rec, 'k', 'LineWidth', 1.5);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal CUADRADA2 Reconstruida con IFFT');
grid on;
```



## PHYTON

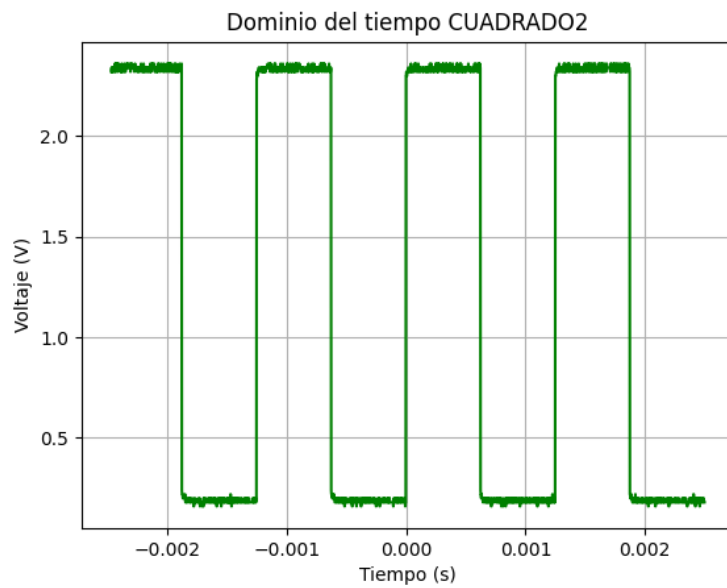
```
import pandas as pd
import matplotlib.pyplot as plt

# Ruta al archivo
archivo = 'DOMINIOTIEMPOCUA2.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar
plt.plot(tiempo, voltaje, color='g')
plt.title('Dominio del tiempo CUADRADO2')
plt.xlabel('Tiempo (s)')
plt.ylabel('Voltaje (V)')
plt.grid(True)
plt.show()
```



```
import pandas as pd

import matplotlib.pyplot as plt

# Ruta al archivo
archivo = 'DOMINIOFRECUENCIACUA2.csv'

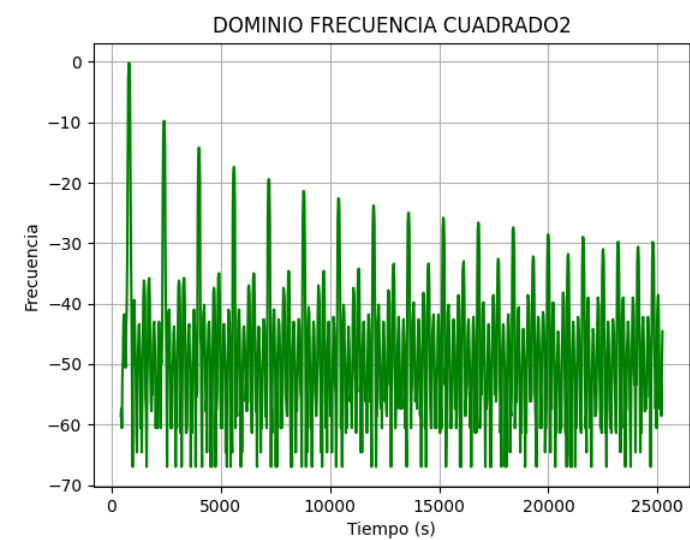
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar
plt.plot(tiempo, voltaje, color='g')
plt.title('DOMINIO FRECUENCIA CUADRADO2')
plt.xlabel('Tiempo (s)')
plt.ylabel('Frecuencia')
```

```
plt.grid(True)
```

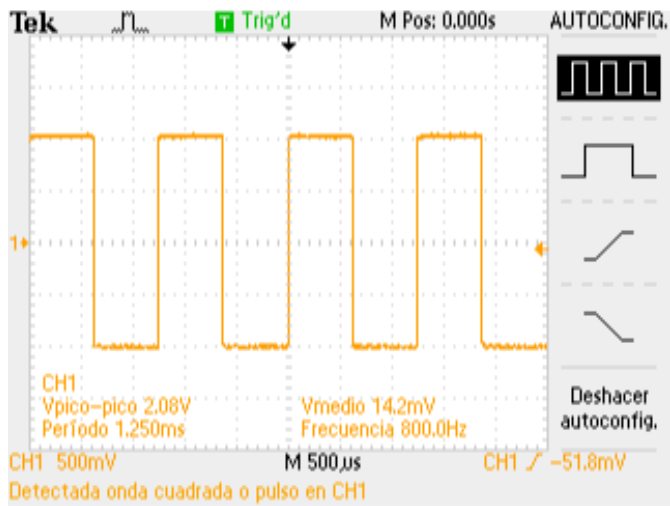
```
plt.show()
```



**PULSO1**

**TOMA DATOS EXPERIMENTALES**

FRECUENCIA	800 Hz					
Vp	1					
DC	0					
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A		
				d.B	Vrms	VP(v)
PULSO1	50%	1	800	-0,19	0,9784	1,3836
		3	2400	-10,2	0,3090	0,4370
		5	4000	-15	0,1778	0,2515
		7	5600	-17,8	0,1288	0,1822
		9	7200	-19,8	0,1023	0,1447
		11	8800	-21,8	0,0813	0,1150



## GRAFICA MATLAB CON CODIGO

```

T=0.00125;
f=1/T;
t=0:1/1000000:2*T;
DC=0.5;
A1=1.3836;
A3=0.4370;
A5=0.2515;
A7=0.1822;
A9=0.1447;
A11=0.1150;
v1=A1*sin(2*pi*f*t);
v2=A3*sin(2*pi*3*f*t);
v3=A5*sin(2*pi*5*f*t);
v4=A7*sin(2*pi*7*f*t);
v5=A9*sin(2*pi*9*f*t);
v6=A11*sin(2*pi*11*f*t);

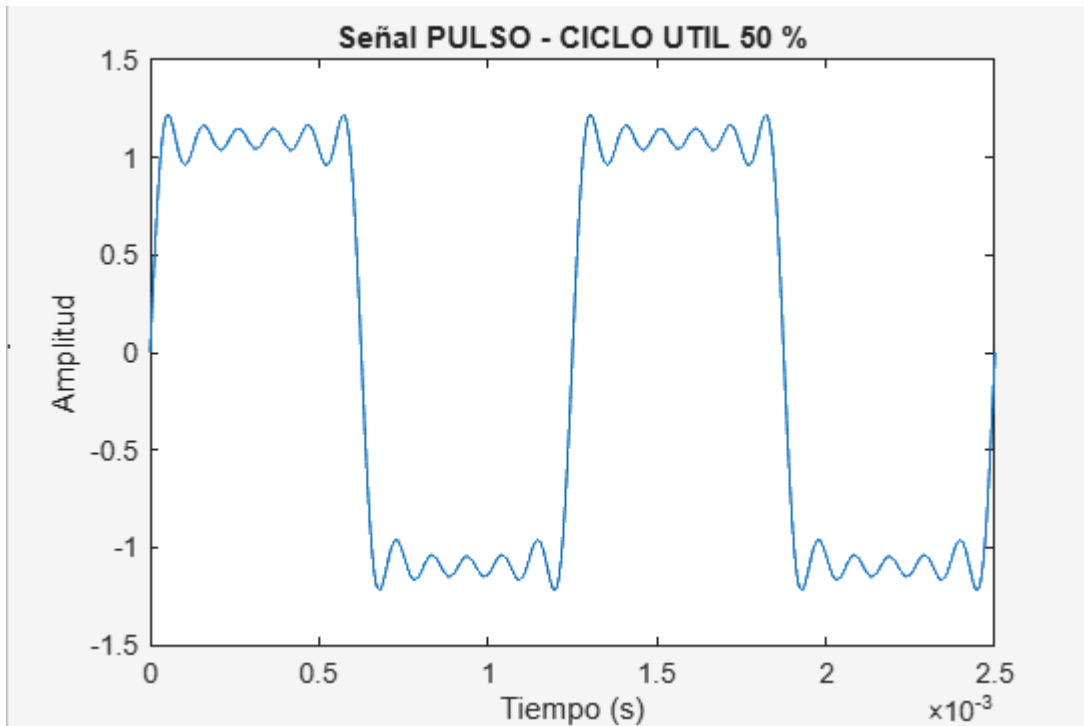
```



```

VT=(v1+v2+v3+v4+v5+v6);
plot(t,VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO - CICLO UTIL 50 %');

```



### GRAFICA MATLAB CON DATOS EXPORTADO

```

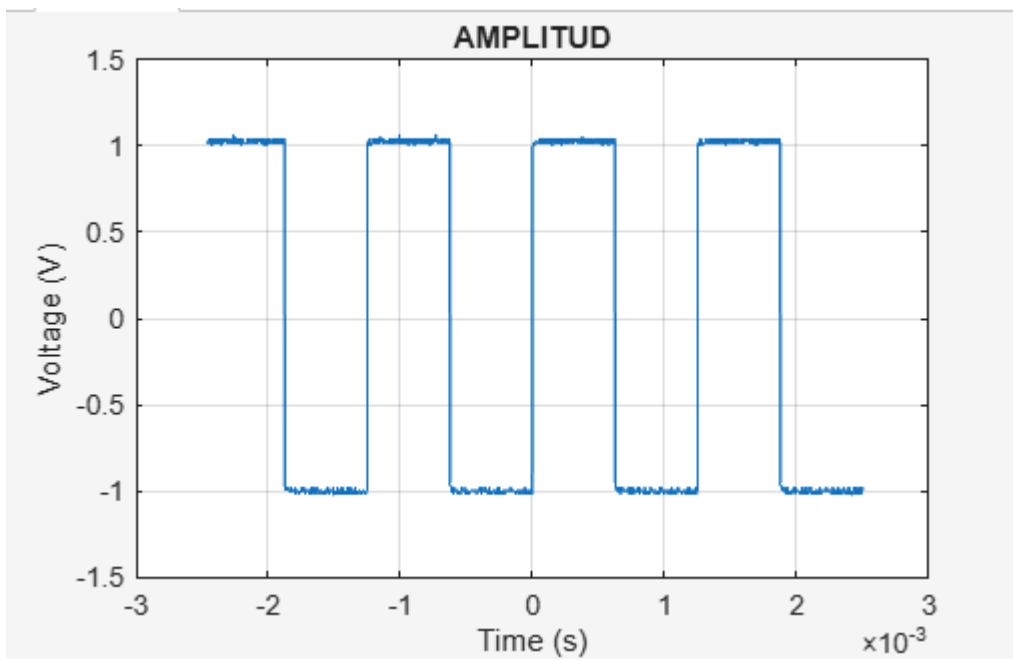
% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOTIEMPOPULSO1.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos
datos = readtable('DOMINIOTIEMPOPULSO1.csv', opts);
% Convertir las columnas correctas (4 y 5) a vectores

```

```

t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje
% Graficar
plot(t, v);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('AMPLITUD');
grid on;

```



## DOMINIO FRECUENCIA

```

% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOFRECUENCIAPULSO1.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos
datos = readtable('DOMINIOFRECUENCIAPULSO1.csv', opts);
% Convertir las columnas correctas (4 y 5) a vectores
t = datos{:,4}; % Columna 4 = Tiempo

```

```

v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);

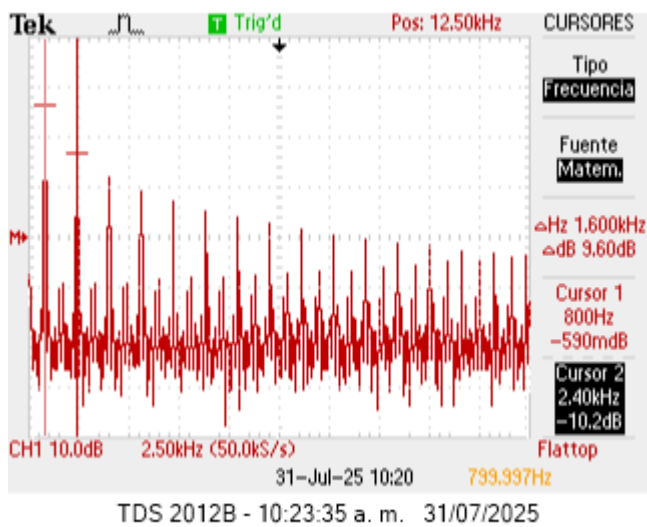
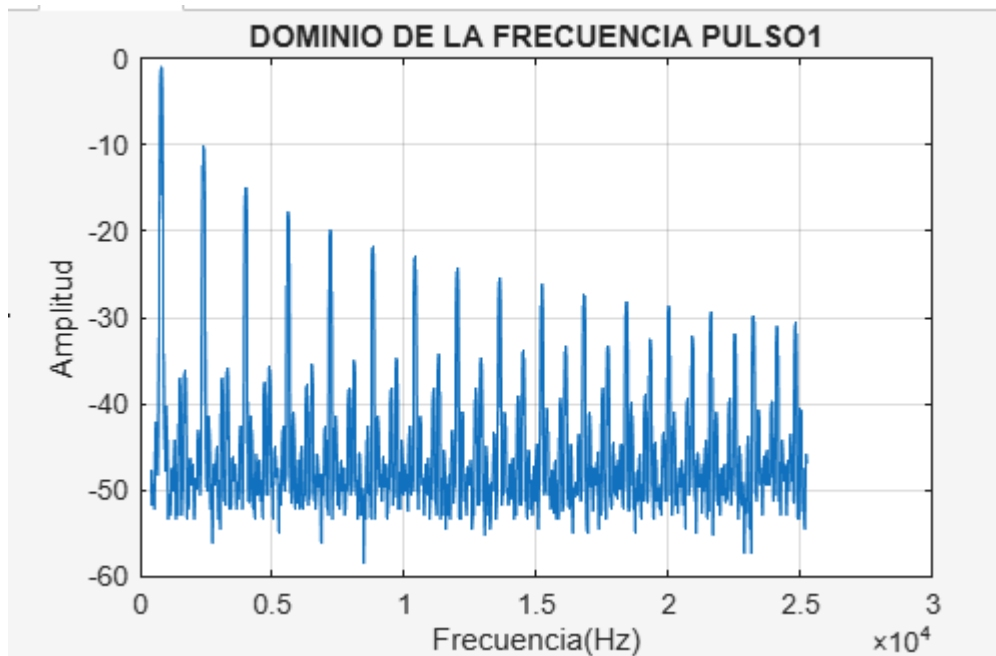
xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA PULSO1');

grid on;

```



## GRAFICA FFT (TEORICA)

% === Señal PULSO - CICLO ÚTIL 50% ===

T = 0.00125;            % Periodo (s)

f = 1/T;                % Frecuencia fundamental (Hz)

Fs = 1e6;              % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T;        % Tiempo total de 2 ciclos

% === Componentes armónicas ===

DC = 0.5;              % Nivel DC (no usado en este caso, pero definido)

A1 = 1.3836;

A3 = 0.4370;

A5 = 0.2515;

A7 = 0.1822;

A9 = 0.1447;

A11 = 0.1150;

% === Construcción de la señal PULSO ===

v1 = A1\*sin(2\*pi\*f\*t);

v2 = A3\*sin(2\*pi\*3\*f\*t);

v3 = A5\*sin(2\*pi\*5\*f\*t);

v4 = A7\*sin(2\*pi\*7\*f\*t);

v5 = A9\*sin(2\*pi\*9\*f\*t);

v6 = A11\*sin(2\*pi\*11\*f\*t);

VT = v1 + v2 + v3 + v4 + v5 + v6;

% === Gráfica de la señal en el dominio del tiempo ===

figure;

plot(t, VT);

xlabel('Tiempo (s)');

```

ylabel('Amplitud');
title('Señal PULSO - CICLO ÚTIL 50%');
grid on;

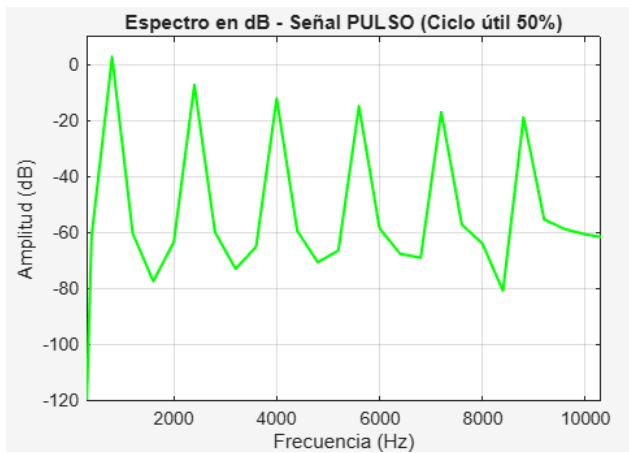
% === FFT de la señal ===
N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

% === Conversión a decibelios ===
Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===
f_axis = (0:N-1)*(Fs/N);

% === Gráfica del espectro en dB ===
figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'g', 'LineWidth', 1.5);
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
title('Espectro en dB - Señal PULSO (Ciclo útil 50%)');
grid on;
xlim([0 10000]); % Hasta 10 kHz para visualización clara
ylim([-120 10]); % Rango típico en dB

```



## GRAFICA IFFT

% === Señal PULSO - CICLO ÚTIL 50% ===

T = 0.00125;            % Periodo (s)

f = 1/T;                % Frecuencia fundamental (Hz)

Fs = 1e6;              % Frecuencia de muestreo (1 MHz)

t = 0:1/Fs:2\*T;        % Tiempo total de 2 ciclos

% === Componentes armónicas ===

DC = 0.5;              % Nivel DC (no usado en este caso, pero definido)

A1 = 1.3836;

A3 = 0.4370;

A5 = 0.2515;

A7 = 0.1822;

A9 = 0.1447;

A11 = 0.1150;

% === Construcción de la señal PULSO ===

v1 = A1\*sin(2\*pi\*f\*t);

v2 = A3\*sin(2\*pi\*3\*f\*t);

v3 = A5\*sin(2\*pi\*5\*f\*t);

```

v4 = A7*sin(2*pi*7*f*t);
v5 = A9*sin(2*pi*9*f*t);
v6 = A11*sin(2*pi*11*f*t);
VT = v1 + v2 + v3 + v4 + v5 + v6;

```

```

% === Gráfica de la señal en el dominio del tiempo ===

```

```

figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO - CICLO ÚTIL 50%');
grid on;

```

```

% === FFT de la señal ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

```

```

% === Conversión a decibelios ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```

% === Eje de frecuencias ===

```

```

f_axis = (0:N-1)*(Fs/N);

```

```

% === Gráfica del espectro en dB ===

```

```

figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'g', 'LineWidth', 1.5);
xlabel('Frecuencia (Hz)');

```

```

ylabel('Amplitud (dB)');

title('Espectro en dB - Señal PULSO (Ciclo útil 50%)');

grid on;

xlim([0 10000]);

ylim([-120 10]);

% === Reconstrucción con IFFT ===

VT_rec = ifft(Y, 'symmetric'); % Reconstrucción en dominio del tiempo

% === Gráfica de la señal reconstruida con IFFT ===

figure;

plot(t, VT_rec, 'g', 'LineWidth', 1.5);

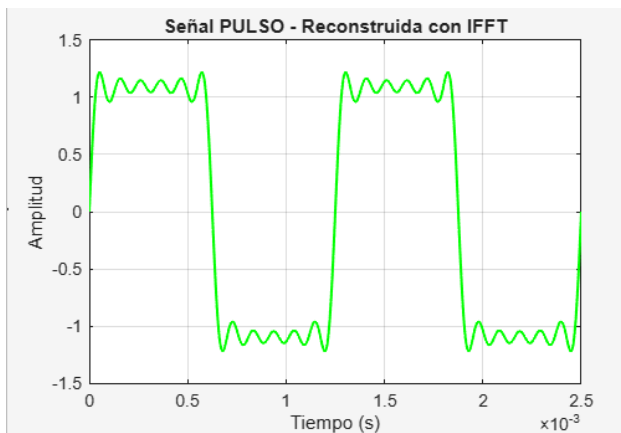
xlabel('Tiempo (s)');

ylabel('Amplitud');

title('Señal PULSO - Reconstruida con IFFT');

grid on;

```



## PHYTON

```

import pandas as pd

import matplotlib.pyplot as plt

# Ruta al archivo

```



```
archivo = 'DOMINIOTIEMPOPULSO1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#800080')
```

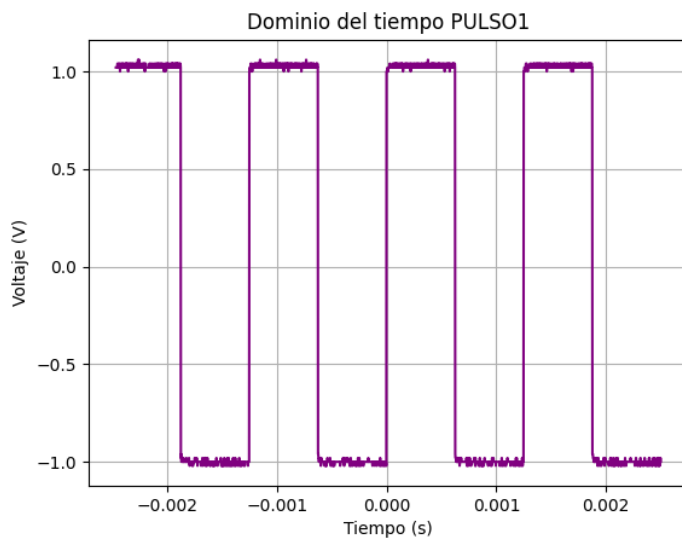
```
plt.title('Dominio del tiempo PULSO1')
```

```
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Voltaje (V)')
```

```
plt.grid(True)
```

```
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIAPULSO1.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#800080')
```

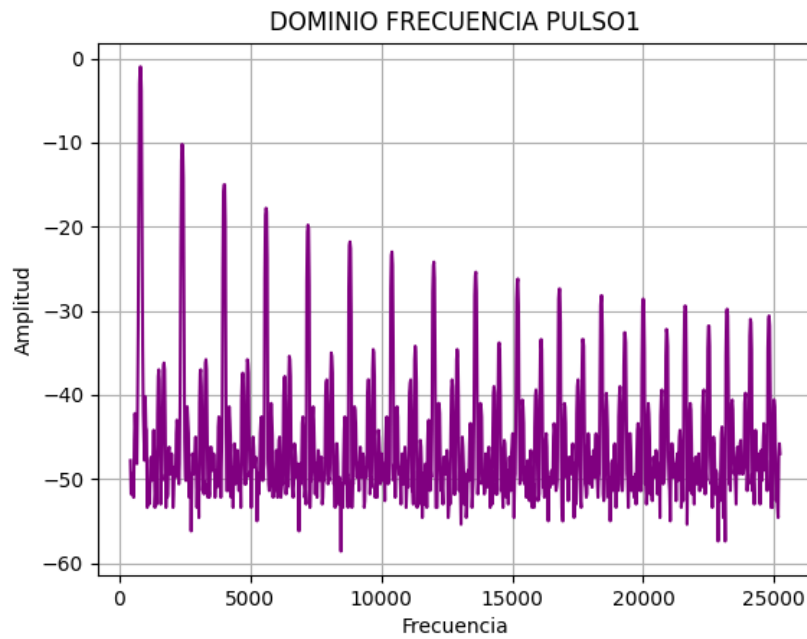
```
plt.title('DOMINIO FRECUENCIA PULSO1')
```

```
plt.xlabel('Frecuencia')
```

```
plt.ylabel('Amplitud')
```

```
plt.grid(True)
```

```
plt.show()
```

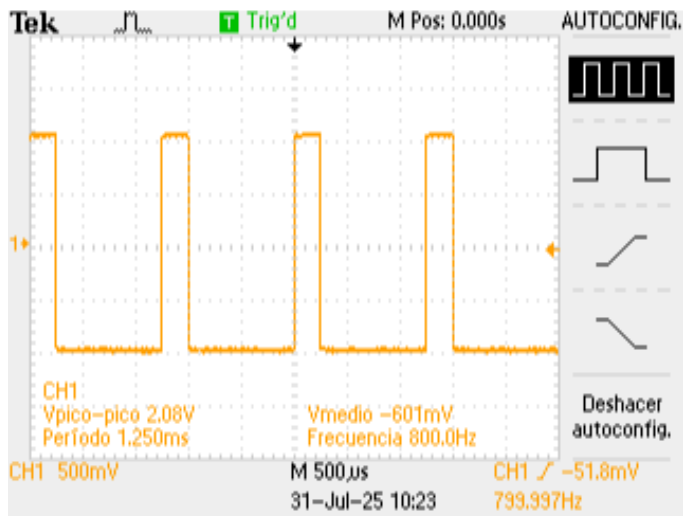


## PULSO2

### TOMA DATOS EXPERIMENTALES

FRECUENCIA	800 Hz						
Vp	1						
DC	0						
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO ( $2 \cdot A / n \cdot \pi$ )
				d.B	Vrms	VP(v)	
PULSO2	20%	1	800	-5,39	0,5377	0,7604	0,63662
		2	1600	-7,39	0,4271	0,6040	0,31831
		3	2400	-11	0,2818	0,3986	0,21221
		4	3200	-17,8	0,1288	0,1822	0,15915
		5	3650	-30,2	0,0309	0,0437	0,12732
		6	4350	-33	0,0224	0,0317	0,10610
		7	4800	-21	0,0891	0,1260	0,09095
		8	5600	-18,2	0,1230	0,1740	0,07958
		9	6400	-19,4	0,1072	0,1515	0,07074
		10	7200	-24,6	0,0589	0,0833	0,06366

		11	8000	- 36,2	0,0155	0,0219	0,05787
--	--	----	------	-----------	--------	--------	---------



TDS 2012B - 10:26:27 a. m. 31/07/2025

## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000000:2\*T;

D=0.2;

A1=0.636;

A2=0.318;

A3=0.212;

A4=0.159;

A5=0.127;

A6=0.106;

A7=0.090;

A8=0.079;

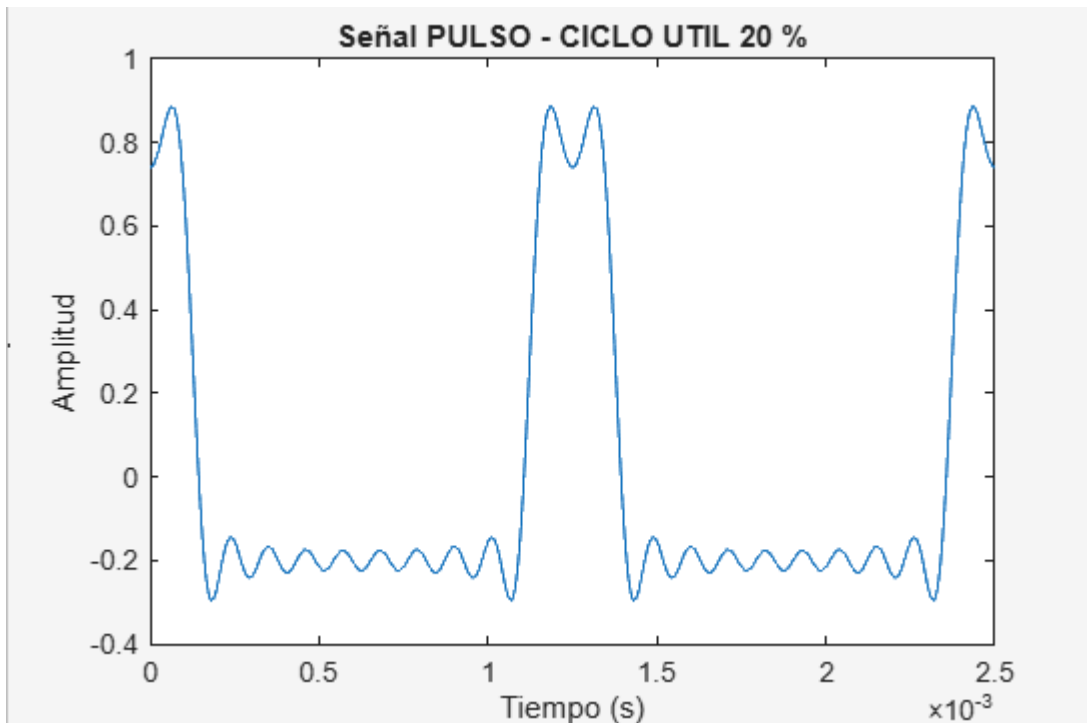
A9=0.070;

A10=0.063;

A11=0.057;

```
v1=A1*sin(1*pi*D)*cos(2*pi*1*f*t);  
v2=A2*sin(2*pi*D)*cos(2*pi*2*f*t);  
v3=A3*sin(3*pi*D)*cos(2*pi*3*f*t);  
v4=A4*sin(4*pi*D)*cos(2*pi*4*f*t);  
v5=A5*sin(5*pi*D)*cos(2*pi*5*f*t);  
v6=A6*sin(6*pi*D)*cos(2*pi*6*f*t);  
v7=A7*sin(7*pi*D)*cos(2*pi*7*f*t);  
v8=A8*sin(8*pi*D)*cos(2*pi*8*f*t);  
v9=A9*sin(9*pi*D)*cos(2*pi*9*f*t);  
v10=A10*sin(10*pi*D)*cos(2*pi*10*f*t);  
v11=A11*sin(11*pi*D)*cos(2*pi*11*f*t);
```

```
VT=(v1+v2+v3+v4+v5+v6+v7+v8+v9+v10+v11);  
plot(t,VT);  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
title('Señal PULSO - CICLO UTIL 20 %');
```



### GRAFICA MATLAB CON DATOS EXPORTADO

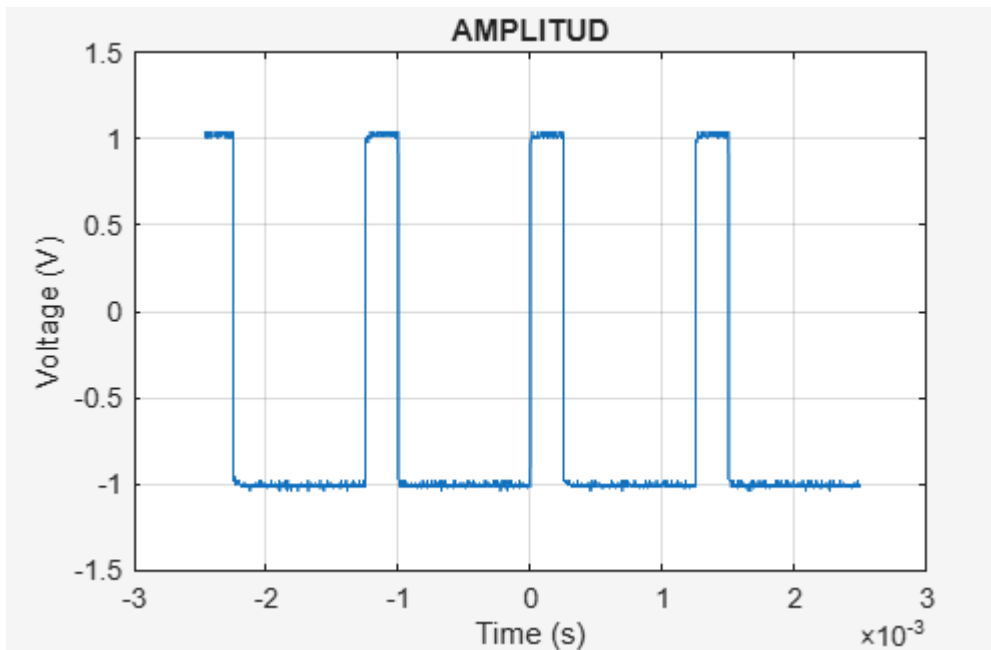
```
% Leer el archivo como tabla, empezando desde la fila 18
opts = detectImportOptions('DOMINIOTIEMPOPULSO2.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ';'; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

% Leer los datos
datos = readtable('DOMINIOTIEMPOPULSO2.csv', opts);

% Convertir las columnas correctas (4 y 5) a vectores
t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje

% Graficar
plot(t, v);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('AMPLITUD');
```

grid on;



## DOMINIO FRECUENCIA

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOFRECUENCIAPULSO2.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOFRECUENCIAPULSO2.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

% Graficar

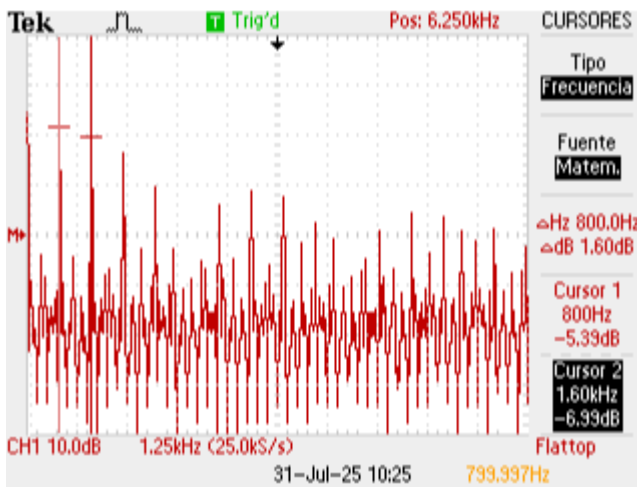
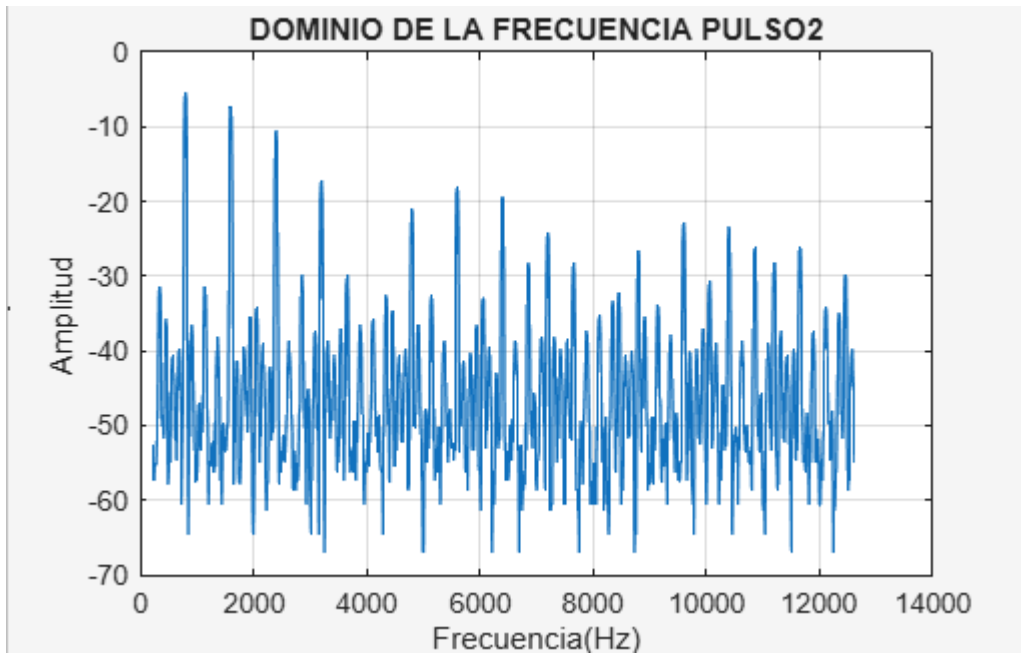
```
plot(t, v);
```

```
xlabel('Frecuencia(Hz)');
```

```
ylabel('Amplitud');
```

```
title('DOMINIO DE LA FRECUENCIA PULSO2');
```

grid on;



### GRAFICA FFT (TEORICA)

% === Señal PULSO 2 - CICLO ÚTIL 20% ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz) = 800

Fs = 1e9; % Frecuencia de muestreo (1 GHz)

t = 0:1/Fs:2\*T; % Tiempo total de 2 ciclos



% === Ciclo útil (Duty cycle) ===

D = 0.2;

% === Amplitudes por armónico (precalculadas) ===

A1 = 0.636; A2 = 0.318; A3 = 0.212; A4 = 0.159; A5 = 0.127;

A6 = 0.106; A7 = 0.090; A8 = 0.079; A9 = 0.070; A10 = 0.063;

A11 = 0.057;

% === Construcción de la señal compuesta ===

v1 = A1 \* sin(1\*pi\*D) \* cos(2\*pi\*1\*f\*t);

v2 = A2 \* sin(2\*pi\*D) \* cos(2\*pi\*2\*f\*t);

v3 = A3 \* sin(3\*pi\*D) \* cos(2\*pi\*3\*f\*t);

v4 = A4 \* sin(4\*pi\*D) \* cos(2\*pi\*4\*f\*t);

v5 = A5 \* sin(5\*pi\*D) \* cos(2\*pi\*5\*f\*t);

v6 = A6 \* sin(6\*pi\*D) \* cos(2\*pi\*6\*f\*t);

v7 = A7 \* sin(7\*pi\*D) \* cos(2\*pi\*7\*f\*t);

v8 = A8 \* sin(8\*pi\*D) \* cos(2\*pi\*8\*f\*t);

v9 = A9 \* sin(9\*pi\*D) \* cos(2\*pi\*9\*f\*t);

v10 = A10 \* sin(10\*pi\*D) \* cos(2\*pi\*10\*f\*t);

v11 = A11 \* sin(11\*pi\*D) \* cos(2\*pi\*11\*f\*t);

VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;

% === Gráfica de la señal en el tiempo ===

figure;

plot(t, VT);

xlabel('Tiempo (s)');

ylabel('Amplitud');

```

title('Señal PULSO - CICLO ÚTIL 20%');

grid on;

% === FFT de la señal ===

N = length(VT);

Y = fft(VT);

Y_mag = abs(Y)/N;

Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes

% === Conversión a decibelios ===

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===

f_axis = (0:N-1)*(Fs/N);

% === Gráfica del espectro en dB ===

figure;

plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.2 0.6 0.2], 'LineWidth', 1.5);

xlabel('Frecuencia (Hz)');

ylabel('Amplitud (dB)');

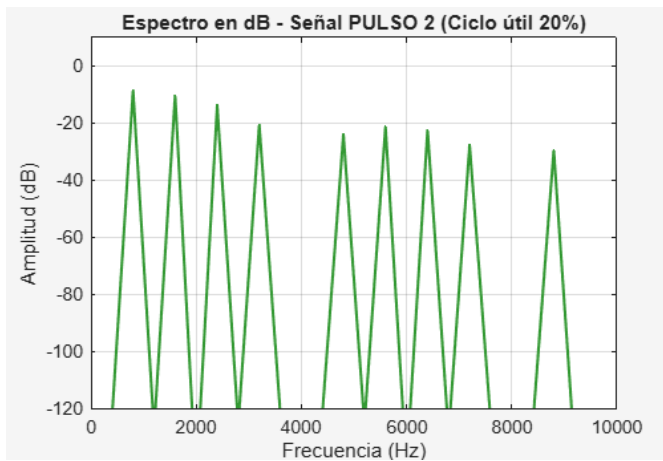
title('Espectro en dB - Señal PULSO (Ciclo útil 20%)');

grid on;

xlim([0 10000]); % Hasta 10 kHz

ylim([-120 10]); % Escala típica en dB

```



## GRAFICA IFFT

% === Señal PULSO 2 - CICLO ÚTIL 20% ===

T = 0.00125;            % Periodo (s)

f = 1/T;                % Frecuencia fundamental (Hz) = 800

Fs = 1e9;              % Frecuencia de muestreo (1 GHz)

t = 0:1/Fs:2\*T;        % Tiempo total de 2 ciclos

% === Ciclo útil (Duty cycle) ===

D = 0.2;

% === Amplitudes por armónico (precalculadas) ===

A1 = 0.636; A2 = 0.318; A3 = 0.212; A4 = 0.159; A5 = 0.127;

A6 = 0.106; A7 = 0.090; A8 = 0.079; A9 = 0.070; A10 = 0.063;

A11 = 0.057;

% === Construcción de la señal compuesta ===

v1 = A1 \* sin(1\*pi\*D) \* cos(2\*pi\*1\*f\*t);

v2 = A2 \* sin(2\*pi\*D) \* cos(2\*pi\*2\*f\*t);

v3 = A3 \* sin(3\*pi\*D) \* cos(2\*pi\*3\*f\*t);

v4 = A4 \* sin(4\*pi\*D) \* cos(2\*pi\*4\*f\*t);

```

v5 = A5 * sin(5*pi*D) * cos(2*pi*5*f*t);
v6 = A6 * sin(6*pi*D) * cos(2*pi*6*f*t);
v7 = A7 * sin(7*pi*D) * cos(2*pi*7*f*t);
v8 = A8 * sin(8*pi*D) * cos(2*pi*8*f*t);
v9 = A9 * sin(9*pi*D) * cos(2*pi*9*f*t);
v10 = A10 * sin(10*pi*D) * cos(2*pi*10*f*t);
v11 = A11 * sin(11*pi*D) * cos(2*pi*11*f*t);

VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;

```

```

% === Gráfica de la señal en el tiempo ===

```

```

figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO 2 - CICLO ÚTIL 20%');
grid on;

```

```

% === FFT de la señal ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1);

```

```

% === Conversión a decibelios ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```

% === Eje de frecuencias ===

```

```

f_axis = (0:N-1)*(Fs/N);

```

```
% === Gráfica del espectro en dB ===
```

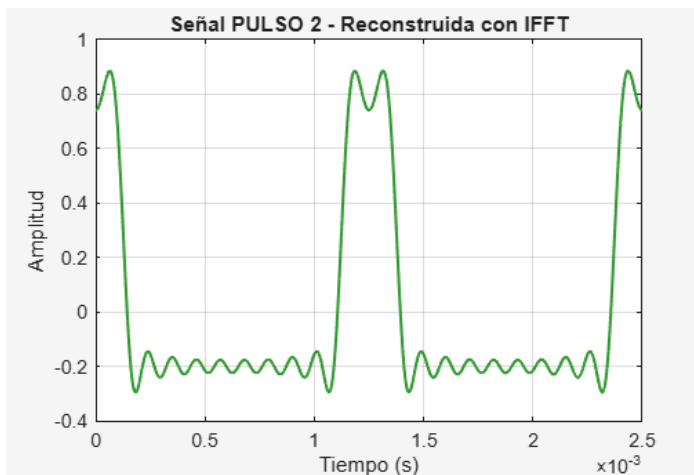
```
figure;  
plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.2 0.6 0.2], 'LineWidth', 1.5);  
xlabel('Frecuencia (Hz)');  
ylabel('Amplitud (dB)');  
title('Espectro en dB - Señal PULSO 2 (Ciclo útil 20%)');  
grid on;  
xlim([0 10000]);  
ylim([-120 10]);
```

```
% === Reconstrucción con IFFT ===
```

```
VT_rec = ifft(Y, 'symmetric');
```

```
% === Gráfica de la señal reconstruida con IFFT ===
```

```
figure;  
plot(t, VT_rec, 'Color', [0.2 0.6 0.2], 'LineWidth', 1.5);  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
title('Señal PULSO 2 - Reconstruida con IFFT');  
grid on;
```



## PHYTON

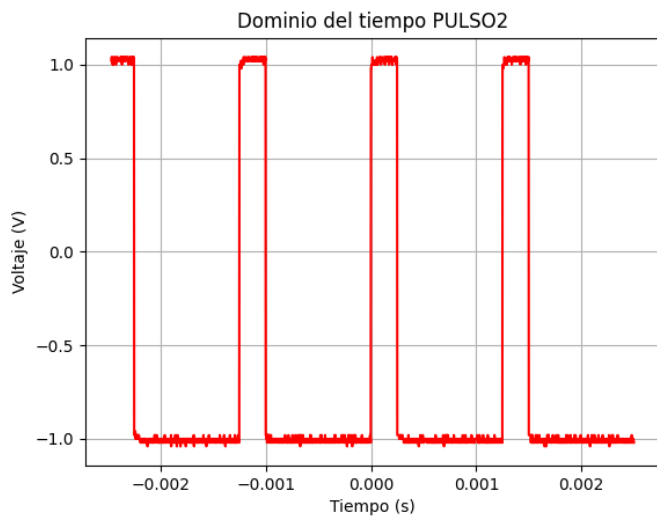
```
import pandas as pd
import matplotlib.pyplot as plt

# Ruta al archivo
archivo = 'DOMINIOTIEMPOPOPULSO2.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar
plt.plot(tiempo, voltaje, color='r')
plt.title('Dominio del tiempo PULSO2')
plt.xlabel('Tiempo (s)')
plt.ylabel('Voltaje (V)')
plt.grid(True)
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIAPULSO2.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='r')
```

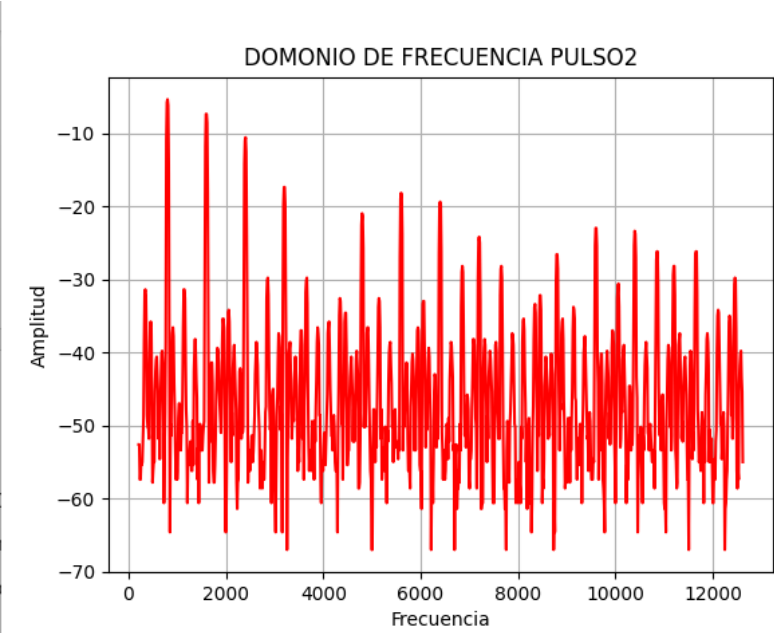
```
plt.title('DOMONIO DE FRECUENCIA PULSO2')
```

```
plt.xlabel('Frecuencia')
```

```
plt.ylabel('Amplitud')
```

```
plt.grid(True)

plt.show()
```



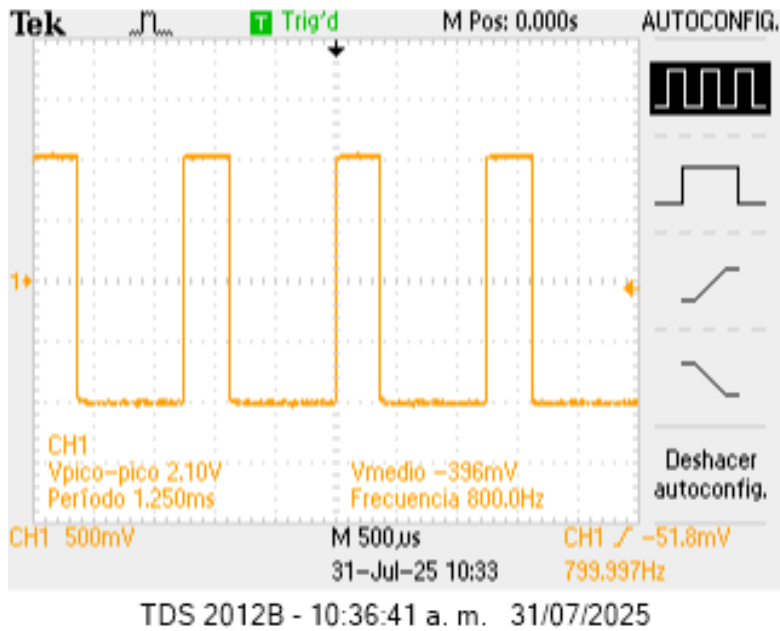
**PULSO3**

**TOMA DATOS EXPERIMENTALES**

FRECUENCIA		800 Hz					
Vp		1					
DC		0					
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO (2*A/n*π)
				d.B	Vrms	VP(v)	
PULSO3	30%	1	800	- 2,59	0,7422	1,0496	0,636620
		2	1600	- 7,39	0,4271	0,6040	0,318310
		3	2400	- 20,6	0,0933	0,1320	0,212207
		4	3200	- 17,4	0,1349	0,1908	0,159155
		5	4000	-15	0,1778	0,2515	0,127324
		6	4800	21	11,2202	15,8677	0,106103
		7	5600	- 28,2	0,0389	0,0550	0,090946
		8	6400	- 19,4	0,1072	0,1515	0,079577
		9	7200	- 21,8	0,0813	0,1150	0,070736



		10	8800	-23,8	0,0646	0,0913	0,063662
		11	9600	-23	0,0708	0,1001	0,057875



## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000000:2\*T;

D=0.3;

A1=0.636;

A2=0.318;

A3=0.212;

A4=0.159;

A5=0.127;

A6=0.106;

A7=0.090;

A8=0.079;

A9=0.070;

A10=0.063;

A11=0.057;

v1=A1\*sin(1\*pi\*D)\*cos(2\*pi\*1\*f\*t);

v2=A2\*sin(2\*pi\*D)\*cos(2\*pi\*2\*f\*t);

v3=A3\*sin(3\*pi\*D)\*cos(2\*pi\*3\*f\*t);

v4=A4\*sin(4\*pi\*D)\*cos(2\*pi\*4\*f\*t);

v5=A5\*sin(5\*pi\*D)\*cos(2\*pi\*5\*f\*t);

v6=A6\*sin(6\*pi\*D)\*cos(2\*pi\*6\*f\*t);

v7=A7\*sin(7\*pi\*D)\*cos(2\*pi\*7\*f\*t);

v8=A8\*sin(8\*pi\*D)\*cos(2\*pi\*8\*f\*t);

v9=A9\*sin(9\*pi\*D)\*cos(2\*pi\*9\*f\*t);

v10=A10\*sin(10\*pi\*D)\*cos(2\*pi\*10\*f\*t);

v11=A11\*sin(11\*pi\*D)\*cos(2\*pi\*11\*f\*t);

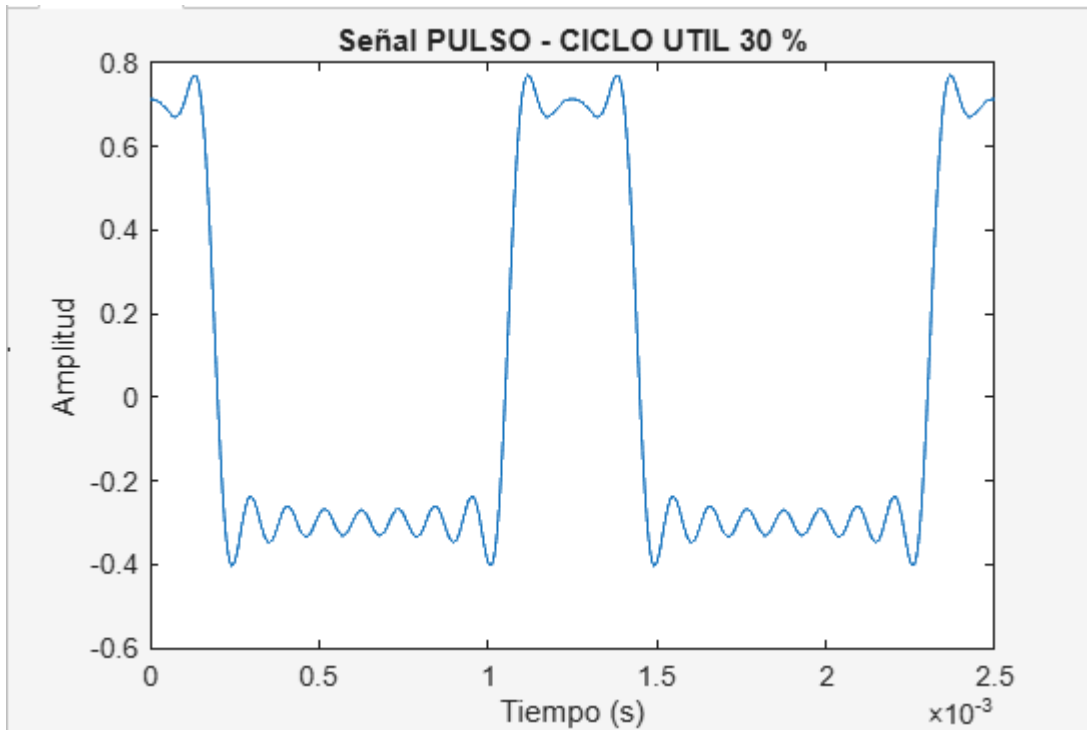
VT=(v1+v2+v3+v4+v5+v6+v7+v8+v9+v10+v11);

plot(t,VT);

xlabel('Tiempo (s)');

ylabel('Amplitud');

title('Señal PULSO - CICLO UTIL 30 %');



### GRAFICA MATLAB CON DATOS EXPORTADO

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOTIEMPOPULSO3.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOTIEMPOPULSO3.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos{:,4}; % Columna 4 = Tiempo
```

```
v = datos{:,5}; % Columna 5 = Voltaje
```

% Graficar

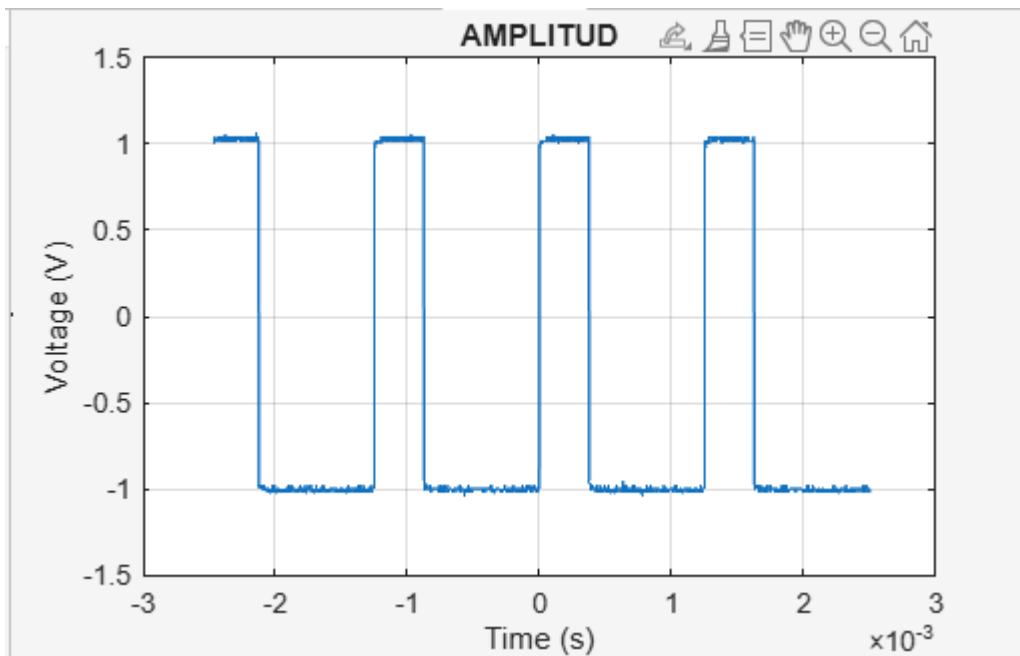
```
plot(t, v);
```

```
xlabel('Time (s)');
```

```
ylabel('Voltage (V)');
```

```
title('AMPLITUD');
```

grid on;



## DOMINIO FRECUENCIA

% Leer el archivo como tabla, empezando desde la fila 18

```
opts = detectImportOptions('DOMINIOFRECUENCIAPULSO3.csv');
```

```
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
```

```
opts.Delimiter = ','; % Delimitador por coma
```

```
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
```

% Leer los datos

```
datos = readtable('DOMINIOFRECUENCIAPULSO3.csv', opts);
```

% Convertir las columnas correctas (4 y 5) a vectores

```
t = datos(:,4); % Columna 4 = Tiempo
```

```
v = datos(:,5); % Columna 5 = Voltaje
```

% Graficar

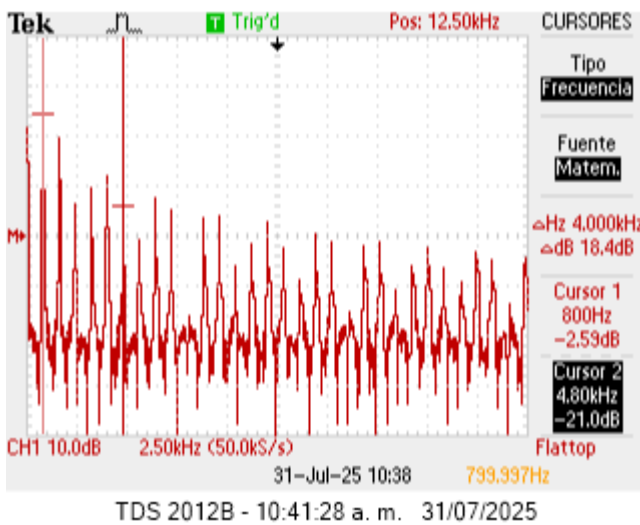
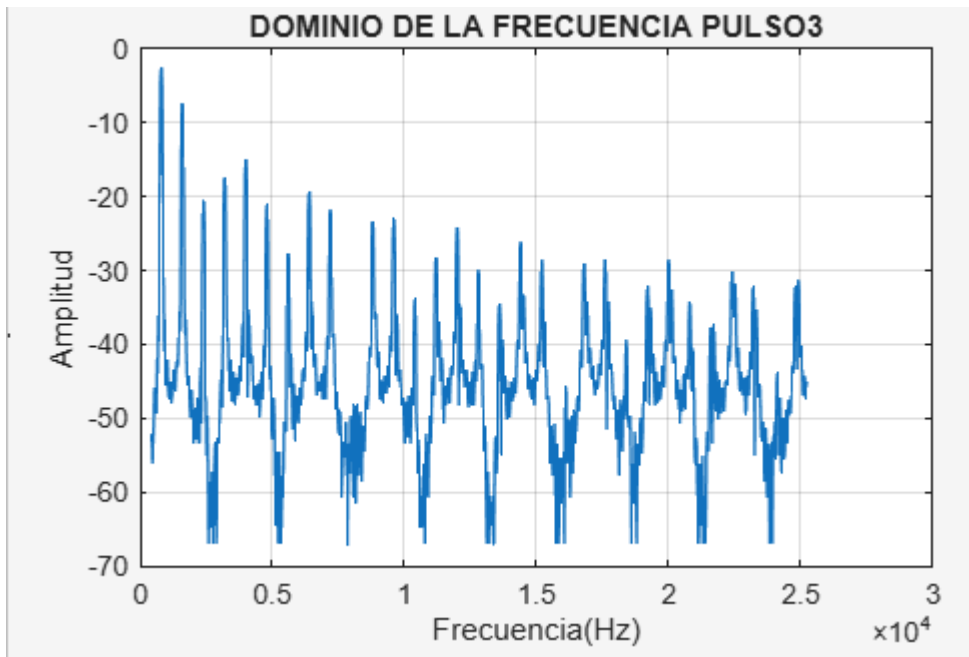
```
plot(t, v);
```

```
xlabel('Frecuencia(Hz)');
```

```
ylabel('Amplitud');
```

```
title('DOMINIO DE LA FRECUENCIA PULSO3');
```

grid on;



### GRAFICA FFT (TEORICA)

% == Señal PULSO 3 - CICLO ÚTIL 30% ==

$T = 0.00125$ ; % Periodo (s)

$f = 1/T$ ; % Frecuencia fundamental (Hz) = 800

$F_s = 1e9$ ; % Frecuencia de muestreo (1 GHz)

$t = 0:1/F_s:2*T$ ; % Tiempo total de 2 ciclos

```
% === Ciclo útil (Duty cycle) ===
```

```
D = 0.3;
```

```
% === Amplitudes por armónico (precalculadas) ===
```

```
A1 = 0.636; A2 = 0.318; A3 = 0.212; A4 = 0.159; A5 = 0.127;
```

```
A6 = 0.106; A7 = 0.090; A8 = 0.079; A9 = 0.070; A10 = 0.063;
```

```
A11 = 0.057;
```

```
% === Construcción de la señal compuesta ===
```

```
v1 = A1 * sin(1*pi*D) * cos(2*pi*1*f*t);
```

```
v2 = A2 * sin(2*pi*D) * cos(2*pi*2*f*t);
```

```
v3 = A3 * sin(3*pi*D) * cos(2*pi*3*f*t);
```

```
v4 = A4 * sin(4*pi*D) * cos(2*pi*4*f*t);
```

```
v5 = A5 * sin(5*pi*D) * cos(2*pi*5*f*t);
```

```
v6 = A6 * sin(6*pi*D) * cos(2*pi*6*f*t);
```

```
v7 = A7 * sin(7*pi*D) * cos(2*pi*7*f*t);
```

```
v8 = A8 * sin(8*pi*D) * cos(2*pi*8*f*t);
```

```
v9 = A9 * sin(9*pi*D) * cos(2*pi*9*f*t);
```

```
v10 = A10 * sin(10*pi*D) * cos(2*pi*10*f*t);
```

```
v11 = A11 * sin(11*pi*D) * cos(2*pi*11*f*t);
```

```
VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;
```

```
% === Gráfica en el dominio del tiempo ===
```

```
figure;
```

```
plot(t, VT);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal PULSO 3 - CICLO ÚTIL 30%');
```

```

grid on;

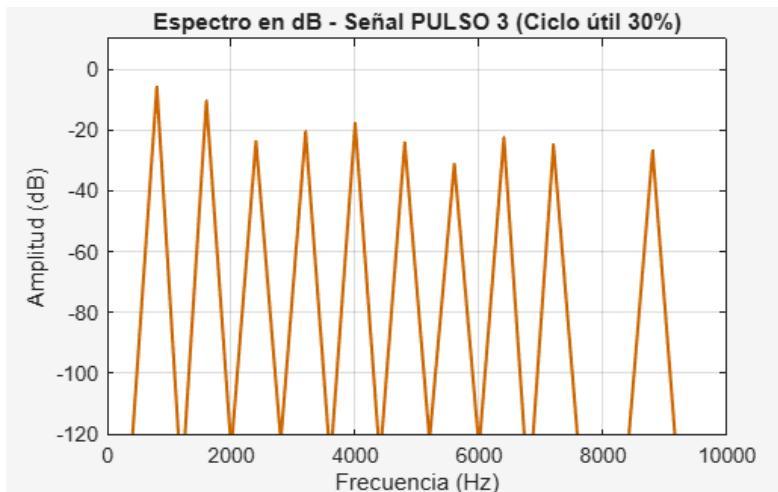
% === FFT ===
N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes excepto DC

% === Conversión a dB ===
Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===
f_axis = (0:N-1)*(Fs/N);

% === Gráfica del espectro en dB ===
figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.8 0.4 0], 'LineWidth', 1.5); % naranja
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
title('Espectro en dB - Señal PULSO 3 (Ciclo útil 30%)');
grid on;
xlim([0 10000]); % Hasta 10 kHz
ylim([-120 10]); % Escala común en dB

```



### GRAFICA IFFT

% === Señal PULSO 3 - CICLO ÚTIL 30% ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz) = 800

Fs = 1e9; % Frecuencia de muestreo (1 GHz)

t = 0:1/Fs:2\*T; % Tiempo total de 2 ciclos

% === Ciclo útil (Duty cycle) ===

D = 0.3;

% === Amplitudes por armónico (precalculadas) ===

A1 = 0.636; A2 = 0.318; A3 = 0.212; A4 = 0.159; A5 = 0.127;

A6 = 0.106; A7 = 0.090; A8 = 0.079; A9 = 0.070; A10 = 0.063;

A11 = 0.057;

% === Construcción de la señal compuesta ===

v1 = A1 \* sin(1\*pi\*D) \* cos(2\*pi\*1\*f\*t);

v2 = A2 \* sin(2\*pi\*D) \* cos(2\*pi\*2\*f\*t);

v3 = A3 \* sin(3\*pi\*D) \* cos(2\*pi\*3\*f\*t);

v4 = A4 \* sin(4\*pi\*D) \* cos(2\*pi\*4\*f\*t);



```

v5 = A5 * sin(5*pi*D) * cos(2*pi*5*f*t);
v6 = A6 * sin(6*pi*D) * cos(2*pi*6*f*t);
v7 = A7 * sin(7*pi*D) * cos(2*pi*7*f*t);
v8 = A8 * sin(8*pi*D) * cos(2*pi*8*f*t);
v9 = A9 * sin(9*pi*D) * cos(2*pi*9*f*t);
v10 = A10 * sin(10*pi*D) * cos(2*pi*10*f*t);
v11 = A11 * sin(11*pi*D) * cos(2*pi*11*f*t);

```

```

VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;

```

```

% === Gráfica de la señal en el dominio del tiempo ===

```

```

figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO 3 - CICLO ÚTIL 30%');
grid on;

```

```

% === FFT ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes excepto DC

```

```

% === Conversión a dB ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```

% === Eje de frecuencias ===

```

```

f_axis = (0:N-1)*(Fs/N);

```

```
% === Gráfica del espectro en dB ===
```

```
figure;
```

```
plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.8 0.4 0], 'LineWidth', 1.5); % naranja
```

```
xlabel('Frecuencia (Hz)');
```

```
ylabel('Amplitud (dB)');
```

```
title('Espectro en dB - Señal PULSO 3 (Ciclo útil 30%)');
```

```
grid on;
```

```
xlim([0 10000]);
```

```
ylim([-120 10]);
```

```
% === Reconstrucción con IFFT ===
```

```
VT_rec = ifft(Y, 'symmetric');
```

```
% === Gráfica de la señal reconstruida con IFFT ===
```

```
figure;
```

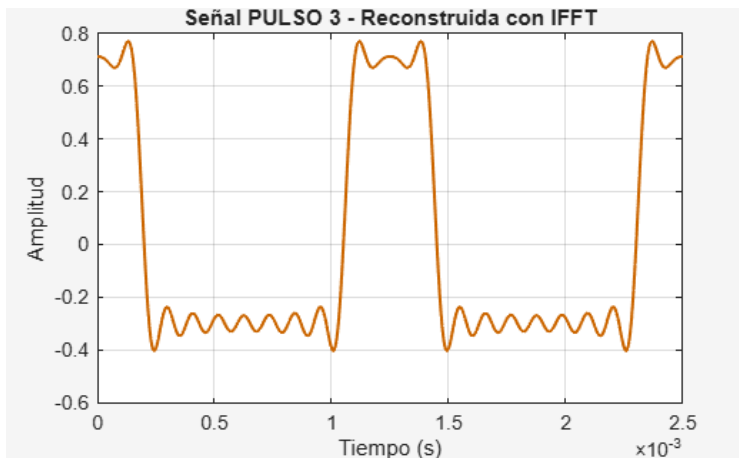
```
plot(t, VT_rec, 'Color', [0.8 0.4 0], 'LineWidth', 1.5);
```

```
xlabel('Tiempo (s)');
```

```
ylabel('Amplitud');
```

```
title('Señal PULSO 3 - Reconstruida con IFFT');
```

```
grid on;
```



## PHYTON

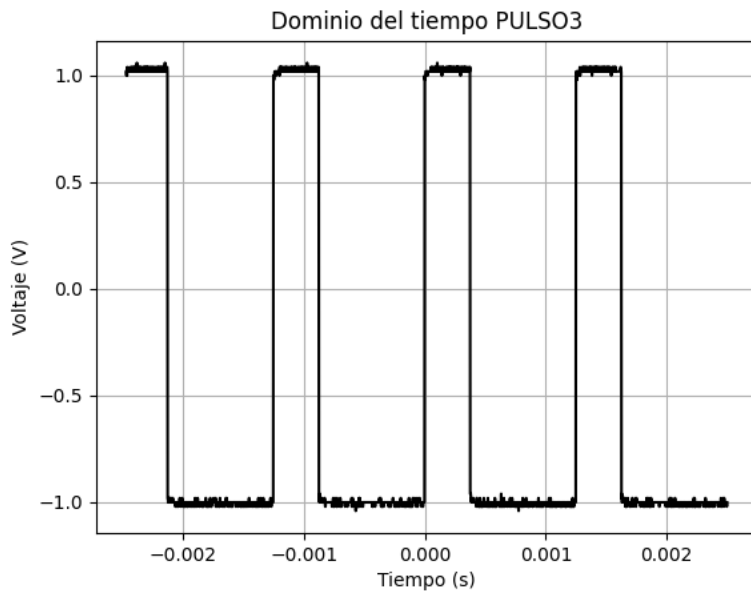
```
import pandas as pd
import matplotlib.pyplot as plt

# Ruta al archivo
archivo = 'DOMINIOTIEMPOPOPULSO3.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar
plt.plot(tiempo, voltaje, color='k')
plt.title('Dominio del tiempo PULSO3')
plt.xlabel('Tiempo (s)')
plt.ylabel('Voltaje (V)')
plt.grid(True)
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIAPULSO3.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='k')
```

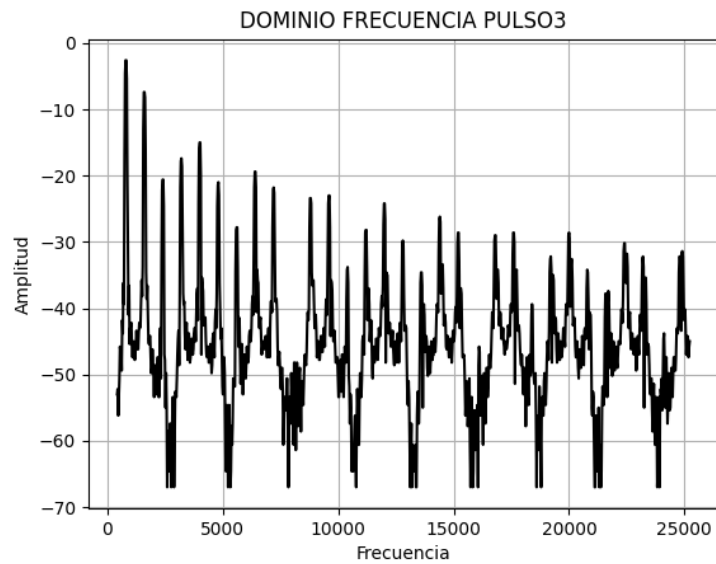
```
plt.title('DOMINIO FRECUENCIA PULSO3')
```

```
plt.xlabel('Frecuencia')
```

```
plt.ylabel('Amplitud')
```

```
plt.grid(True)
```

```
plt.show()
```

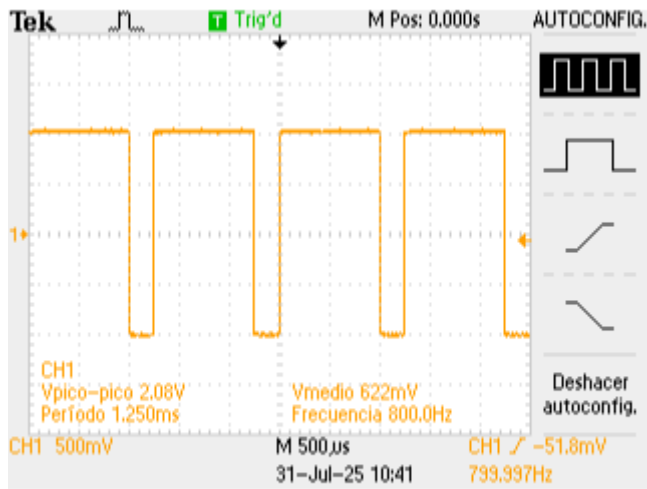


## PULSO 4

### TOMA DE DATOS EXPERIMENTALES

FRECUENCIA	800 Hz						
Vp	1						
DC	0						
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO ( $2 \cdot A / n \cdot \pi$ )
				d.B	Vrms	VP(v)	
PULSO4	80%	1	800	-5,39	0,5377	0,7604	0,63662
		2	1600	-12,8	0,2291	0,3240	0,31831
		3	2400	-15,6	0,1660	0,2347	0,21221
		4	3200	-18,8	0,1148	0,1624	0,15915
		5	3650	-21,2	0,0871	0,1232	0,12732
		6	4350	-23	0,0708	0,1001	0,10610
		7	4800	-24	0,0631	0,0892	0,09095
		8	5600	-25	0,0562	0,0795	0,07958
		9	6400	-26	0,0501	0,0709	0,07074
		10	7200	-28	0,0398	0,0563	0,06366

		11	8000	-	33,2	0,0219	0,0309	0,05787
--	--	----	------	---	------	--------	--------	---------



## GRAFICA MATLAB CON CODIGO

T=0.00125;

f=1/T;

t=0:1/1000000000:2\*T;

D=0.8;

A1=0.669;

A2=0.324;

A3=0.234;

A4=0.162;

A5=0.123;

A6=0.100;

A7=0.089;

A8=0.079;

A9=0.070;

A10=0.056;

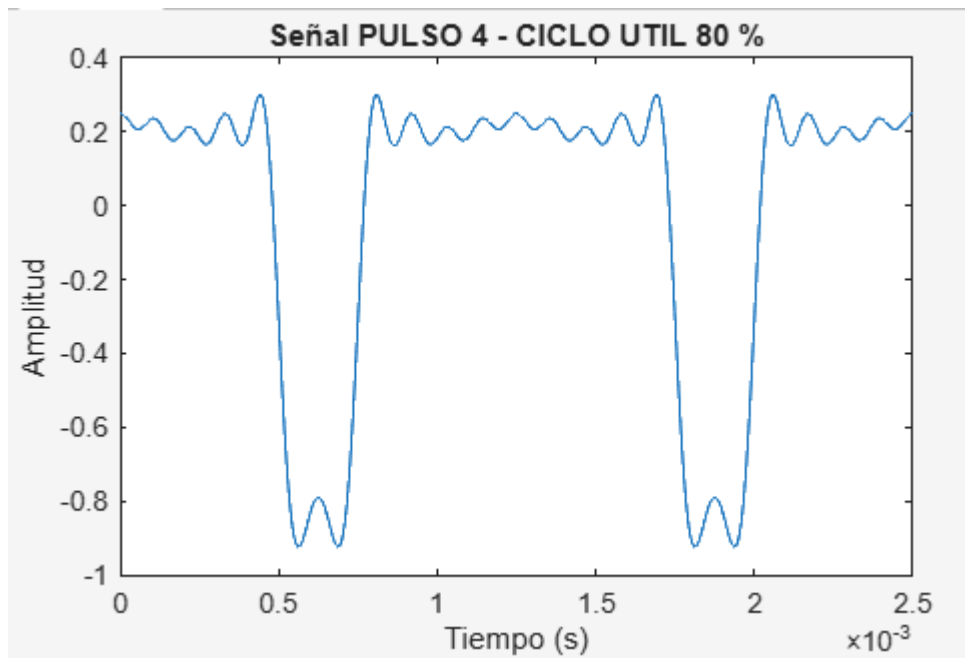
A11=0.050;

v1=A1\*sin(1\*pi\*D)\*cos(2\*pi\*1\*f\*t);

```

v2=A2*sin(2*pi*D)*cos(2*pi*2*f*t);
v3=A3*sin(3*pi*D)*cos(2*pi*3*f*t);
v4=A4*sin(4*pi*D)*cos(2*pi*4*f*t);
v5=A5*sin(5*pi*D)*cos(2*pi*5*f*t);
v6=A6*sin(6*pi*D)*cos(2*pi*6*f*t);
v7=A7*sin(7*pi*D)*cos(2*pi*7*f*t);
v8=A8*sin(8*pi*D)*cos(2*pi*8*f*t);
v9=A9*sin(9*pi*D)*cos(2*pi*9*f*t);
v10=A10*sin(10*pi*D)*cos(2*pi*10*f*t);
v11=A11*sin(11*pi*D)*cos(2*pi*11*f*t);
VT=(v1+v2+v3+v4+v5+v6+v7+v8+v9+v10+v11);
plot(t,VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO - CICLO UTIL 30 %');

```



**GRAFICA MATLAB CON DATOS EXPORTADO**

```

% Leer el archivo como tabla, empezando desde la fila 18

opts = detectImportOptions('DOMINIOTIEMPOPULSO4.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

% Leer los datos

datos = readtable('DOMINIOTIEMPOPULSO4.csv', opts);

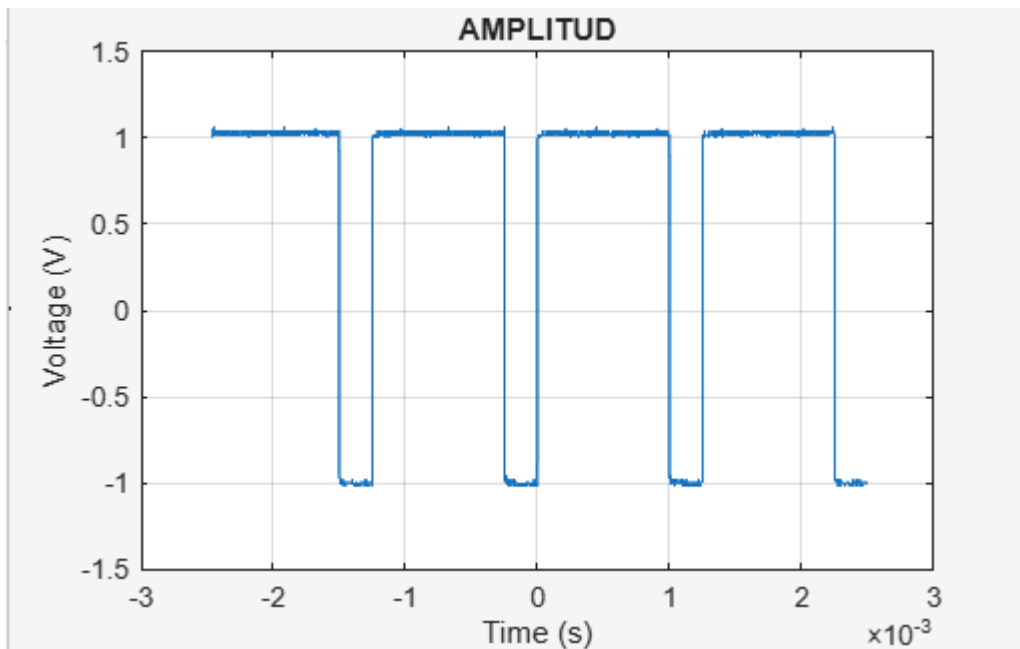
% Convertir las columnas correctas (4 y 5) a vectores

t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje

% Graficar

plot(t, v);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('AMPLITUD');
grid on;

```



## DOMINIO FRECUENCIA

```

% Leer el archivo como tabla, empezando desde la fila 18

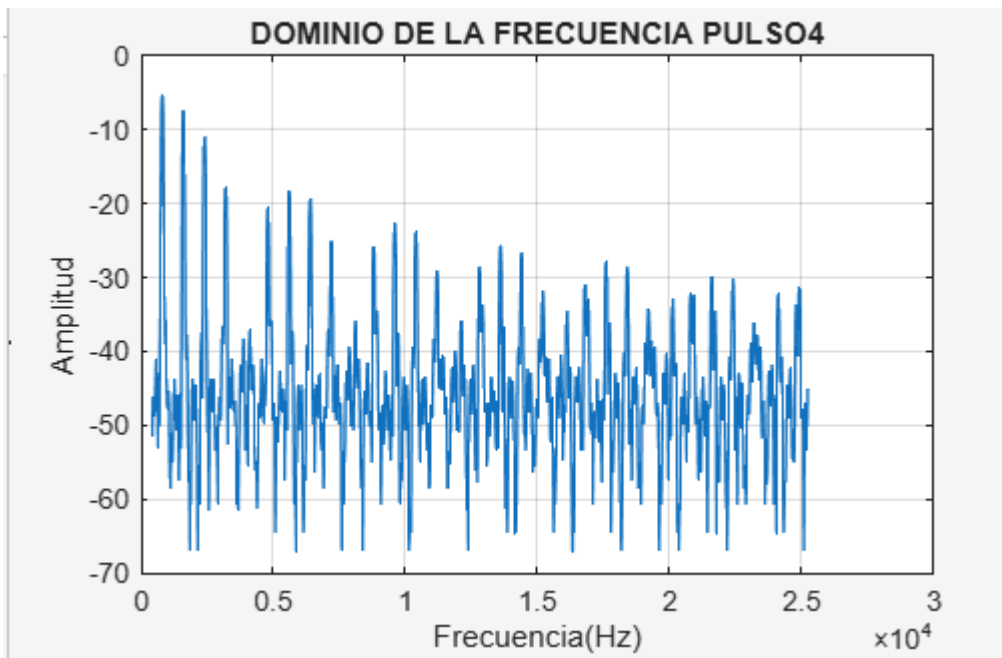
```

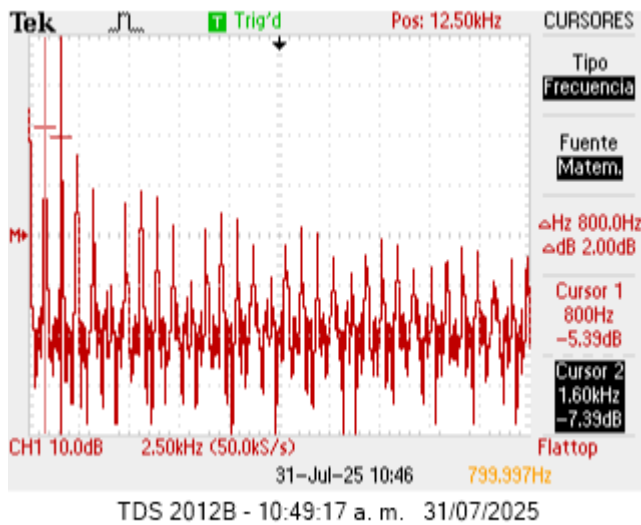


```

opts = detectImportOptions('DOMINIOFRECUENCIAPULSO4.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18
% Leer los datos
datos = readtable('DOMINIOFRECUENCIAPULSO4.csv', opts);
% Convertir las columnas correctas (4 y 5) a vectores
t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje
% Graficar
plot(t, v);
xlabel('Frecuencia(Hz)');
ylabel('Amplitud');
title('DOMINIO DE LA FRECUENCIA PULSO4');
grid on;

```





### GRAFICA FFT (TEORICA)

% === Señal PULSO 4 - CICLO ÚTIL 80% ===

T = 0.00125; % Periodo (s)

f = 1/T; % Frecuencia fundamental (Hz) = 800

Fs = 1e9; % Frecuencia de muestreo (1 GHz)

t = 0:1/Fs:2\*T; % Tiempo total de 2 ciclos

% === Ciclo útil (Duty cycle) ===

D = 0.8;

% === Amplitudes por armónico ===

A1 = 0.669; A2 = 0.324; A3 = 0.234; A4 = 0.162; A5 = 0.123;

A6 = 0.100; A7 = 0.089; A8 = 0.079; A9 = 0.070; A10 = 0.056;

A11 = 0.050;

% === Construcción de la señal compuesta ===

v1 = A1 \* sin(1\*pi\*D) \* cos(2\*pi\*1\*f\*t);

v2 = A2 \* sin(2\*pi\*D) \* cos(2\*pi\*2\*f\*t);

```

v3 = A3 * sin(3*pi*D) * cos(2*pi*3*f*t);
v4 = A4 * sin(4*pi*D) * cos(2*pi*4*f*t);
v5 = A5 * sin(5*pi*D) * cos(2*pi*5*f*t);
v6 = A6 * sin(6*pi*D) * cos(2*pi*6*f*t);
v7 = A7 * sin(7*pi*D) * cos(2*pi*7*f*t);
v8 = A8 * sin(8*pi*D) * cos(2*pi*8*f*t);
v9 = A9 * sin(9*pi*D) * cos(2*pi*9*f*t);
v10 = A10 * sin(10*pi*D) * cos(2*pi*10*f*t);
v11 = A11 * sin(11*pi*D) * cos(2*pi*11*f*t);

```

```

% === Señal total ===

```

```

VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;

```

```

% === Gráfica en el dominio del tiempo ===

```

```

figure;
plot(t, VT, 'b');
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO 4 - CICLO ÚTIL 80%');
grid on;

```

```

% === FFT de la señal ===

```

```

N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

```

```

% === Conversión a dB ===

```

```

Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

```

```
% === Eje de frecuencias ===
```

```
f_axis = (0:N-1)*(Fs/N);
```

```
% === Gráfica del espectro en dB ===
```

```
figure;
```

```
plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.3 0 0.7], 'LineWidth', 1.5); % púrpura oscuro
```

```
xlabel('Frecuencia (Hz)');
```

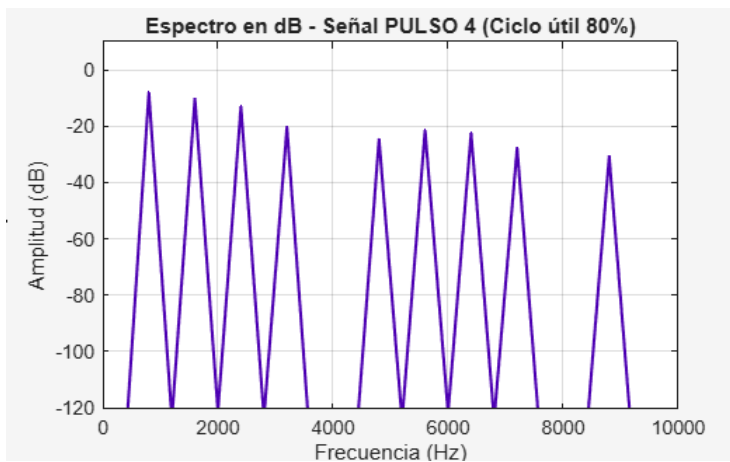
```
ylabel('Amplitud (dB)');
```

```
title('Espectro en dB - Señal PULSO 4 (Ciclo útil 80%)');
```

```
grid on;
```

```
xlim([0 10000]); % Visualizar hasta 10 kHz
```

```
ylim([-120 10]); % Escala típica en dB
```



## GRAFICA IFFT

```
% === Señal PULSO 4 - CICLO ÚTIL 80% ===
```

```
T = 0.00125; % Periodo (s)
```

```
f = 1/T; % Frecuencia fundamental (Hz) = 800
```

```
Fs = 1e9; % Frecuencia de muestreo (1 GHz)
```

```
t = 0:1/Fs:2*T; % Tiempo total de 2 ciclos
```

% === Ciclo útil (Duty cycle) ===

D = 0.8;

% === Amplitudes por armónico ===

A1 = 0.669; A2 = 0.324; A3 = 0.234; A4 = 0.162; A5 = 0.123;

A6 = 0.100; A7 = 0.089; A8 = 0.079; A9 = 0.070; A10 = 0.056;

A11 = 0.050;

% === Construcción de la señal compuesta ===

v1 = A1 \* sin(1\*pi\*D) \* cos(2\*pi\*1\*f\*t);

v2 = A2 \* sin(2\*pi\*D) \* cos(2\*pi\*2\*f\*t);

v3 = A3 \* sin(3\*pi\*D) \* cos(2\*pi\*3\*f\*t);

v4 = A4 \* sin(4\*pi\*D) \* cos(2\*pi\*4\*f\*t);

v5 = A5 \* sin(5\*pi\*D) \* cos(2\*pi\*5\*f\*t);

v6 = A6 \* sin(6\*pi\*D) \* cos(2\*pi\*6\*f\*t);

v7 = A7 \* sin(7\*pi\*D) \* cos(2\*pi\*7\*f\*t);

v8 = A8 \* sin(8\*pi\*D) \* cos(2\*pi\*8\*f\*t);

v9 = A9 \* sin(9\*pi\*D) \* cos(2\*pi\*9\*f\*t);

v10 = A10 \* sin(10\*pi\*D) \* cos(2\*pi\*10\*f\*t);

v11 = A11 \* sin(11\*pi\*D) \* cos(2\*pi\*11\*f\*t);

% === Señal total ===

VT = v1 + v2 + v3 + v4 + v5 + v6 + v7 + v8 + v9 + v10 + v11;

% === Gráfica en el dominio del tiempo ===

figure;

plot(t, VT, 'b');

xlabel('Tiempo (s)');

ylabel('Amplitud');

```

title('Señal PULSO 4 - CICLO ÚTIL 80%');

grid on;

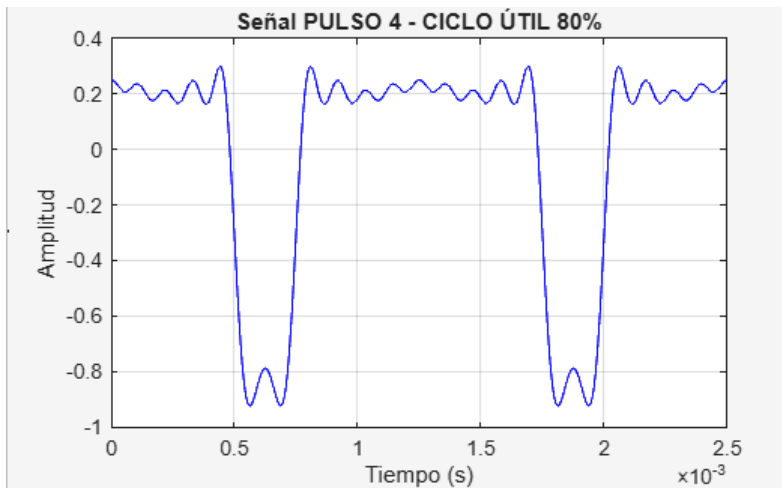
% === FFT de la señal ===
N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar amplitudes (excepto DC y Nyquist)

% === Conversión a dB ===
Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === Eje de frecuencias ===
f_axis = (0:N-1)*(Fs/N);

% === Gráfica del espectro en dB ===
figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'Color', [0.3 0 0.7], 'LineWidth', 1.5); % púrpura oscuro
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
title('Espectro en dB - Señal PULSO 4 (Ciclo útil 80%)');
grid on;
xlim([0 10000]); % Visualizar hasta 10 kHz
ylim([-120 10]); % Escala típica en dB

```



## PHYTON

```
import pandas as pd

import matplotlib.pyplot as plt

# Ruta al archivo

archivo = 'DOMINIOTIEMPOPOPULSO4.csv'

# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1

datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')

# Extraer columnas de interés

tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)

voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)

# Graficar

plt.plot(tiempo, voltaje, color='#FFFF00')

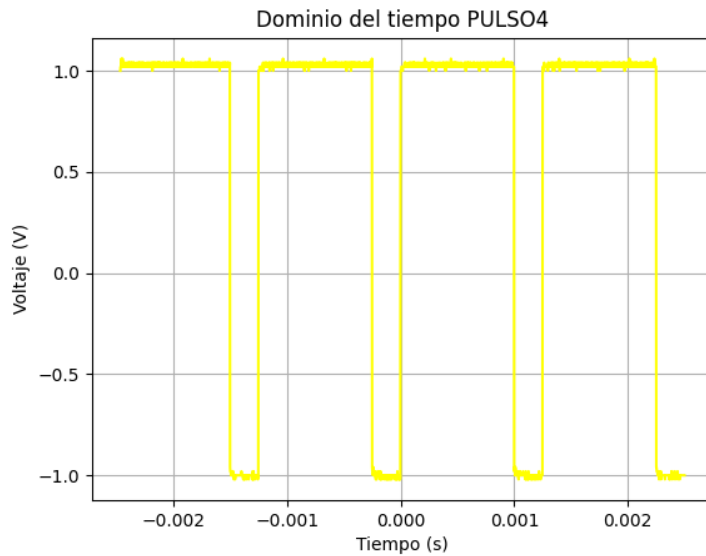
plt.title('Dominio del tiempo PULSO4')

plt.xlabel('Tiempo (s)')

plt.ylabel('Voltaje (V)')

plt.grid(True)
```

```
plt.show()
```



## FRECUENCIA

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
# Ruta al archivo
```

```
archivo = 'DOMINIOFRECUENCIAPULSO4.csv'
```

```
# Leer archivo desde fila 18, sin encabezado, con codificación Latin-1
```

```
datos = pd.read_csv(archivo, skiprows=17, header=None, encoding='latin1')
```

```
# Extraer columnas de interés
```

```
tiempo = datos.iloc[:, 3] # Columna 4 en MATLAB (índice 3)
```

```
voltaje = datos.iloc[:, 4] # Columna 5 en MATLAB (índice 4)
```

```
# Graficar
```

```
plt.plot(tiempo, voltaje, color='#FFFF00')
```

```
plt.title('Dominio del tiempo PULSO4')
```



```
plt.xlabel('Frecuencia')
```

```
plt.ylabel('Amplitud')
```

```
plt.grid(True)
```

```
plt.show()
```

