

# CONCEPTOS DE PROCESAMIENTO DE SEÑALES PARA COMUNICACIONES

Angie Ximena Diaz Bejarano  
[est.angie.xdiaz@unimilitar.edu.co](mailto:est.angie.xdiaz@unimilitar.edu.co)  
 Docente: José De Jesús Rugeles

**Resumen:** Durante el desarrollo del laboratorio 2 se realizaron adquisiciones de señales periódicas con distintas formas (senoidales, triangulares, cuadradas y pulsos), las cuales fueron procesadas para obtener sus espectros en frecuencia mediante la FFT. Posteriormente, se aplicó la IFFT para reconstruir las señales en el dominio del tiempo y verificar su fidelidad respecto a las originales. Se utilizaron herramientas como MATLAB y Thonny con Python para visualizar y analizar los resultados, y se trabajó con archivos en formato CSV exportados desde el osciloscopio. El análisis permitió corroborar la frecuencia fundamental de cada señal, evidenciar la presencia de armónicos en las formas no senoidales, y entender la importancia del muestreo adecuado. El trabajo se complementó con la validación gráfica de los espectros y las reconstrucciones, obteniendo resultados consistentes con la teoría del análisis de Fourier..

## INTRODUCCIÓN

El presente laboratorio tuvo como objetivo fundamental el análisis de señales periódicas en el dominio del tiempo y su correspondiente transformación al dominio de la frecuencia, mediante la aplicación de la Transformada Rápida de Fourier (FFT) e Inversa (IFFT). Se trabajó con diversas formas de onda generadas experimentalmente, como senoidales, triangulares, cuadradas y pulsos, las cuales fueron capturadas mediante un osciloscopio digital y exportadas en formato CSV para su posterior procesamiento en MATLAB y Python. A partir de los datos adquiridos, se construyeron gráficas de espectros de frecuencia y señales reconstruidas, evaluando su correspondencia con las señales originales y analizando su comportamiento en función del periodo y la frecuencia fundamental. Este proceso permitió observar empíricamente la equivalencia entre el análisis temporal y espectral de una señal, reforzando conceptos fundamentales en el procesamiento digital de señales.

## DESARROLLO EXPERIMENTO

Se inicia la practica con el osciloscopio y el generador de señales con el propósito de configurarlos y poder mostrar una a una de las señales solicitadas, dos senos, dos triangulares, dos cuadradas y 4 pulsos. Se selecciona el grupo con el cual se trabajará y este es el 1, el cual tiene designada la frecuencia de 800Hz con un VP de 1V, se hace consecutivamente la toma de datos iniciando con la onda seno1, tener en cuenta que se activara la función MATH, FFT en el osciloscopio y se almacenaran los resultados en formato csv.

## 1. SEÑAL SENO 1

Para el inicio de esta señal se configura el osciloscopio y el generador de señales lo cual nos permite observar una señal periódica seno como se muestra en la ilustración 1.

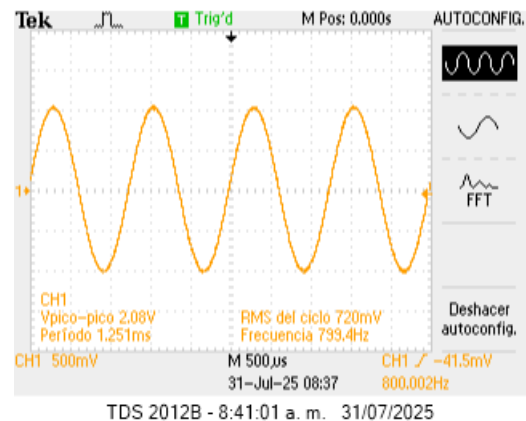


Ilustración 1. Señal seno en osciloscopio

Teniendo en cuenta esta señal se toman los siguientes datos

FRECUENCIA	800HZ
VOLTAJE PICO	1V

DC	0
ARMONICA	1
FRECUENCIA(Hz)	800
dB	-2,99
Vrms	0,7087
Vp (1)	1,0023

Estos datos se obtuvieron teniendo el osciloscopio en FFT a lo que se obtuvo una sola armónica, los valores que se observaron en el osciloscopio están en dBs por lo que se aplica la siguiente formula

$$V(\text{en voltios}) = V(\text{referencia} \times 10^{\frac{dB}{20}})$$

$$V(\text{en voltios}) = 1v \times 10^{\frac{-2,99}{20}} = 0,708v$$

Pero lo necesitamos en voltaje pico por lo que se multiplicara por raíz de dos, por lo que el resultado es:

$$V_{pico} = V_{RMS} \times \sqrt{2}$$

$$V_{pico} = 0,708V \times \sqrt{2} = 1,002V$$

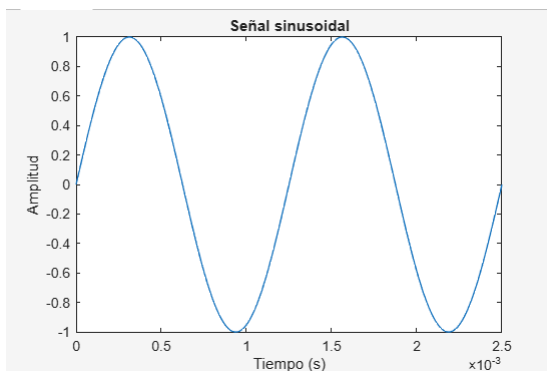
Con estas formulas se obtuvo la tabla anteriormente mostrada por lo que se usara Excel como método de calculo para facilitar

los datos, ahora bien, teniendo estos datos se procede a graficar en Matlab los datos teóricos, usando la formula de una señal sinusoidal en el dominio del tiempo. Tener en cuenta que en esta onda no se hará grafica de datos medidos ya que solo tiene una armónica y esta será la misma mostrada en la ilustración 2, ya que los datos son los mismos.

$$v(t)=A \cdot \sin(2\pi ft + \phi)$$

Entonces se inicia con la gráfica teórica:

```
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');
```

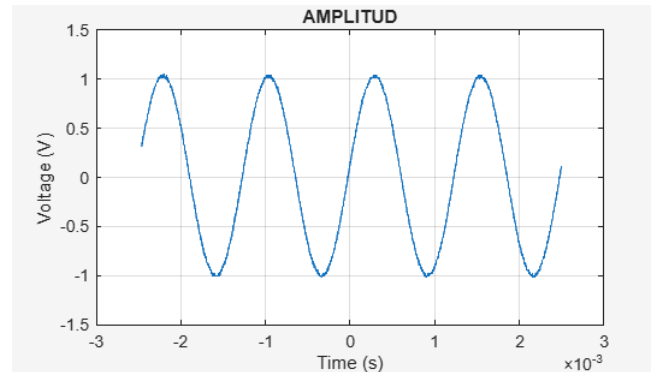


**Ilustración 2. Señal Seno1 datos teóricos**

Por consiguiente, se procede a graficar la señal seno, pero esta vez exportando los datos directamente del csv que se guardo anteriormente, por lo que se usa el siguiente código en Matlab:

```
opts = detectImportOptions('DOMINIOTIEMPO1.csv');
opts.DataLines = [18 Inf];
opts.Delimiter = ',';
opts.VariableNamesLine = 0;
datos = readtable('DOMINIOTIEMPO1.csv', opts);
t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje
plot(t, v);
xlabel('Time (s)');
```

```
ylabel('Voltage (V)');
title('AMPLITUD');
grid on;
```



**Ilustración 3. Señal seno1 con datos exportados**

Esta misma señal se graficará, pero en Python:

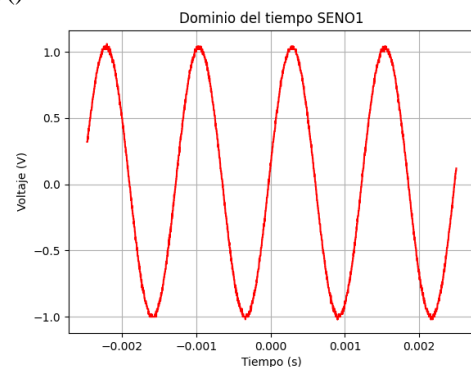
```
import pandas as pd
import matplotlib.pyplot as plt

# 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)

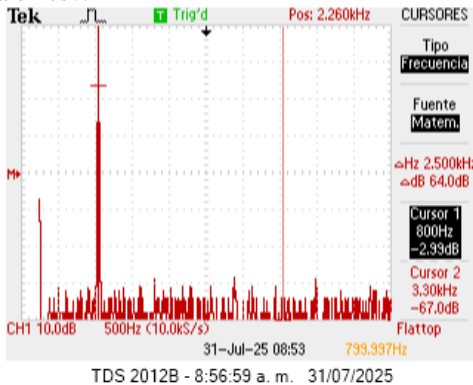
# Graficar
plt.plot(tiempo, voltaje, color='red')
plt.title('Dominio del tiempo SEN01')
plt.xlabel('Tiempo (s)')
plt.ylabel('Voltaje (V)')
plt.grid(True)
plt.show()
```



**Ilustración 4. Señal seno1 en python Thonny**

Ahora bien, es bueno preguntarse ¿En que cambia cada gráfica? Ya que una fue hecha teóricamente por medio de formula y otra se exporto directamente del osciloscopio, realmente el cambio no es mucho, además de que en el

osciloscopio se observa un poco de ruido a la señal y en el programa es pura totalmente, que en el tiempo total en la gráfica teórica se observan 2 ciclos y en la experimental más de 4 ciclos y que pues en una se observa el realismo ideal y el otro el real, ahora bien, ¿Cuál es la tasa de muestreo del osciloscopio utilizado? Se usó un osciloscopio Tektronix TDS2012B, el cual tiene banda ancha de 100 MHz y una tasa de muestreo en tiempo real de 1.000.000.000 muestras/segundo, por lo que en los archivos csv que se exportan tienen un intervalo de muestras de  $2\mu s = 1.9999999 \times 10^{-6}$  s por lo que son 500.000 muestras por segundo (500Hz), lo cual es suficiente para capturar señales de hasta 250kHz sin aliasing. En este caso la señal es de 800 Hz, por lo que se está muestreando 625 veces por ciclo ( $500.000 / 800$ ). Por otro lado, tenemos la señal seno1 en el dominio de la frecuencia tomada del osciloscopio y de la misma manera exportada en csv.



**Ilustración 5. Señal sen1 en el dominio de la frecuencia**

En la tabla se observa que ya hay un valor de dB el cual se puede evidenciar en la ilustración 5, por lo que ahora se grafica la misma señal en Matlab:

```
opts =
detectImportOptions('DOMINIOFRECUENCIASEN1.csv');

opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma

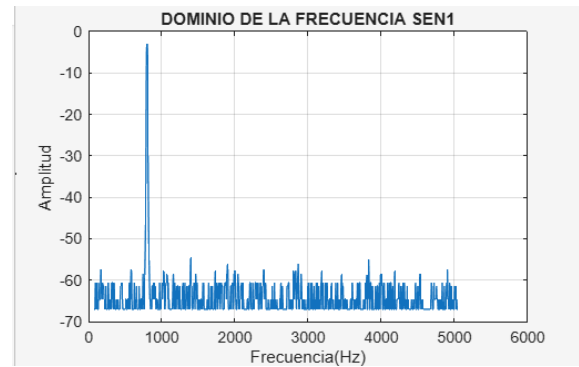
opts.VariableNamesLine = 0; % No hay nombres de variables
en la fila 18

% Leer los datos
datos = readtable('DOMINIOFRECUENCIASEN1.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;



**Ilustración 6. Señal seno1 en Matlab**

Teniendo esta grafica se procede a hacer lo mismo, pero en python:

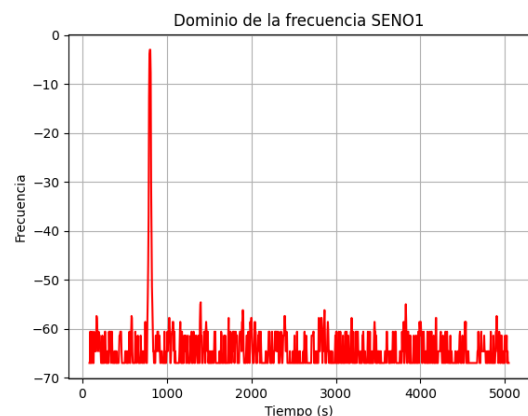
```
import pandas as pd
import matplotlib.pyplot as plt

# 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()
```



**Ilustración 7. Señal sen1 en Python Thonny**

Haciendo el análisis general de la señal del osciloscopio y la de Matlab. Ambas gráficas confirman que la señal medida es una senoidal pura de 800 Hz, como fue planteado en la teoría. El ruido de fondo es bajo en ambas mediciones ( $\sim -65$  dB), lo cual indica buena calidad de adquisición. No se detectan armónicos significativos, lo cual es consistente con una señal senoidal sin

distorsión. El pico en 3.3 kHz en la segunda gráfica es probablemente ruido o interferencia del entorno (muy bajo en comparación al fundamental). La FFT en MATLAB te da más control sobre el análisis (ventanas, resolución, zoom, filtrado), pero la FFT del osciloscopio es inmediata y práctica.

Teniendo esto en cuenta se procede a graficar los valores que se tomaron del tiempo para hacer la transformada de Fourier FFT.

$T = 0.00125$ ;

$f = 1/T$ ;

$F_s = 1e5$ ;

$t = 0:1/F_s:2*T$ ;

$A_1 = 1$ ;

% Generar señal senoidal

$v_1 = A_1 * \sin(2*\pi*f*t)$ ;

% Cálculo de la FFT

$N = \text{length}(v_1)$ ;

$Y = \text{fft}(v_1)$ ; % Transformada rápida de Fourier

$Y\_mag = \text{abs}(Y)/N$ ;

$Y\_mag(2:\text{end}-1) = 2*Y\_mag(2:\text{end}-1)$ ;

% Escala de frecuencia

$f\_axis = (0:N-1)*(F_s/N)$ ; % Vector de frecuencias asociado a la FFT

% Convertir magnitud a decibelios

$Y\_dB = 20*\log_{10}(Y\_mag + \text{eps})$ ; % eps evita log(0) y errores numéricos

% Graficar el espectro (solo la mitad positiva)

$\text{plot}(f\_axis(1:N/2), Y\_dB(1:N/2), 'r', 'LineWidth', 1.5)$ ;

$\text{xlabel}('Frecuencia (Hz)')$ ;

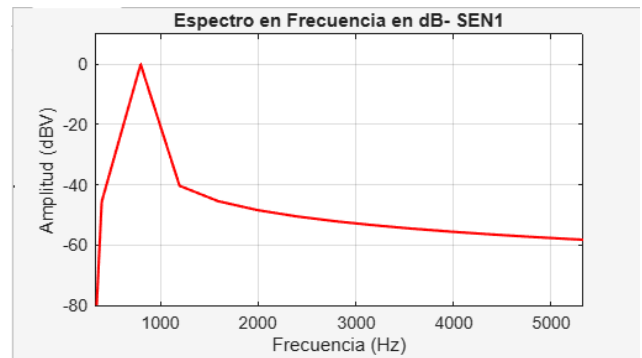
$\text{ylabel}('Amplitud (dBV)')$ ;

$\text{title}('Espectro en Frecuencia en dB- SEN1')$ ;

$\text{grid on}$ ;

$\text{xlim}([0 \ 5000])$ ;

$\text{ylim}([-80 \ 10])$ ;



**Ilustración 8. Señal seno1 con FFT**

Entonces se observa que a partir de la generación de la señal senoidal pura en el tiempo se aplica la transformada de Fourier usando la función `fft()` que convierte la señal del tiempo al espectro de frecuencias. Se calcula la magnitud de cada componente de frecuencia con la función  $Y\_mag = \text{abs}(Y)/N$ ; y  $Y\_mag(2:\text{end}-1) = 2*Y\_mag(2:\text{end}-1)$ ; y bueno, finalmente se convierte la magnitud en dBs y se grafica. Recordemos que es importante verificar si una señal tiene armonía o distorsión, también la FFT ayuda a detectar frecuencias ocultas.

Ahora bien, se tiene en cuenta que la FFT es una transformación compleja, se requiere de magnitud y fase para reconstruir completamente la señal, se está solicitando reconstruirla con el csv pero este solo contiene magnitud en dB pero no tiene la parte imaginaria ni la fase, por lo que no es posible graficar IFFT directamente desde la tabla, por lo que se tomará el código anterior mostrado y se reconstruirá la FFT en el dominio del tiempo.

% Parámetros de la señal SEN1

$T = 0.00125$ ;

$f = 1/T$ ;

$F_s = 1e5$ ;

$t = 0:1/F_s:2*T$ ;

$A_1 = 1$ ;

% Señal original (onda seno)

$v_1 = A_1 * \sin(2*\pi*f*t)$ ;

% FFT de la señal

$N = \text{length}(v_1)$ ; % Número de muestras

$Y = \text{fft}(v_1)$ ; % FFT de la señal

% Reconstrucción por IFFT

$v_{1\_rec} = \text{ifft}(Y, 'symmetric')$ ; % Reconstruir señal aplicando IFFT

% Gráfica final de la señal reconstruida

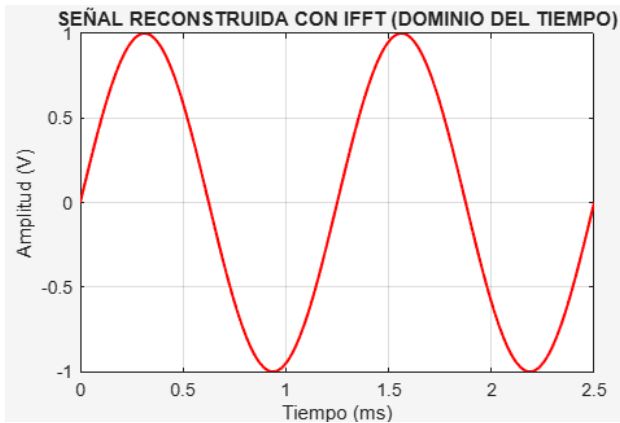
$t\_ms = t * 1000$ ; % Convertir a milisegundos para graficar

`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;

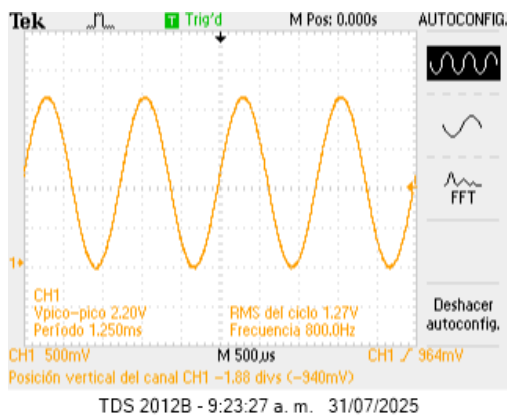
```



**Ilustración 9. Señal sen1 con IFFT**

## 2. SEÑAL SEN2 CON DC 1

En este caso conservamos los datos, pero habrá un DC 1 lo que significa que desfasará la señal 1V. Por lo que observamos lo siguiente en el osciloscopio:



Se recopilan los datos obtenidos en tablas:

FRECUENCIA	800HZ
VOLTAJE PICO	1V

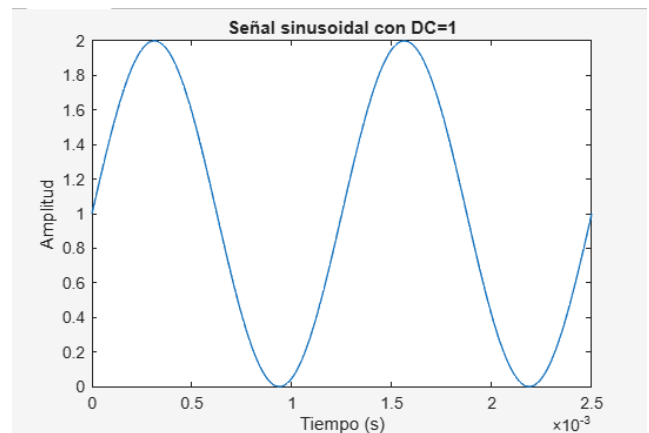
DC	0
ARMONICA	1
FRECUENCIA(Hz)	800
dB	-2,19
Vrms	0,77714
Vp (1)	1,09904

Por la anterior tabla ya sabemos cómo se obtuvieron estos valores por lo que se procede a graficar estos datos en Matlab:

```

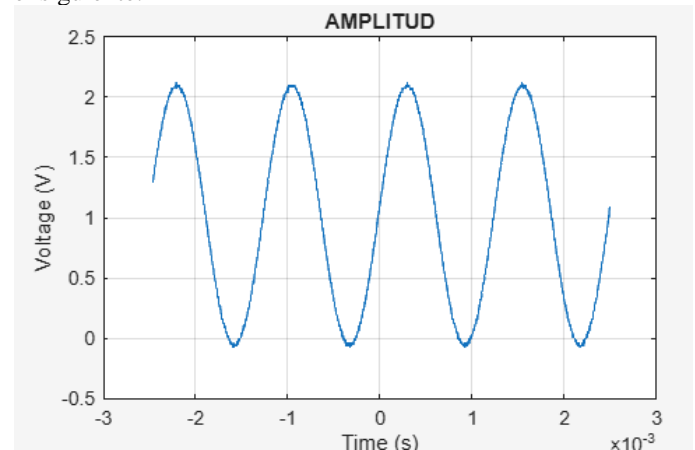
T=0.00125;
f=1/T;
t=0:1/100000:2*T;
A1=1;
v1=1+A1*sin(2*pi*f*t);
plot(t,v1)
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal sinusoidal con DC=1');

```



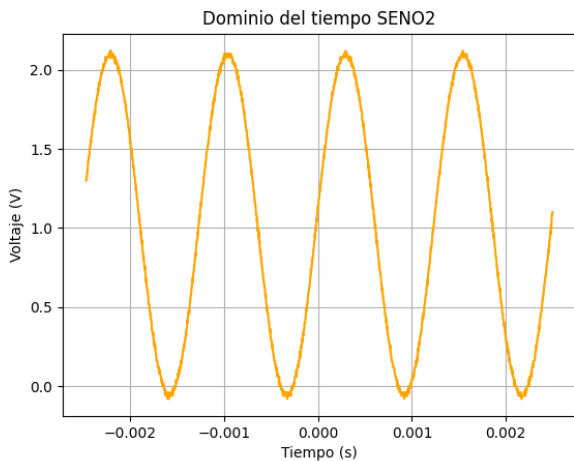
**Ilustración 10. Señal sen2 teórica**

Como se habían exportado los datos del osciloscopio a csv se procede a hacer la misma gráfica, pero con los datos exportados, por lo que se usa el anterior código para llamar al documento, pero esta vez en SEN2, por lo que el resultado es el siguiente:



**Ilustración 11. Señal sen2 datos exportados**

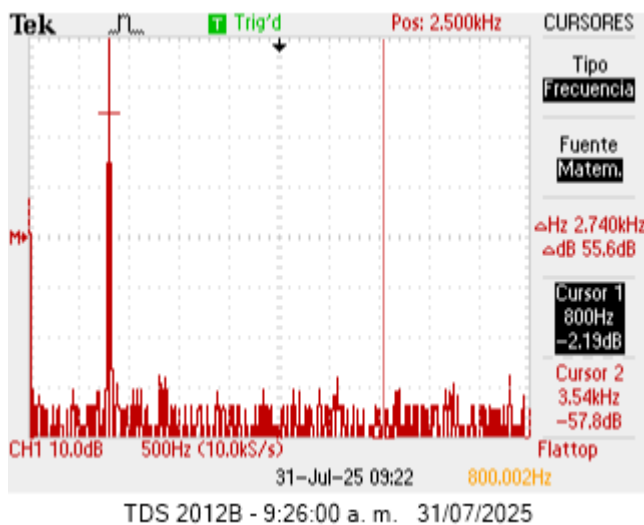
Teniendo estas dos graficas se procede a procesar la tercera pero esta vez en Python Thonny, ejecutando el código que anteriormente se mostró pero cambiando el archivo csv.



**Ilustración 12. Señal sen2 en Python Thonny**

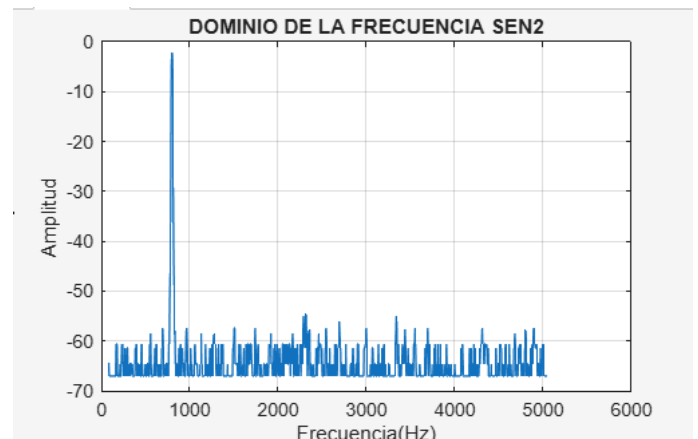
Se puede observar que las tres gráficas representan una señal senoidal de 800 Hz, con voltaje pico de 1 V y un componente de corriente continua (DC) de 1 V, obtenida desde diferentes medios. La primera gráfica, generada teóricamente en MATLAB, muestra una señal ideal, suave y perfectamente simétrica, desplazada verticalmente por el componente DC, oscilando entre 0 V y 2 V. La segunda gráfica proviene de datos reales exportados del osciloscopio en formato CSV, y aunque mantiene el mismo patrón de la señal, presenta ligeras variaciones en la forma de onda y en la amplitud, lo que puede atribuirse al ruido de medición y a la resolución del instrumento. La tercera gráfica, elaborada en Python con los mismos datos del osciloscopio, reproduce fielmente esa señal, mostrando también el desplazamiento DC, pero visualmente destaca por el color y estilo de línea más definidos. En comparación, la señal teórica es perfectamente limpia, mientras que las señales adquiridas muestran pequeñas imperfecciones propias del entorno real, lo cual permite validar la fidelidad del modelo simulado frente a la señal medida.

Ahora se toman los datos de la frecuencia, la cual en el osciloscopio se ve de esta manera con su valor en dB:



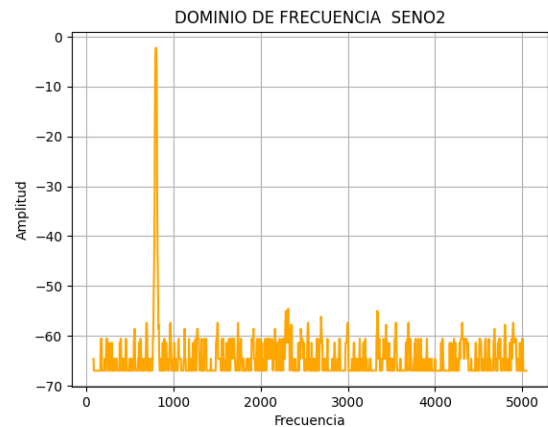
**Ilustración 13. Señal sen2 en el dominio de la frecuencia**

Se exporta el csv para poder graficar en Matlab usando el código que anteriormente se mostró en sen1 ya que se usa el mismo solo cambia el nombre del archivo:



**Ilustración 14. Señal se2 en Matlab**

Esta misma señal se graficará en Phytion Thonny usando de la misma manera el código de sen1 peor cambiando el nombre del archivo csv



**Ilustración 15. Señal sen2 en Phytion Thonny**

En las tres gráficas se observa claramente un pico dominante en los 800 Hz, correspondiente a la frecuencia fundamental de la señal, pero con diferencias notables en la forma y comportamiento del espectro. En la primera imagen, obtenida directamente del osciloscopio, se identifica además del pico principal una elevación más visible del ruido en frecuencias cercanas, y se destacan componentes menores como el cursor en 3.54 kHz a -57.8 dB, lo cual sugiere la presencia de ruido o armónicos débiles propios del entorno real de medición. En la segunda gráfica, realizada en MATLAB a partir de los datos exportados en CSV, el pico en 800 Hz es más definido y el resto del espectro presenta un ruido más disperso pero contenido entre -60 y -70 dB, con picos más agudos pero aislados, lo que refleja una medición más limpia pero aún influenciada por el entorno del osciloscopio. En contraste, la tercera gráfica, procesada en Python con los mismos datos, muestra una representación aún más estable: el pico en 800 Hz se mantiene claro y el nivel de ruido está más uniforme y plano, sin variaciones abruptas, lo que indica un mayor filtrado o suavizado en el procesamiento. Comparando las tres, el osciloscopio muestra un espectro más realista con imperfecciones visibles, MATLAB ofrece mayor resolución y



detalle en la dispersión del ruido, mientras que Python entrega una representación más estética y homogénea del mismo análisis.

Siguiendo con el orden se procede a graficar los datos obtenidos en el dominio del tiempo, pero aplicándole la FFT, que anteriormente ya mencionamos su uso y su importancia  
 $T = 0.00125$ ;      % Periodo (s)

$f = 800$ ;

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

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

$A_1 = 1.09904379$ ;      % Amplitud pico real

% Generar señal senoidal

$v_1 = A_1 * \sin(2*\pi*f*t)$ ;

% Cálculo de la FFT

$N = \text{length}(v_1)$ ;      % Número total de muestras

$Y = \text{fft}(v_1)$ ;      % Transformada rápida de Fourier

$Y_{\text{mag}} = \text{abs}(Y)/N$ ;      % Magnitud normalizada

$Y_{\text{mag}}(2:\text{end}-1) = 2*Y_{\text{mag}}(2:\text{end}-1)$ ; % Duplicar componentes (menos DC y Nyquist)

$f_{\text{axis}} = (0:N-1)*(F_s/N)$ ;      % Vector de frecuencias asociado a la FFT

$Y_{\text{dB}} = 20*\log_{10}(Y_{\text{mag}} + \text{eps})$ ;      % eps evita log(0) y errores numéricos

% Graficar el espectro (solo la mitad positiva)

$\text{plot}(f_{\text{axis}}(1:N/2), Y_{\text{dB}}(1:N/2), 'g', 'LineWidth', 1.5)$ ;

$\text{xlabel}(\text{'Frecuencia (Hz)'})$ ;

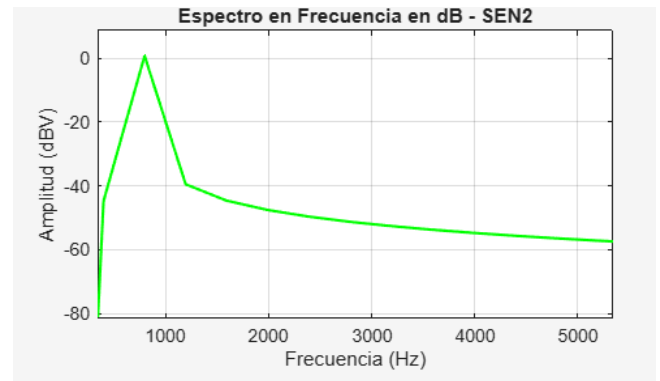
$\text{ylabel}(\text{'Amplitud (dBV)'})$ ;

$\text{title}(\text{'Espectro en Frecuencia en dB - SEN2'})$ ;

$\text{grid on}$ ;

$\text{xlim}([0 \ 5000])$ ;

$\text{ylim}([-80 \ 10])$ ;



**Ilustración 16. Señal sen2 con FFT**

Ahora bien, se tiene en cuenta que la FFT es una transformación compleja, se requiere de magnitud y fase para reconstruir completamente la señal, se está solicitando reconstruirla con el csv pero este solo contiene magnitud en dB pero no tiene la parte imaginaria ni la fase, por lo que no es posible graficar IFFT directamente desde la tabla, por lo que se tomará el código anterior mostrado y se reconstruirá la FFT en el dominio del tiempo.

% Señal SEN2

% === Parámetros de la señal ===

$T = 0.00125$ ;

$f = 800$ ;

$F_s = 1e5$ ;

$t = 0:1/F_s:2*T$ ;

$A_1 = 1.09904379$ ;

% === Generar señal senoidal ===

$v_1 = A_1 * \sin(2*\pi*f*t)$ ;

% === Cálculo de la FFT ===

$N = \text{length}(v_1)$ ;

$Y = \text{fft}(v_1)$ ;

$Y_{\text{mag}} = \text{abs}(Y)/N$

$Y_{\text{mag}}(2:\text{end}-1) = 2*Y_{\text{mag}}(2:\text{end}-1)$ ;

% === Escala de frecuencia ===

$f_{\text{axis}} = (0:N-1)*(F_s/N)$ ;

% === Convertir magnitud a decibelios ===

$Y_{\text{dB}} = 20*\log_{10}(Y_{\text{mag}} + \text{eps})$ ;

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

$\text{figure}$ ;

$\text{plot}(f_{\text{axis}}(1:N/2), Y_{\text{dB}}(1:N/2), 'g', 'LineWidth', 1.5)$ ;

$\text{xlabel}(\text{'Frecuencia (Hz)'})$ ;

```
ylabel('Amplitud (dBV)');
title('Espectro en Frecuencia en dB - SEN2');
grid on;
xlim([0 5000]);
ylim([-80 10]);
v1_rec = ifft(Y, 'symmetric');
t_ms = t * 1000;
% === 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]);
```

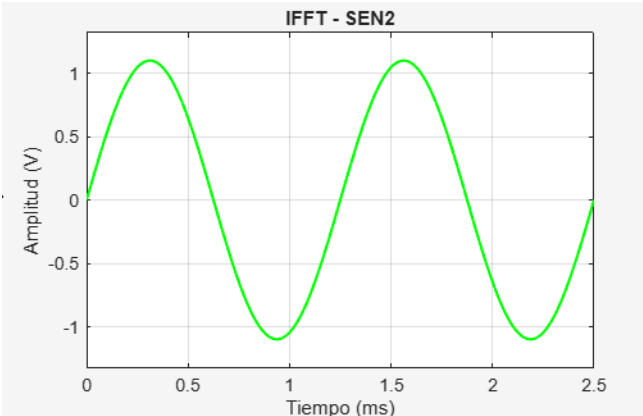


Ilustración 17. Señal sen2 con IFFT

Se observa que en la sen1 y sen2 la diferencia es el Dc, lo cual que entre una señal senoidal con componente DC de 1 V y una sin componente DC (es decir, con DC = 0) radica en el **desplazamiento vertical** de toda la señal sobre el eje del tiempo. Cuando una señal tiene DC = 0, oscila de manera simétrica alrededor del eje horizontal (cero voltios), es decir, sus valores positivos y negativos están equilibrados, y su valor promedio en el tiempo es cero. En cambio, al agregar un componente DC de 1 V, toda la señal se **eleva** verticalmente, haciendo que su valor mínimo y máximo aumenten en 1 V, y que la onda ya no oscile alrededor del cero, sino alrededor de 1 V. Esto implica que su valor medio ya no es cero, sino 1 V.

3. SEÑAL TRIANGULAR 1

FRECUENCIA	800HZ
------------	-------

VOLTAJE PICO	1V
DC	0

AMRONICA	1	3	5	7	9	11
FRECUENCIA	800	2400	4000	5600	7200	8800
dB	-	-23,8	-	-38.6	-43	-47
	4,59		32,6			
Vrms	0,58	0,06	0,02	0,011	0,0071	0,0045
Voltaje Pico	0,83	0,091	0,33	0,016	0,010	0,006
Teórico	0,8	0,09	0,03	0,01	0,01	0,0006

La ecuacion que se uso para los coeficientes de de la serie de Foerier, la cual es para armonicos impares , sirve para reconstruir una señal periodica compleja a partir de la suma de senos y cosenos con diferentes frecuencias y amplitudes. En este caso se usa:

$$A_n = \frac{8V}{n^2\pi^2}$$

$$x(t) = a_0 + \sum_{n=1,3,5,...}^{\infty} a_n \cos\left(\frac{2\pi n t}{T}\right)$$

Se remplazan valores y se ubican en la tabla previamnete puesta. Ahora se siguen los pasos de observar la señal en el osciloscopio

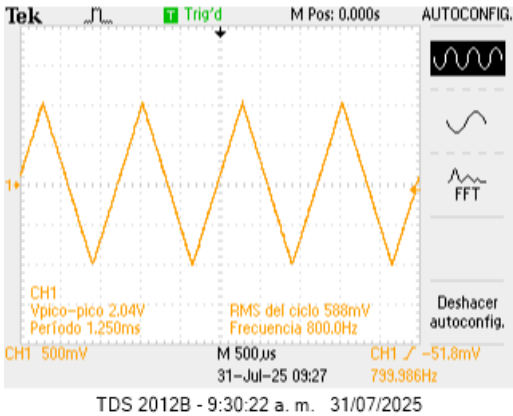


Ilustración 18. Señal Tri1 en osciloscopio

Se grafican los datos teoricamente en Matlab a partir de

$$v(t)=A\cdot\cos\cdot (2\pi ft)$$

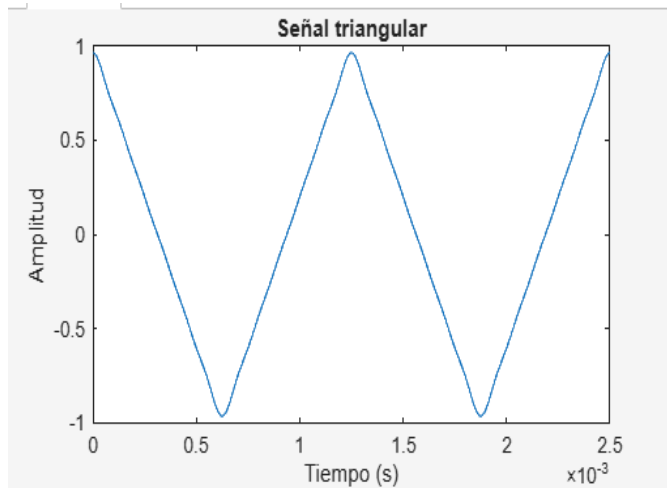
```
T=0.00125;
f=1/T;
t=0:1/100000000:2*T;
A1=0.810;
A3=0.090;
A5=0.032;
A7=0.017;
```



```

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');

```



**Ilustración 19. Señal Tri1 teórica Matlab**

Ahora se toman los datos medidos y se grafican, los datos medidos son los voltajes picos que se muestran en la tabla.  
 $T=0.00125$ ;

```

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);

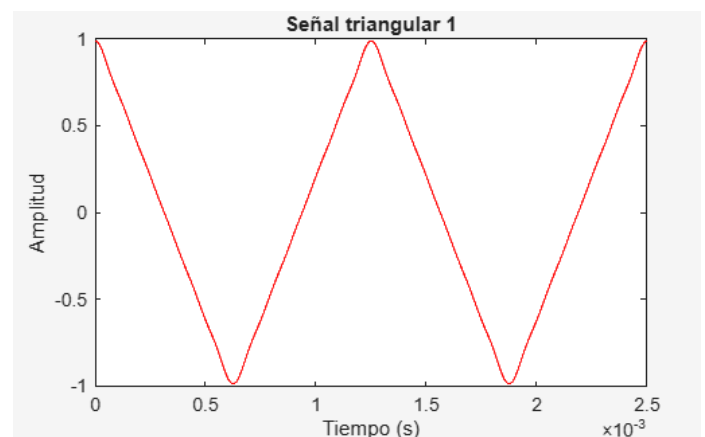
```

```

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');

```



**Ilustración 20. Señal Tri1 con valores medidos Matlab**

Ahora se exportan los datos por el csv en Matlab:

```

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);

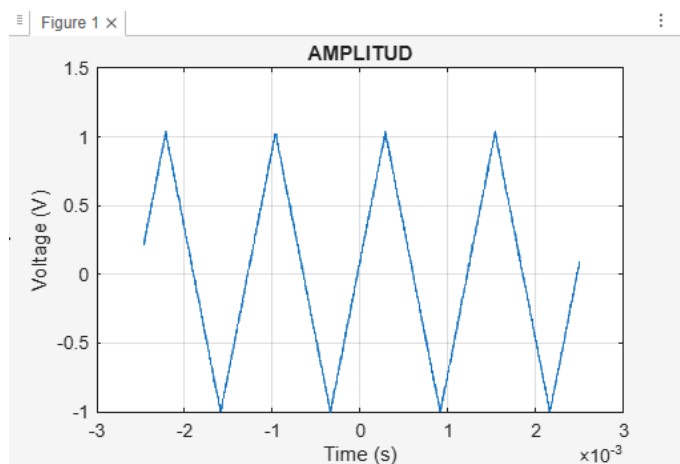
% Convertir las columnas correctas (4 y 5) a vectores
t = datos{:,4}; % Columna 4 = Tiempo
v = datos{:,5}; % Columna 5 = Voltaje

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

```

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

```
grid on;
```



**Ilustración 21. Señal Tri1 con datos exportados en Matlab**

Y por ultimo en el dominio del tiempo se grafica en python:

```
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')
```

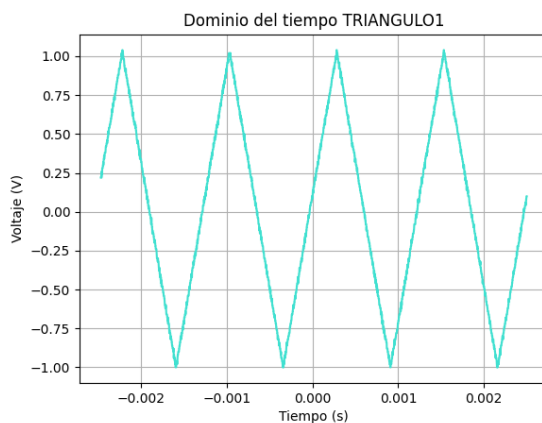
```
plt.title('Dominio del tiempo TRIANGULO1')
```

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

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

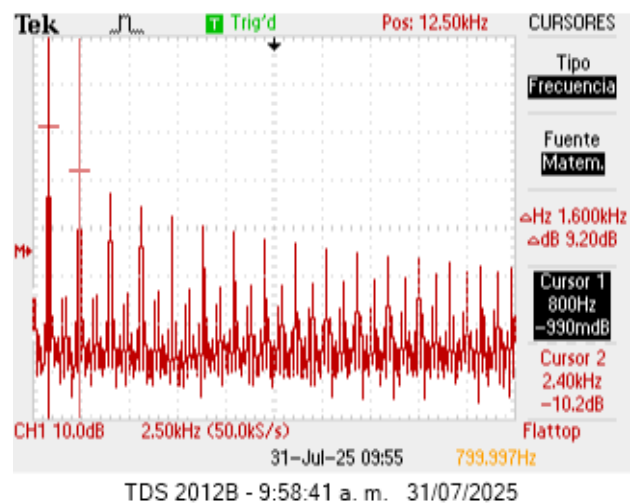
```
plt.grid(True)
```

```
plt.show()
```



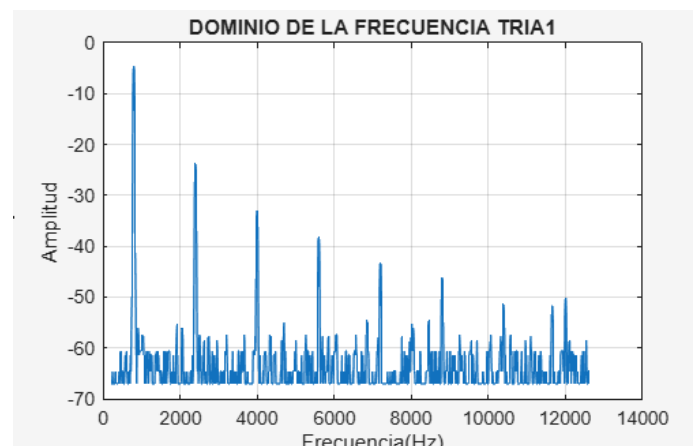
**Ilustración 22. Señal Tri1 en Python**

La primera imagen corresponde a la captura directa del osciloscopio, donde la señal aparece con un trazado más grueso y menos definido, producto del muestreo real y del ruido del entorno, además de mostrar un offset que desplaza la señal hacia el lado positivo. Las siguientes gráficas fueron generadas en MATLAB y Python, donde se observa una señal triangular más precisa, centrada en cero, con líneas limpias y simetría perfecta, lo que refleja una construcción ideal sin distorsión ni ruido. Sin embargo, dentro de estas, hay diferencias visuales: algunas tienen líneas más suaves, otras más segmentadas o con bordes marcados, dependiendo de la herramienta utilizada (por ejemplo, Python muestra una señal con puntos más agudos y cambio más abrupto de pendiente). En resumen, la señal del osciloscopio representa una medición real con imperfecciones, mientras que las demás son simulaciones limpias, diferenciándose por su entorno de generación y estilo gráfico. Ahora se toman los datos de la frecuencia, la cual en el osciloscopio se ve de esta manera con su valor en dB:

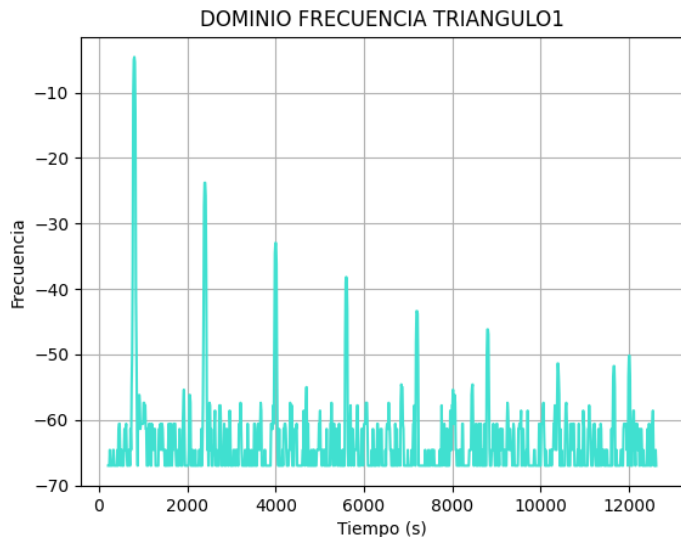


**Ilustración 23. Señal Tri1 en el dominio de la frecuencia osciloscopio**

Se procesan los datos en archivo csv y se grafica en matlab usando el código anteriormente proporcionado en la señal triangular para llamar archivos cuyo resultado es:



**Ilustración 24 señal tria1 en Dominio de frecuencia**



**Ilustración 25. Señal Tri1 dominio de la frecuencia Thonny**

En las tres gráficas se analiza la misma señal triangular en el dominio de la frecuencia, y se observa una distribución típica de **armónicos impares** decrecientes. La primera gráfica, tomada del osciloscopio, muestra una serie de picos espaciados regularmente, con el fundamental en 800 Hz y armónicos en múltiplos impares (1600 Hz, 2400 Hz, etc.), con una caída progresiva en amplitud, característica de la señal triangular. La segunda gráfica, obtenida en MATLAB, presenta el mismo patrón, pero con mayor definición y menor ruido visual, lo que permite apreciar mejor la caída en amplitud de los armónicos. La tercera gráfica, realizada en Python, reproduce el mismo comportamiento, aunque con una escala de frecuencia mal rotulada (aparece como "Tiempo (s)") y una representación más simplificada. En conjunto, todas confirman que la señal tiene una estructura armónica rica, dominada por componentes impares que disminuyen en magnitud, tal como se espera teóricamente para una onda triangular.

Se procede a hacer la FFT en la señal del tiempo:

```
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);
```

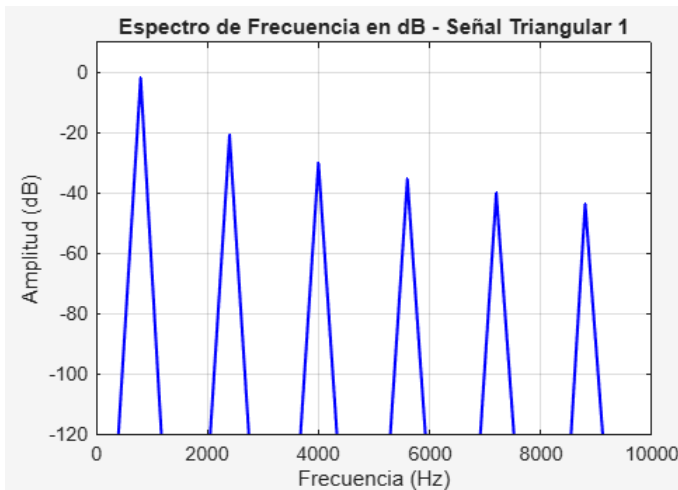
```
% === Gráfica en el dominio de la frecuencia (dB) ===
```

```
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

```



**Ilustración 26. Señal Tri1 con FFT**

Luego se grafica la IFFT

```

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;

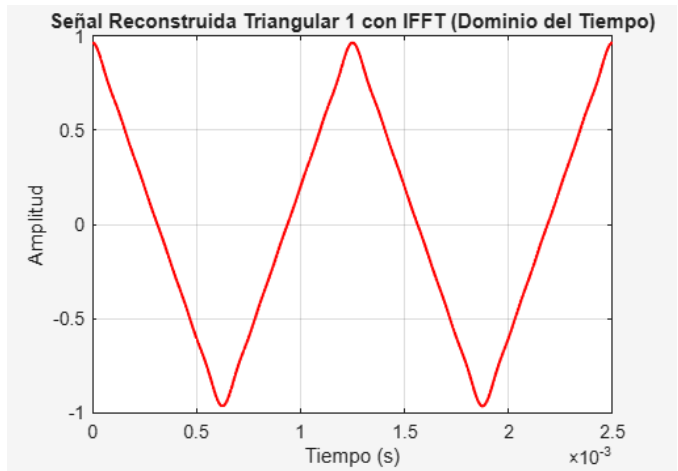


Ilustración 27. Señal Tri1 con IFFT

4. SEÑAL TRIANGULAR 2 CON D.C 1.5

FRECUENCIA	800 Hz
Vp	1

SEÑAL	DC	ARMONICA	RECUENCIA(Hz)	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

Se repite el proceso del anterior ítem:

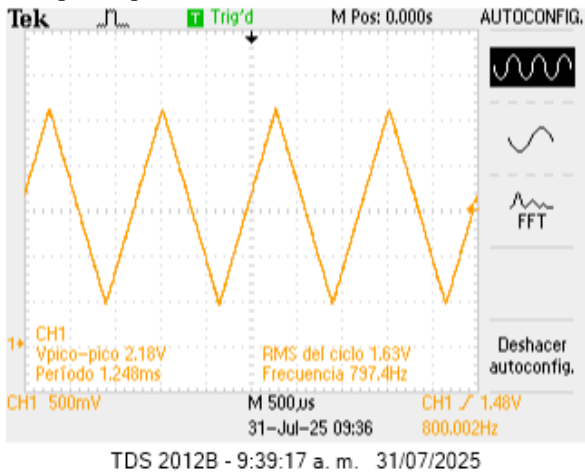


Ilustración 28. Señal TRI2 en el osciloscopio

Se porocede a hacer el grafico teorico cuyo reusltado es:

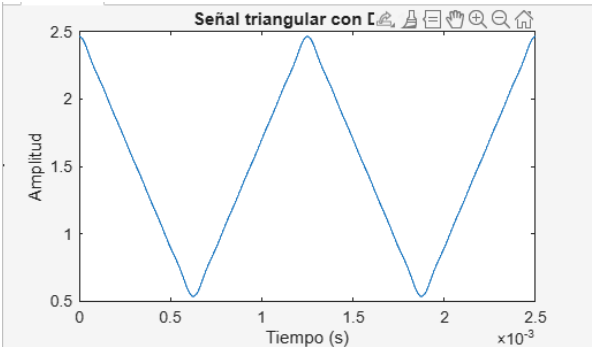


Ilustración 29. Señal Tri2 con datos teóricos Matlab

Luego se grafican los datos medidos en voltaje pico desde el osciloscopio, cuyo resultado es:

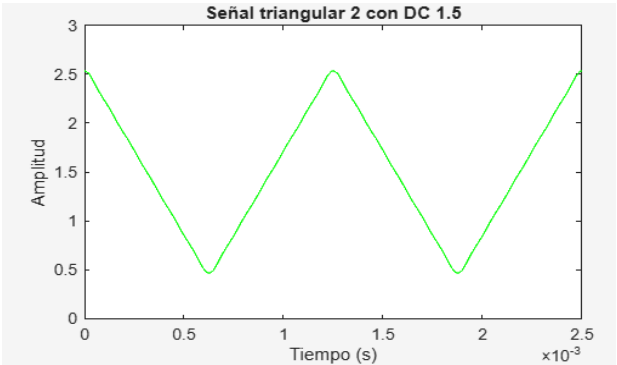


Ilustración 30. Señal Tri2 con datos medidos Matlab

Finalmente se grafica con los datos exportados:

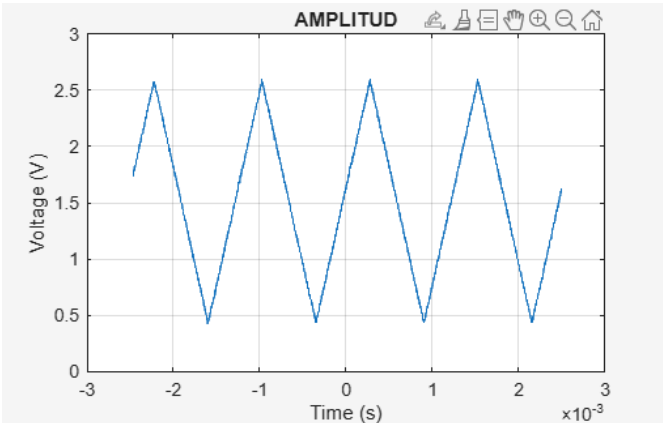
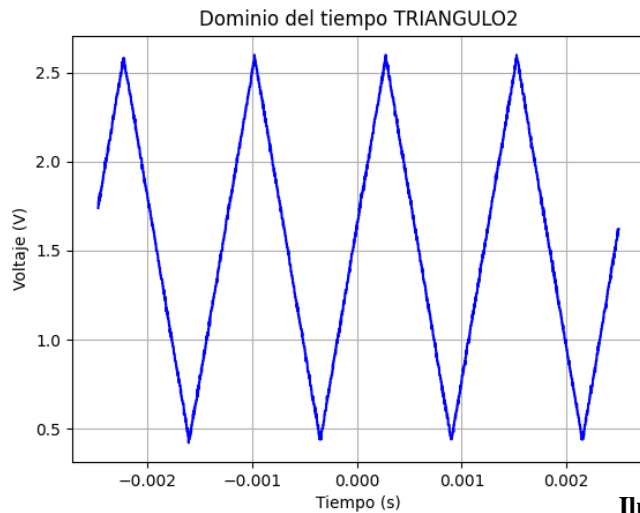


Ilustración 31. Señal Tri2 con datos exportados Matlab

Y en Phytton :

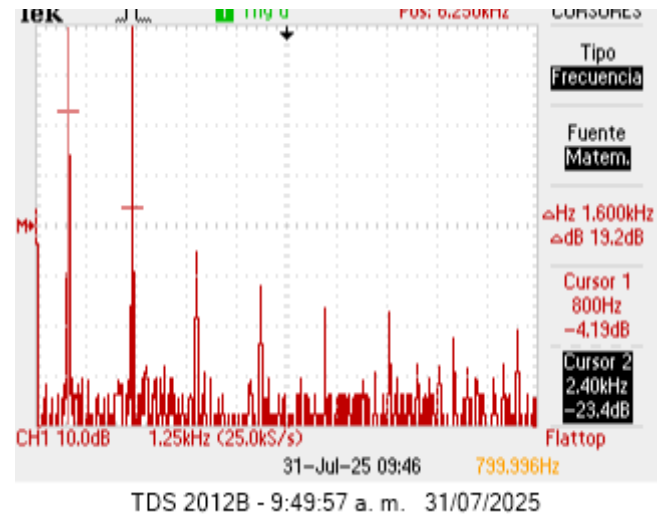


**Ilustración 32. Señal Tri2 en python Thonny**

En esta señal triangular 2 que incluye un componente de corriente continua (DC) de 1.5 V, a diferencia de la señal triangular anterior que no tenía DC y oscilaba simétricamente alrededor de 0 V. En la gráfica del osciloscopio, se observa cómo toda la señal se encuentra desplazada hacia arriba, con un valor medio claramente positivo, lo que confirma la presencia del DC. Al graficarla teóricamente, ese desplazamiento también se refleja de forma limpia y precisa, mostrando una onda triangular centrada en 1.5 V, es decir, con un recorrido entre aproximadamente 0.4 V y 2.6 V. En contraste, la señal triangular sin DC tenía un comportamiento simétrico respecto al eje horizontal, con un valor medio de 0 V y amplitud distribuida entre -1 V y 1 V. La inclusión del DC no modifica la forma de la onda, pero sí cambia su nivel de referencia, su valor promedio y su comportamiento frente a sistemas sensibles a componentes de continua. Por tanto, al comparar ambas señales, se concluye que el DC de 1.5 V desplaza toda la señal en el eje vertical sin alterar su periodicidad ni su contenido armónico.

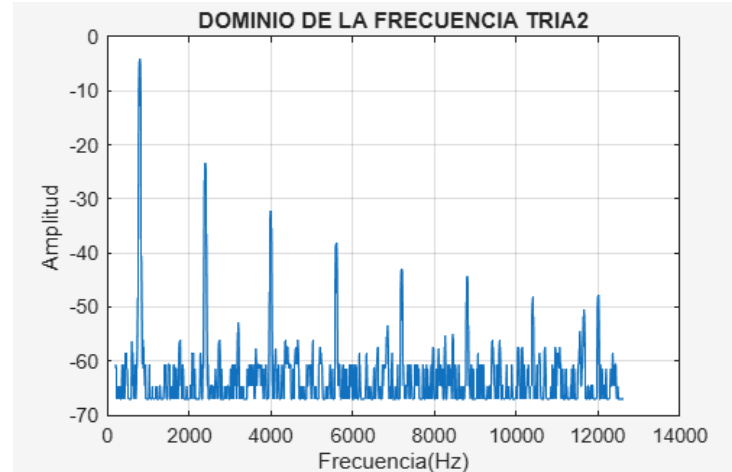
Ahora se procede a ubicar en el dominio de las frecuencias siguiendo los pasos anteriormente mostrados, en este caso solo se pondrán los resultados de los gráficos ya que se lleva el mismo proceso.

Se observa la señal generada por el osciloscopio



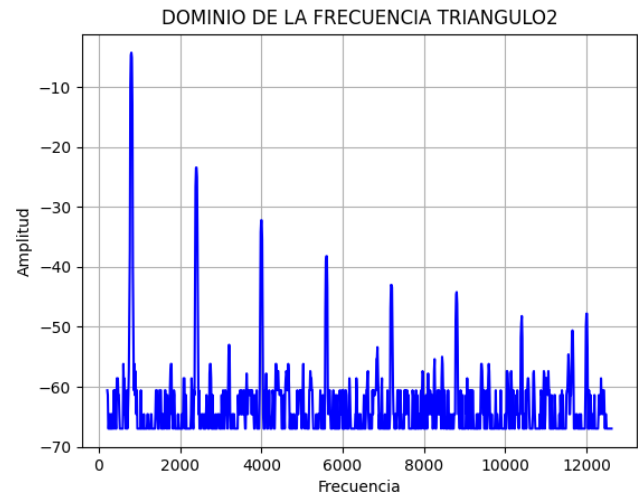
**Ilustración 33. Señal Tri2 en el dominio de la frecuencia**

Luego se exportan los datos al matlab:



**Ilustración 34- Señal Tri2 con datos exportados Matlab**

Por consiguiente se grafica en Phytion:



**Ilustración 35. Señal Tri2 en python dominio frecuencia**

Las tres gráficas representan el análisis en frecuencia de una señal triangular con componente DC, evidenciado por el pico



en 0 Hz y la presencia exclusiva de armónicos impares. La primera gráfica, obtenida desde el osciloscopio, muestra claramente el comportamiento real de la señal: un pico principal en la frecuencia fundamental y armónicos impares con amplitud decreciente, aunque con mayor ruido y menor resolución debido a las limitaciones del equipo. La segunda gráfica, generada en MATLAB, presenta una mejor definición de los armónicos, con picos bien marcados hasta frecuencias elevadas y un piso de ruido más bajo, lo que facilita una lectura más precisa.

Luego se halla la FFT al dominio del tiempo.

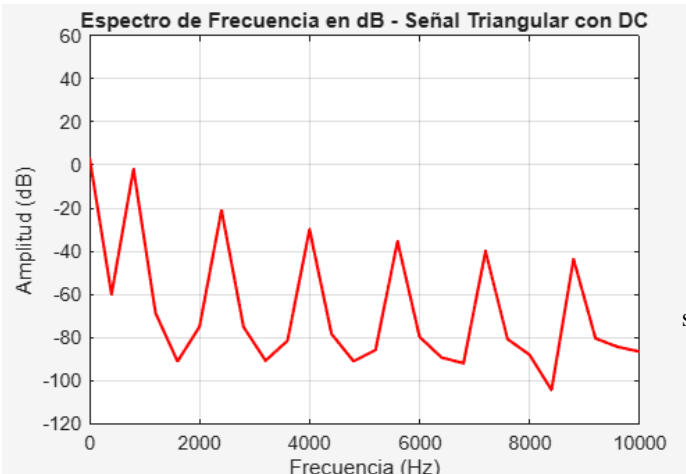


Ilustración 36. Señal Tri2 con FFT

Finalmente se halla la IFFT

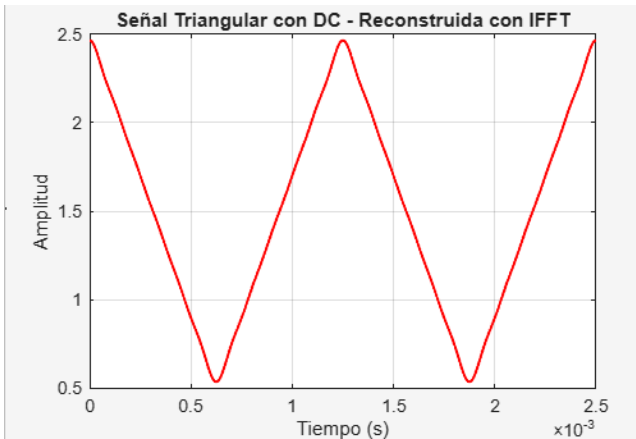


Ilustración 37. Señal Tri2 con IFFT

En la primera gráfica, la FFT permite descomponer la señal en sus componentes de frecuencia, revelando un espectro formado por armónicos impares decrecientes, típicos de una onda triangular, junto con un pico en 0 Hz que representa el valor DC. Esta información permite entender cómo se distribuye la energía de la señal en el dominio de la frecuencia. Posteriormente, la IFFT, como se muestra en la segunda gráfica, permite reconstruir la señal original en el dominio del tiempo a partir de esos componentes espectrales. El resultado es una señal triangular desplazada verticalmente, que conserva su forma y nivel de continua, validando la precisión del análisis

espectral. En conjunto, la FFT facilita el estudio de la estructura armónica de la señal, mientras que la IFFT comprueba que esa información puede ser usada para recuperar fielmente la señal original, incluso con su desplazamiento DC.

5. SEÑAL CUADRADA 1

En esta señal se recopilan los siguientes datos:

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

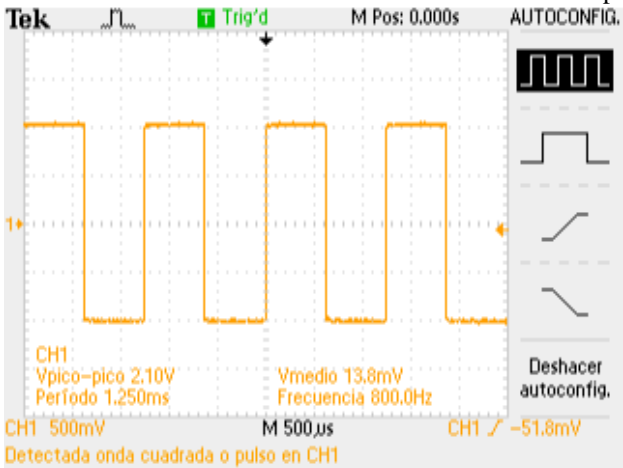
La formula que se uso para la FFT teorica en la tabla es la siguiente:

$$A_n = \frac{4V}{n\pi}$$

$$x(t) = A_{DC} + \frac{4A}{\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n} \sin(2\pi n f_0 t)$$

Esta formula se usa para señales cuadradas ya que solo aparecen armónicas impares, esto corresponde a los coeficientes de la serie de Fourier para descomponer señales periódicas. El valor de AnA\_nAn depende de la forma de la onda y determina la contribución de cada armónico.

Teniendo esto en cuenta se muestra la señal en el osciloscopio:



TDS 1212B - 9:54:25 a. m. 31/07/2025 Ilustración 38. Señal Cua1 en el osciloscopio

Ahora se grafican los valores teoricos en Matlab: T=0.00125;

f=1/T;

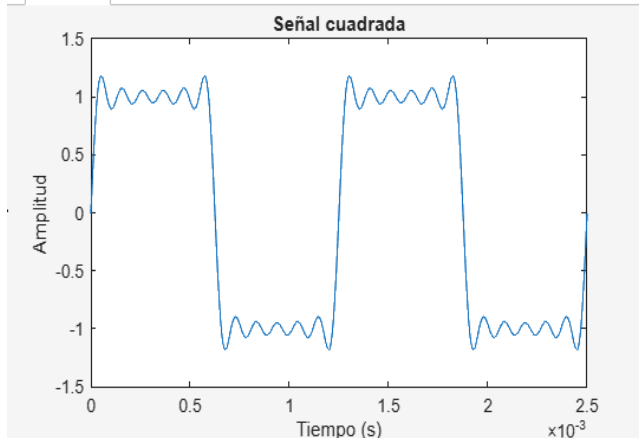
t=0:1/1000000:2\*T;

```

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');

```



**Ilustración 39. Señal Cua1 con datos teóricos Matlab**

Ahora se grafican los datos que fueron medidos en voltaje pico en el osciloscopio:

```
T=0.00125;
```

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

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

```
A1=1.260;
```

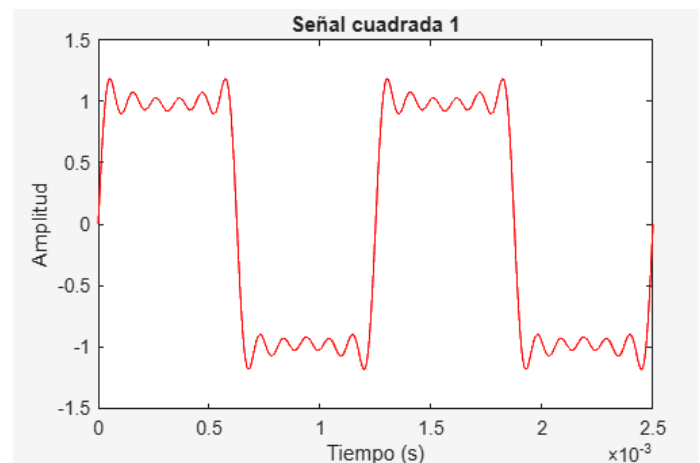
```
A3=0.437;
```

```

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');

```



**Ilustración 40. Señal Cua1 con datos medidos Matlab**

Por consiguiente se grafica las tablas exportadas csv:

```

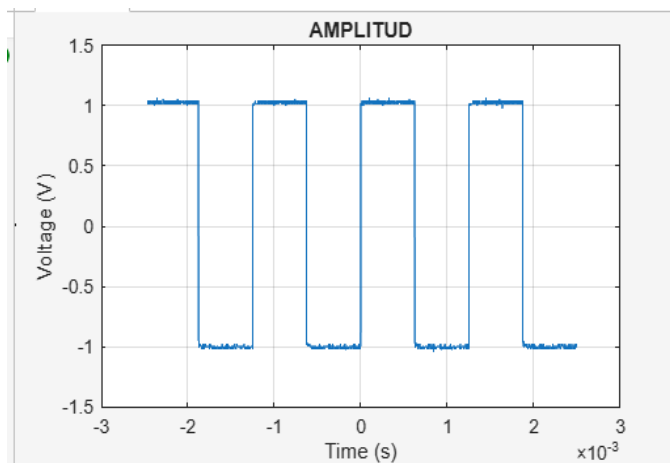
opts = detectImportOptions('DOMINIOTIEMPOCUA1.csv');
opts.DataLines = [18 Inf]; % Solo desde la fila 18 en adelante
opts.Delimiter = ','; % Delimitador por coma
opts.VariableNamesLine = 0; % No hay nombres de variables en la fila 18

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

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

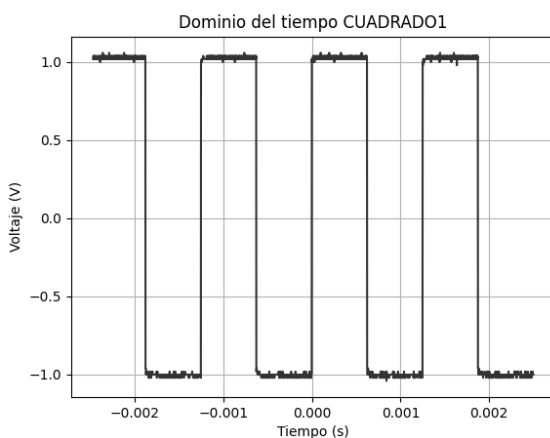
```

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



**Ilustración 41. Señal Cua1 con datos exportados Matlab**

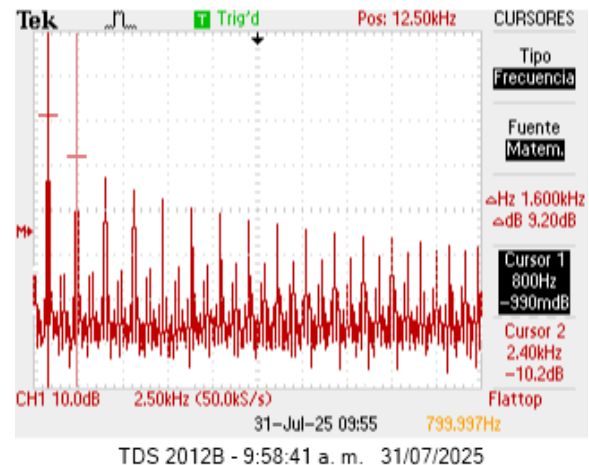
Por ultimo se grafica en Phyton:



**Ilustración 42. Señal Cua1 en Phyton**

La señal observada en el osciloscopio presenta un comportamiento ideal, con un voltaje pico a pico de 2.10 V y un periodo de 1.25 ms, lo que sirve como referencia para comparar las demás representaciones. En MATLAB se reconstruyó la señal cuadrada utilizando armónicos impares (1, 3, 5, 7, 9 y 11), lo que permitió evidenciar el fenómeno de Gibbs, donde se observan oscilaciones en los flancos de la señal debido a la cantidad limitada de armónicos empleados. Esto genera una aproximación teórica que, aunque cercana, no alcanza la perfección de la señal ideal. Por otro lado, las gráficas generadas en Python muestran una forma de onda cuadrada más definida, con transiciones abruptas entre los niveles alto y bajo, similares a la señal vista en el osciloscopio.

Esto sugiere que en Python se utilizó una representación digital directa o una reconstrucción más completa. Ahora veamos en el dominio de la frecuencia



**Ilustración 43. Señal Cua1 dominio frecuencia osciloscopio**

En el dominio de la frecuencia, se analiza la composición espectral de la señal cuadrada de 800 Hz. La primera imagen del osciloscopio muestra picos armónicos claramente definidos en múltiplos impares de 800 Hz (fundamental, 3ra, 5ta, etc.), lo cual es característico de una señal cuadrada ideal. Los armónicos disminuyen progresivamente en amplitud, lo que evidencia la riqueza armónica de esta señal.

Ahora con el archivo csv:

```
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

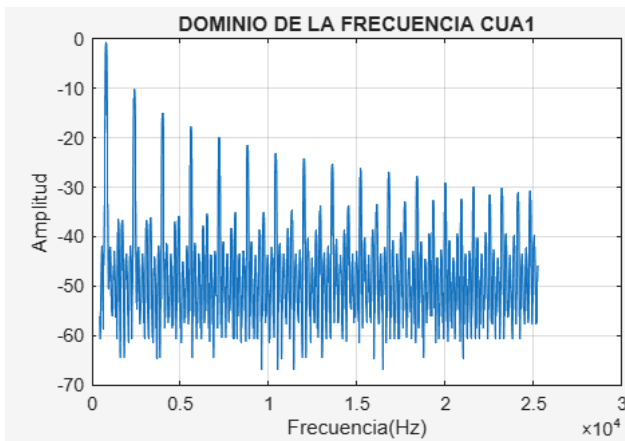
plot(t, v);

xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA CUA1');

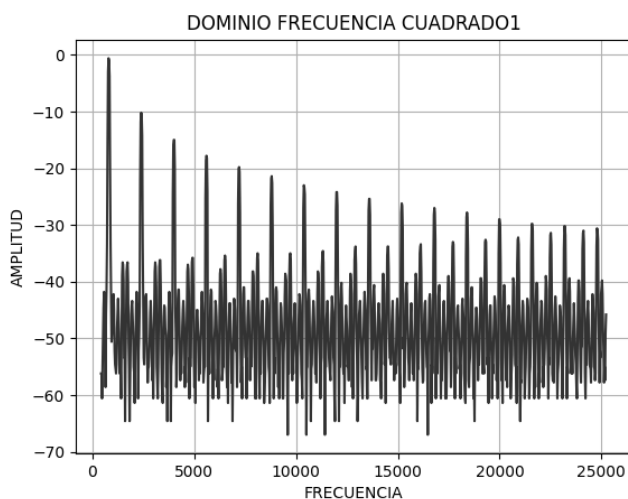
grid on;
```



**Ilustración 44. Señal Cua1 en Matlab**

La segunda gráfica, generada en MATLAB, confirma esta estructura: se observan picos notables en frecuencias impares con una caída gradual en la magnitud (en dB). Se evidencia ruido de fondo, pero la estructura teórica se mantiene, validando el análisis de Fourier.

Y finalizamos con phytton :



**Ilustración 45. Señal Cua1 en Python**

En esta gráfica del dominio de la frecuencia, generada desde Python, se observa claramente la descomposición espectral de una señal cuadrada de 800 Hz. Se destacan picos bien definidos en frecuencias múltiplos impares de la fundamental (800 Hz, 2400 Hz, 4000 Hz, etc.), lo cual coincide con la teoría de Fourier para una señal cuadrada ideal.

La amplitud de los armónicos disminuye de manera progresiva conforme aumenta la frecuencia, reflejando el comportamiento inversamente proporcional esperado para este tipo de señales. A partir de los 10 kHz, la amplitud de los armónicos se reduce considerablemente, predominando el ruido de fondo por debajo de los -40 dB.

Ahora con FFT

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)');
```

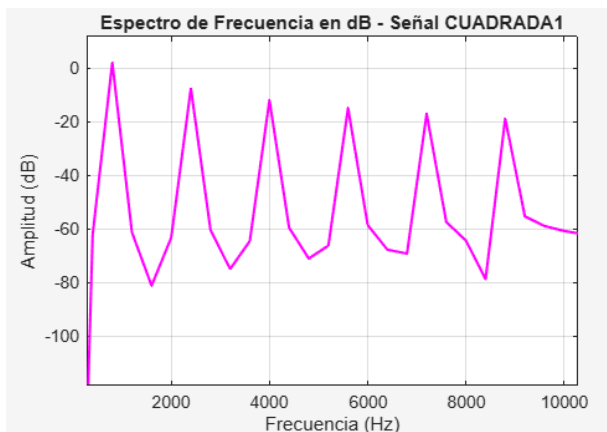
```
ylabel('Amplitud (dB)');
```

```
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
```



**Ilustración 46. Señal Cua1 con FFT**

Con IFFT

```
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 ===
```

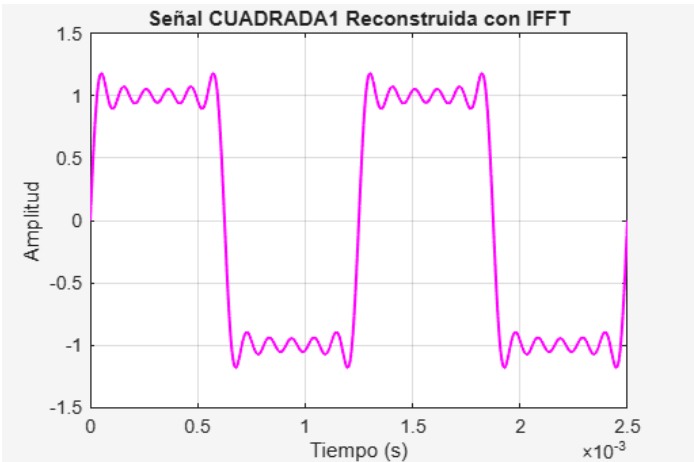
```
figure;
```

```
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;
```



**Ilustración 47. Señal Cua1 con IFFT**

Las gráficas obtenidas mediante FFT e IFFT corresponden correctamente a la señal cuadrada trabajada durante el laboratorio. En la primera imagen, se observa la reconstrucción temporal de la señal cuadrada utilizando la IFFT, la cual presenta una forma característica con transiciones abruptas entre los niveles alto y bajo, así como oscilaciones en los flancos conocidas como efecto Gibbs. Estas oscilaciones son esperadas debido al número limitado de armónicos utilizados en la reconstrucción y no representan un error, sino una propiedad matemática propia de las señales no continuas.

Por otro lado, el espectro de frecuencia obtenido mediante FFT muestra picos en los armónicos impares de la frecuencia fundamental (800 Hz, 2400 Hz, 4000 Hz, etc.), lo que es coherente con la teoría de Fourier para una señal cuadrada. Además, la magnitud de los armónicos disminuye

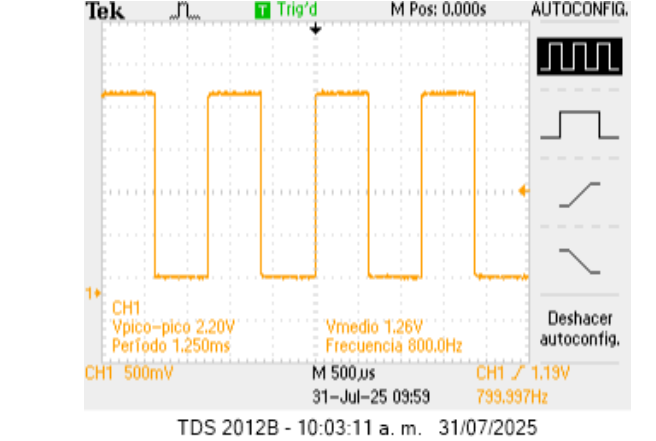
progresivamente, lo cual también coincide con los datos obtenidos en la tabla experimental y las gráficas anteriores.

**6. SEÑAL CUADRADA 2 CON DC 1.25**

FRECUCENCIA		800 Hz					
Vp		1					
SEÑAL	DC	ARMONICA	FRECUCENCIA(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

Se realiza el mismo proceso de la anterior señal, es decir, la cuadrada1, lo que quiere decir que no se pondran codigos por lo que solo se analizaran las graficas obtenidas remplazando los datos en los codigos anteriores.

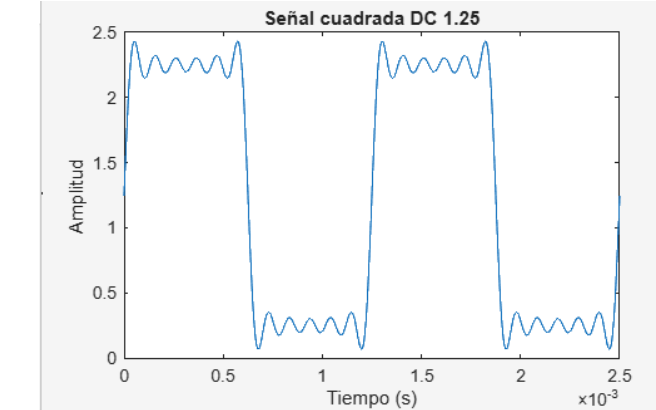
Entonces se inicia con el osciloscopio



**Ilustración 48. Señal Cua2 dominio tiempo osciloscopio**

La señal capturada en el osciloscopio presenta una forma cuadrada con valor pico a pico de 2.20 V y un valor medio de 1.26 V. Esto indica una desplazada hacia arriba, debido al componente DC de 1.25 V. El periodo sigue siendo de 1.25 ms, correspondiente a 800 Hz.

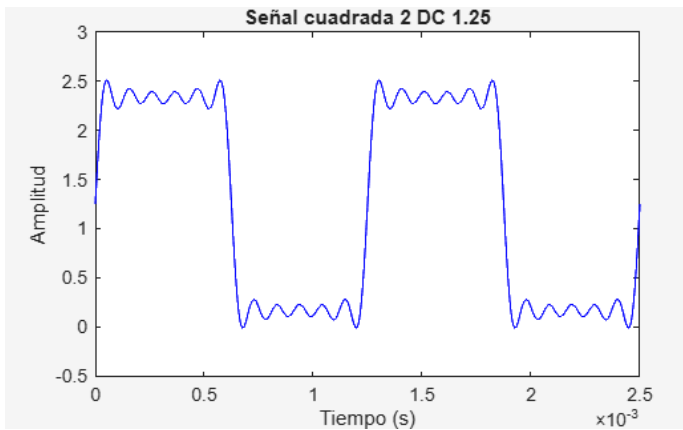
Con datos teoricos en matlab:



**Ilustración 49. Señal Cua2 teórico Matlab**

Con los volatjes pico medidos :

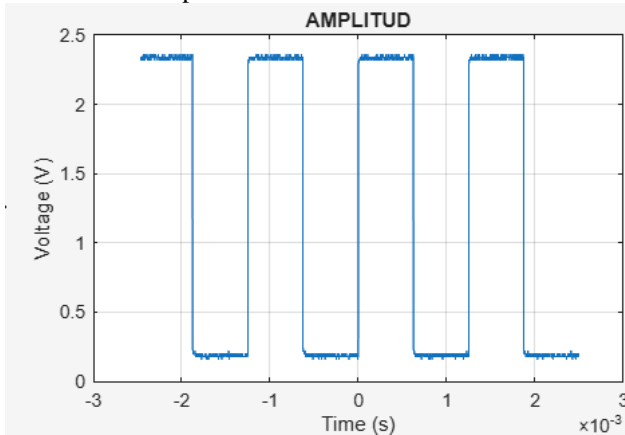




**Ilustración 50. Señal Cua2 medido Matlab**

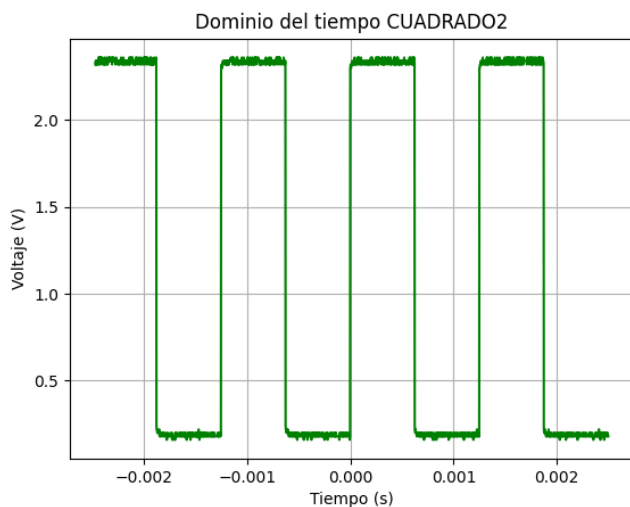
Las dos gráficas generadas en MATLAB muestran señales cuadradas reconstruidas mediante armónicos. La forma de onda mantiene la característica de una señal cuadrada, pero con ondulaciones en los flancos, resultado del efecto Gibbs. Además, se nota el desplazamiento vertical de la señal, producto de la componente DC.

Con los datos exportados csv:



**Ilustración 51. Señal Cua2 datos exportados Matlab**

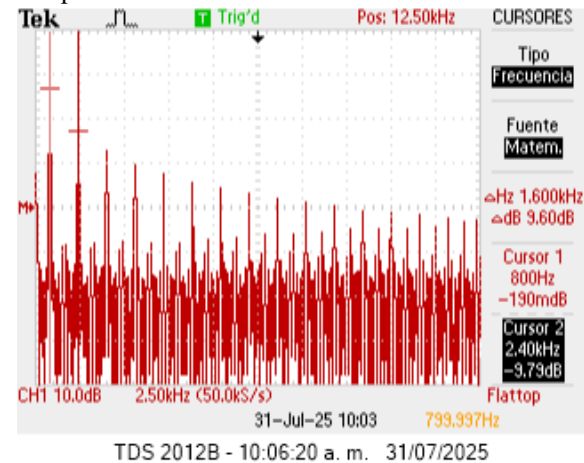
Y finalmente en phyton:



**Ilustración 52. Señal Cua2 Python**

La señal cuadrada con componente DC = 1.25 V mantiene su frecuencia fundamental de 800 Hz y presenta armónicos impares decrecientes, como se espera teóricamente. Las gráficas en MATLAB y Python, junto con los datos del osciloscopio, validan que la presencia de DC no afecta la periodicidad ni el contenido armónico de la señal, pero sí desplaza su nivel de voltaje promedio.

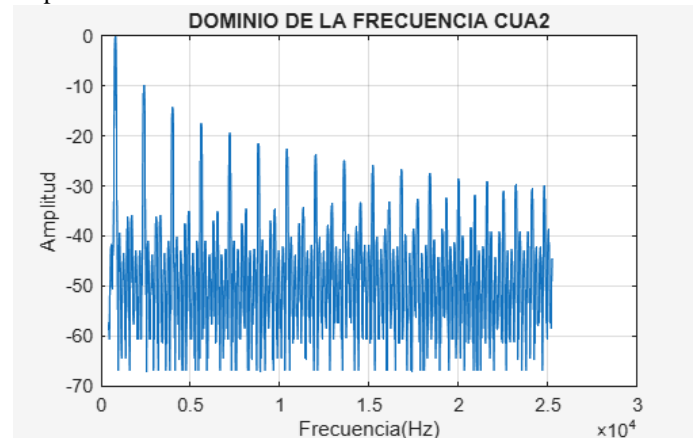
Ahora en el dominio de la frecuencia, primero con el osciloscopio



**Ilustración 53. Señal Cua2 Dominio frecuencia**

La gráfica del osciloscopio muestra un espectro con componentes armónicos impares claramente visibles (800 Hz, 2400 Hz, 4000 Hz...), coherentes con la estructura de la señal cuadrada. A diferencia de la señal sin DC, el primer pico a 0 Hz (DC) tiene una amplitud más alta, lo cual indica correctamente la presencia de una componente continua de 1.25 V. La magnitud de los armónicos decae de forma progresiva, y se observa algo de ruido en frecuencias altas.

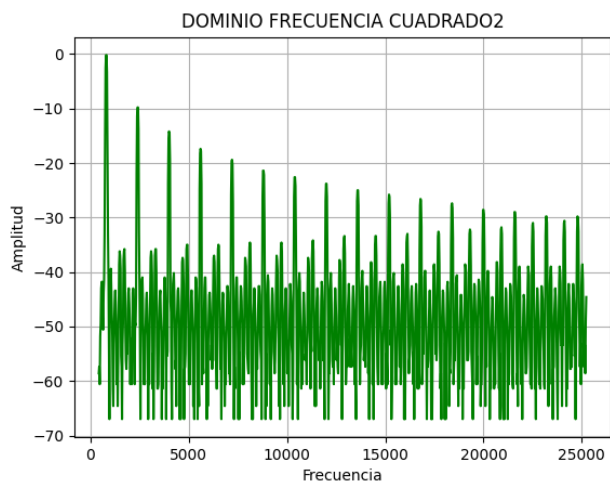
Exportando los datos csv



**Ilustración 54 señal cuad2 dominio de frecuencia**

Se distinguen múltiples picos armónicos espaciados uniformemente, cuya amplitud disminuye de forma clara. Aunque el ruido de fondo está presente, se visualiza también una componente fuerte en bajas frecuencias, asociada al valor DC. Las etiquetas de los ejes son correctas, lo que facilita su interpretación.

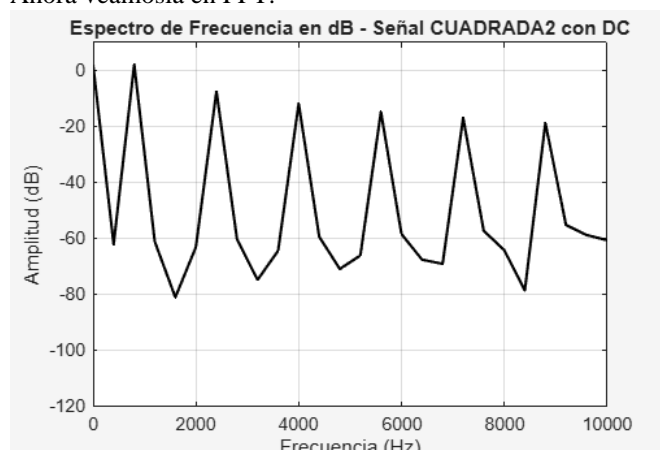
Finalmente en python



**Ilustración 55 señal cuadrada 2 dominio frecuencia-Python**

Las tres gráficas muestran consistencia en cuanto al contenido armónico de la señal cuadrada con  $DC = 1.25$  V. El pico en frecuencia cero confirma la presencia de la componente continua. Los armónicos impares dominan el espectro, como dicta la teoría. Las herramientas utilizadas —osciloscopio, MATLAB y Python— permiten validar que el comportamiento espectral de la señal cuadrada desplazada es coherente

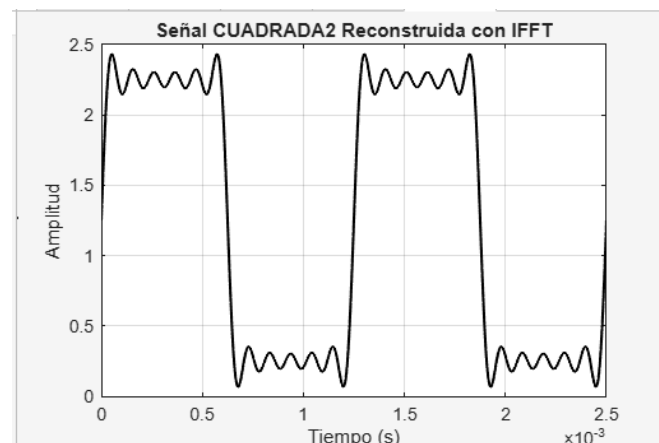
Ahora veámosla en FFT:



**Ilustración 56 señal cuad2 en dominio de frecuencia mediante FFT**

Se observan picos bien definidos en los múltiplos impares de la frecuencia fundamental (800 Hz, 2400 Hz, 4000 Hz, 5600 Hz, 7200 Hz, 8800 Hz...), lo cual es característico de una señal cuadrada ideal. Estos armónicos muestran una disminución progresiva en su magnitud, lo cual concuerda con el comportamiento teórico de una señal cuadrada, donde la amplitud de los armónicos decrece con el número de orden. La presencia de un pico muy marcado en las bajas frecuencias confirma la existencia de una componente continua (DC) en la señal. Esto es coherente con el desplazamiento de la señal hacia valores positivos (1.25 V), ya que la FFT detecta energía en la frecuencia 0 Hz. La caída en magnitud entre los armónicos es regular, sin distorsión aparente, lo que indica una adquisición limpia y correcta aplicación de la FFT.

Con la IFFT



**Ilustración 57 señal cuad2 dominio del tiempo por IFFT**

Esta señal representa la señal temporal obtenida al aplicar la IFFT sobre el espectro calculado anteriormente. Se observa una forma de onda cuadrada claramente definida, con alternancia periódica entre valores altos y bajos. A diferencia de una señal cuadrada centrada en cero, esta señal está desplazada verticalmente, oscilando aproximadamente entre 0.25 V y 2.25 V, lo cual valida la presencia de un nivel DC de 1.25 V (valor medio de la señal). Se presentan oscilaciones en los flancos de subida y bajada de la señal, producto del efecto Gibbs, causado por la suma de una cantidad limitada de armónicos en la reconstrucción. Esto no es un error, sino una consecuencia matemática natural. El periodo de la señal es de aproximadamente 1.25 ms, correspondiente a una frecuencia de 800 Hz, lo cual concuerda con la señal original.

La diferencia principal entre ambas señales radica en la presencia de la componente continua (DC) en la señal CUADRADA2, que provoca un desplazamiento vertical de toda la onda sin afectar su periodicidad ni estructura armónica general. Ambas señales comparten la misma frecuencia fundamental (800 Hz) y presentan armónicos impares en su espectro, lo que es característico de las señales cuadradas. Sin embargo, la CUADRADA2 añade una componente en frecuencia 0 Hz, confirmando el valor DC.

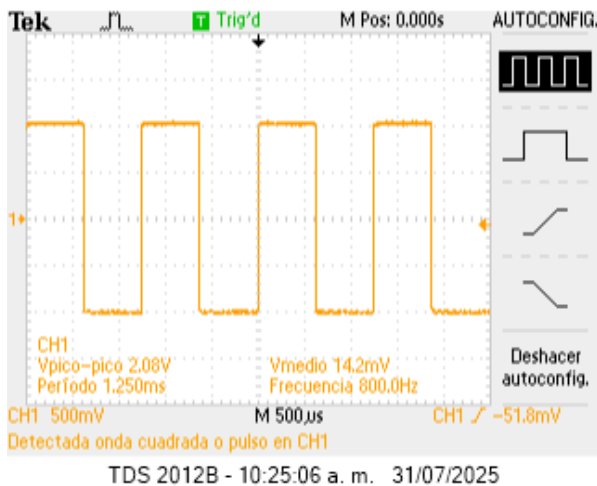
## 7. SEÑAL PULSO1 CICLO UTIL 50%

Una señal pulso es un tipo de señal que cambia bruscamente entre dos niveles, generalmente uno alto y uno bajo, durante un tiempo muy corto. Se caracteriza por tener un inicio y un fin definidos, es decir, no es continua en el tiempo, sino que aparece de forma repentina y de corta duración, como un destello.

Ahora bien, se tiene la siguiente tabla con los datos

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

Se obtiene la siguiente señal en el osciloscopio:



#### Ilustración 58 señal pulso 1 dominio tiempo - osciloscopio

La señal observada presenta un patrón cuadrado clásico con frecuencia de 800 Hz, periodo de 1.25 ms y un Vpico-pico de 2.08 V. El voltaje medio es 142 mV, lo cual indica que está centrada en cero con valores positivos y negativos casi simétricos, confirmando el ciclo útil del 50%. El osciloscopio detecta la señal correctamente como una onda cuadrada o pulso periódico.

Se procede a graficar el pulso teorico en Matlab:

```
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);
```

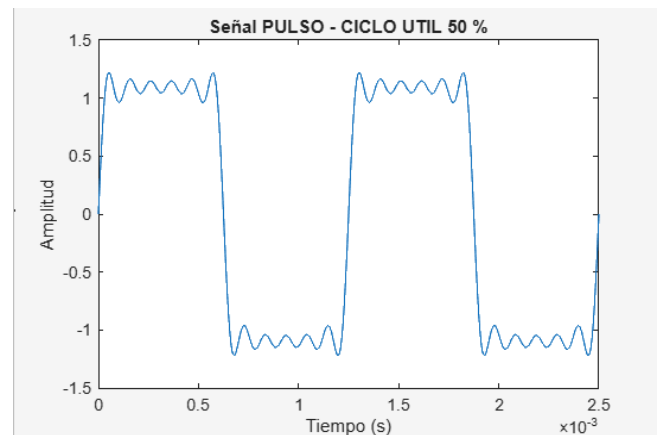
```
VT=(v1+v2+v3+v4+v5+v6);
```

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

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

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

```
title('Señal PULSO - CICLO UTIL 50 %');
```



#### Ilustración 59 señal pulso uno dominio del tiempo- Matlab

La forma de la onda generada con fórmulas teóricas presenta bordes definidos, pero con ondulaciones en los flancos (efecto Gibbs), típico al usar un número limitado de armónicos en la reconstrucción. La señal oscila entre +1 V y -1 V, con transiciones bruscas, confirmando la forma de onda pulsada con ciclo útil del 50%.

Se procede a hacer la grafica con los datos exportados en csv:

```
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

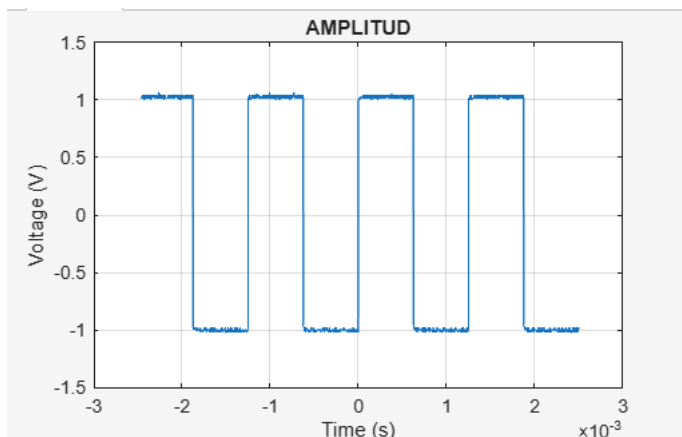
plot(t, v);

xlabel('Time (s)');

ylabel('Voltage (V)');

title('AMPLITUD');

grid on;



**Ilustración 60 señal pulso1 de archivo .CSV tiempo, Matlab**

Muestra la misma forma cuadrada, validando experimentalmente los datos teóricos, aunque con pequeñas imperfecciones propias de la medición. Se conserva el patrón de simetría, los niveles de amplitud y el periodo.

En Python:

```
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')
```

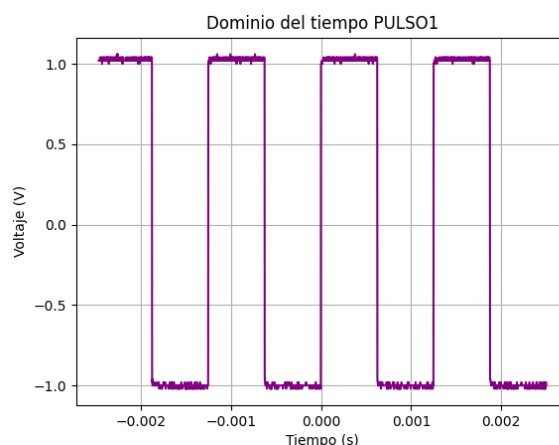
```
plt.title('Dominio del tiempo PULSO1')
```

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

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

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

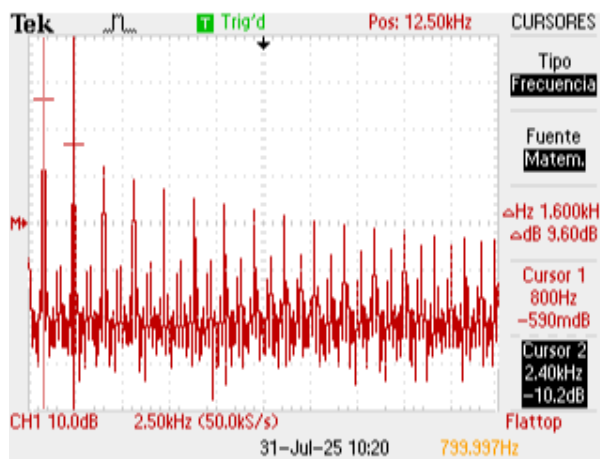
```
plt.show()
```



**Ilustración 61 señal pulso1 de archivo .CSV tiempo- Python**

La señal visualizada en Python también corresponde a una onda cuadrada pura con transición rápida entre niveles alto y bajo. No presenta el efecto Gibbs, ya que se usó una representación ideal o con más armónicos. El voltaje oscila entre +1 V y -1 V, sin desplazamiento vertical, coincidiendo con la tabla y el osciloscopio.

Ahora se observa en el dominio de la frecuencia



**Ilustración 62 señal pulso1 dominio frecuencia- osciloscopio**

La amplitud de los armónicos decrece progresivamente, reflejando una distribución armónica esperada para un tren de pulsos con ciclo útil del 50%. Como se trata de una señal simétrica respecto al eje horizontal, solo aparecen armónicos impares, como también lo muestra la tabla de datos

Se exporta los datos a Matlab:

```
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);

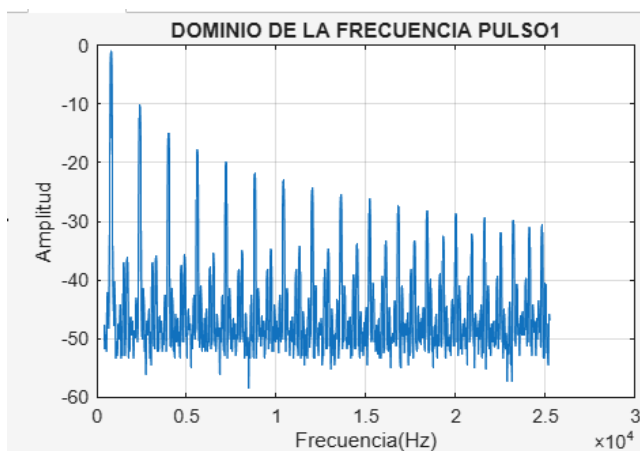
xlabel('Frecuencia(Hz)');

ylabel('Amplitud');

title('DOMINIO DE LA FRECUENCIA PULSO1');

grid on;

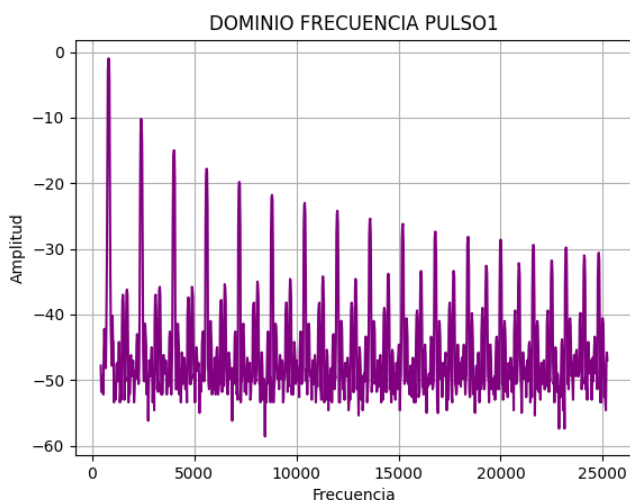
```



**Ilustración 63 pulso1 en frecuencia desde .CSV-Matlab**

Los picos de mayor magnitud aparecen también en los múltiplos impares de 800 Hz, y la caída de amplitud en dB es consistente con la teoría de Fourier. La gráfica también muestra ruido espectral de fondo (por debajo de -40 dB), propio de mediciones reales y discretización.

Para finalizae en Phyton:



**Ilustración 64 pulso1 en frecuencia-Python**

La escala y el espaciado entre armónicos siguen el mismo comportamiento, confirmando la correspondencia entre teoría, simulación y medición. Surge una duda ¿Por qué esta señal se parece a una señal cuadrada si es pulso? La señal PULSO1 se parece a una señal cuadrada porque esencialmente lo es: tiene un ciclo útil del 50%, es simétrica, no tiene componente DC, y sus armónicos son impares, como se espera de una cuadrada ideal. De hecho, en teoría de señales, una onda cuadrada es un tipo especial de señal de pulso con esa proporción exacta entre tiempo encendido y apagado.

Ahora bien, veamos la señal FFT

```

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;

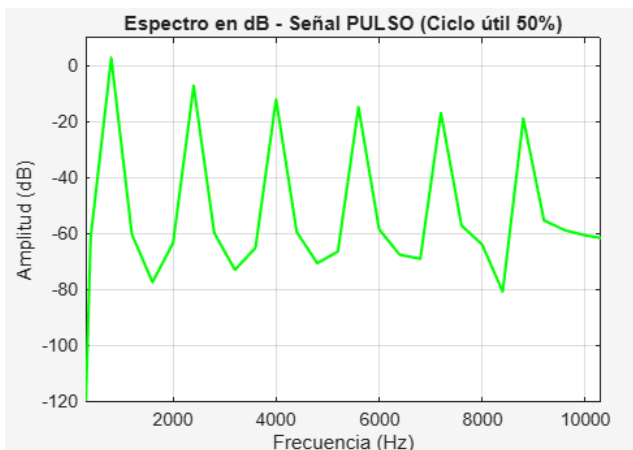
% ==== FFT de la señal ====
N = length(VT);
Y = fft(VT);
Y_mag = abs(Y)/N;
Y_mag(2:end-1) = 2 * Y_mag(2:end-1); % Duplicar
amplitudes (excepto DC y Nyquist)

% ==== 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

```



**Ilustración 65 pulso 1 en frecuencia mediante FFT**

Esta FFT confirma que la señal PULSO1 posee un espectro típico de una onda cuadrada, con componentes espectrales impares y simétricas

Por consiguiente IFFT

```

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 ===
```

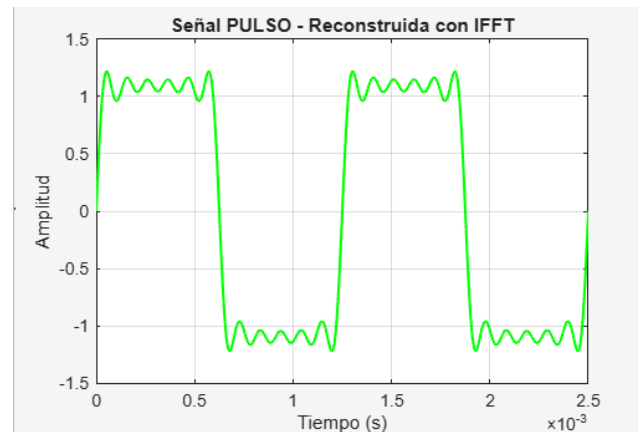
```
figure;
plot(f_axis(1:N/2), Y_dB(1:N/2), 'g', 'LineWidth', 1.5);
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
title('Espectro en dB - Señal PULSO (Ciclo útil 50%)');
grid on;
xlim([0 10000]);
ylim([-120 10]);
```

```
% === 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;
```



**Ilustración 66 pulso 1 en dominio de tiempo mediante IFFT**

La IFFT demuestra que es posible recuperar fielmente la forma temporal de la señal PULSO1 a partir de su contenido armónico, con mínima distorsión y respetando su simetría y frecuencia.

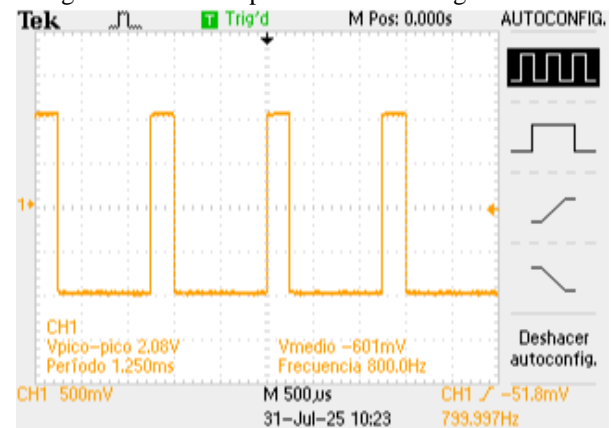
### 8. SEÑAL PULSO 2 CICLO UTIL 20%

Se realiza el mismo proceso que en el pulso1, solo que en esta sección solo se remplazarán valores en los códigos por lo que solo se mostrarán y analizarán las gráficas obtenidas. Ahora, se muestra la fórmula base para los cálculos teóricos y se mostrará la tabla con sus respectivos valores

$$v(t) = \sum_{n=1}^{\infty} \frac{2A}{n\pi} \cdot \sin(n\pi D) \cdot \cos(2\pi n f t)$$

FRECUENCIA	800 Hz						
Vp	1						
DC	0						
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO
				d.B	Vrms	VP(v)	(2*A/n*pi)
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

Al configurar el osciloscopio se muestra la siguiente señal:



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

**Ilustración 67 señal pulso 2 dominio tiempo - osciloscopio**

La señal capturada tiene un periodo de 1.25 ms y un  $V_{pico}$  de 2.08 V, lo que corresponde perfectamente con la frecuencia fundamental de 800 Hz. El valor medio es negativo (−601 mV), lo que indica un mayor tiempo en nivel bajo que alto, validando que se trata de un ciclo útil del 20%. La forma visual muestra pulsos breves con pausas prolongadas, lo que distingue esta señal de una onda cuadrada simétrica.

Se grafica en Matlab con datos teoricos:

$T=0.00125$ ;

$f=1/T$ ;

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

$D=0.2$ ;

$A1=0.636$ ;

$A2=0.318$ ;

$A3=0.212$ ;

$A4=0.159$ ;

$A5=0.127$ ;

$A6=0.106$ ;

$A7=0.090$ ;

$A8=0.079$ ;

$A9=0.070$ ;

$A10=0.063$ ;

$A11=0.057$ ;

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

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

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

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

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

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

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

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

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

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

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

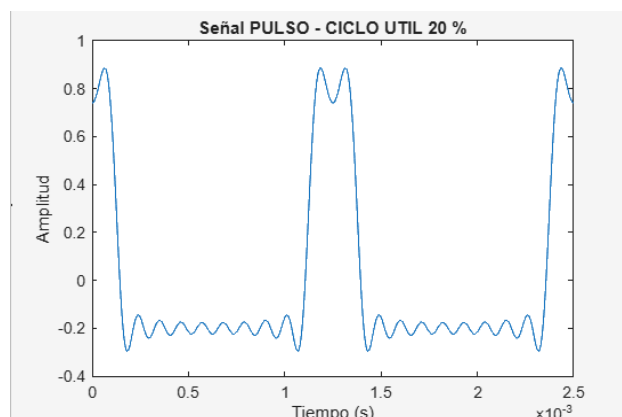
$VT=(v1+v2+v3+v4+v5+v6+v7+v8+v9+v10+v11)$ ;

`plot(t,VT);`

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

`ylabel('Amplitud');`

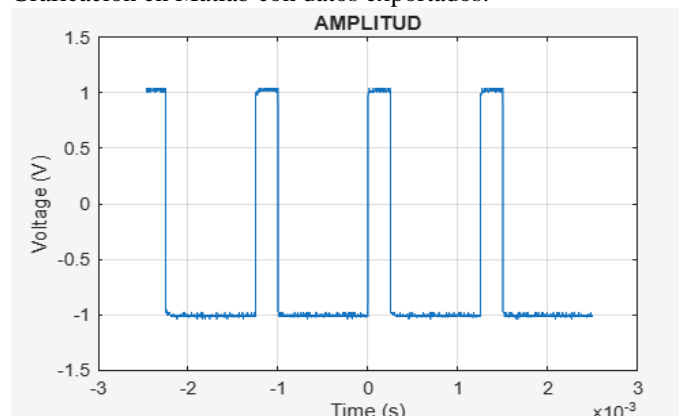
`title('Señal PULSO - CICLO UTIL 20 %');`



**Ilustración 68** señal pulso 2 dominio del tiempo- Matlab

La gráfica generada con fórmulas en MATLAB muestra una señal de pulso angosto con un ciclo útil del 20%. Es una reconstrucción teórica a partir de los armónicos. Se observan oscilaciones notorias en los flancos del pulso (efecto Gibbs), producto de la cantidad limitada de armónicos usados. La amplitud máxima es cercana a 0.8 V, y el resto del tiempo la señal permanece en valores bajos o cercanos a cero, como era de esperarse.

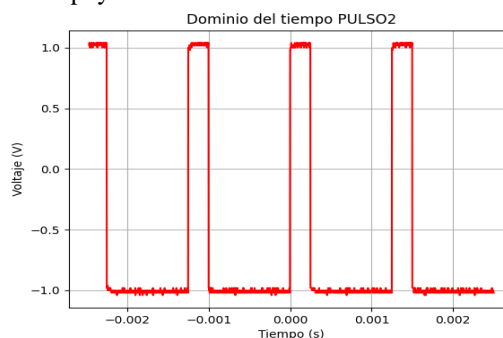
Graficacion en Matlab con datos exportados:



**Ilustración 69** señal pulso2 de archivo .CSV tiempo, Matlab

Esta reconstrucción se generó a partir de datos reales medidos, y muestra una forma de onda muy similar a la teórica. Los pulsos tienen la misma duración y forma, aunque con ligeras imperfecciones atribuibles al ruido o precisión de adquisición.

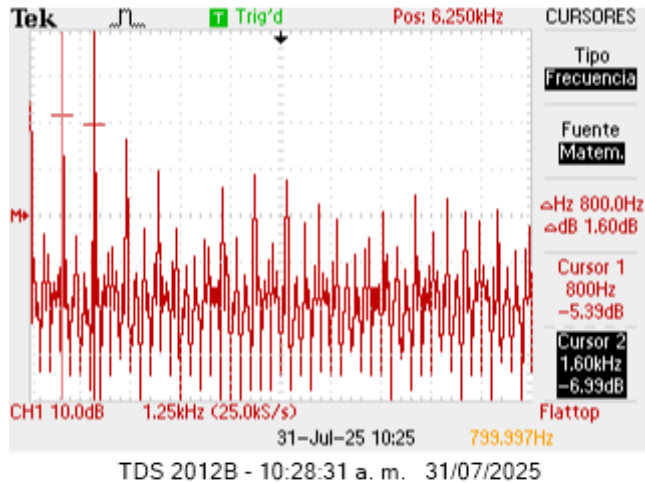
Se grafica en phyton:



**Ilustración 70** señal pulso2 de archivo .CSV tiempo, Python

El voltaje oscila entre +1 y -1 V, respetando la amplitud esperada de la señal original. Al igual que en las otras gráficas, la forma de los pulsos es clara, las transiciones son abruptas, y el comportamiento temporal es completamente coherente.

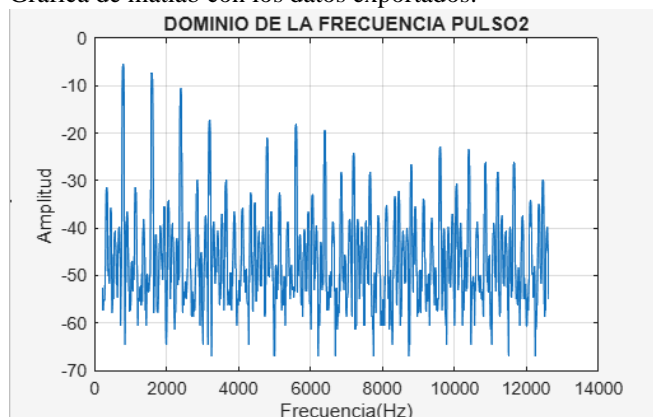
Ahora veamos las graficas en el dominio de la frecuencia, iniciando con el osciloscopio:



**Ilustración 71 señal pulso2 dominio frecuencia- osciloscopio**

Se observan armónicos tanto impares como pares, evidenciando que la señal no es simétrica respecto al eje horizontal, lo cual es coherente con su ciclo útil de solo 20%. El primer pico significativo aparece en la frecuencia fundamental de 800 Hz, con una amplitud de aproximadamente -5.39 dB. La caída de los armónicos no es tan regular como en señales cuadradas puras; hay una mayor densidad y ruido de fondo, lo que es típico de señales de pulsos angostos. El espectro es más rico en componentes debido a la naturaleza discontinua y breve del pulso.

Grafica de matlab con los datos exportados:

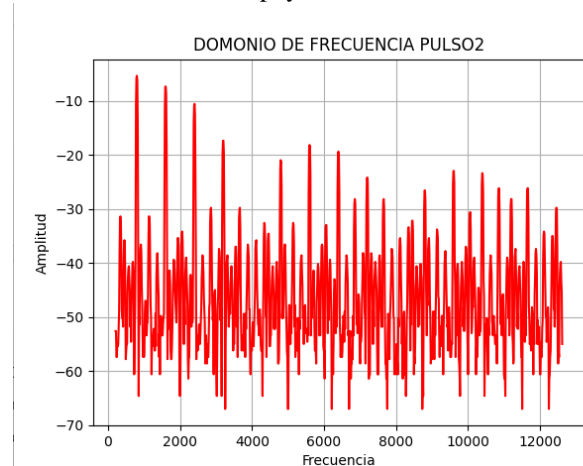


**Ilustración 72 pulso2 en frecuencia desde .CSV-Matlab**

Se aprecia un ruido de fondo a partir de los -40 dB, similar al de la medición real, y coherente con la resolución espectral de la transformada. El eje de frecuencia está correctamente en Hz, y los picos armónicos se alinean con múltiplos de la frecuencia base (800 Hz), confirmando su periodicidad. El espectro no presenta simetría y tiene un mayor número de componentes no

nulos, lo cual es natural en señales de ciclo útil pequeño.

Finalmente la señal en python:



**Ilustración 73 pulso2 en frecuencia-Python**

Visualmente, la gráfica es más nítida, y los picos son más fáciles de identificar gracias al color y trazo. El **primer armónico** (800 Hz) es el más fuerte, y la energía decrece a medida que la frecuencia aumenta.

Esta señal, al tener un ciclo útil más bajo, se aleja visualmente y espectralmente de una onda cuadrada pura, pero sigue siendo periódica, y su análisis permite ver cómo el ancho del pulso afecta directamente el contenido armónico.

Grafica en FFT

$T = 0.00125$ ; % 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) ===

$D = 0.2$ ;

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

$A_1 = 0.636$ ;  $A_2 = 0.318$ ;  $A_3 = 0.212$ ;  $A_4 = 0.159$ ;  $A_5 = 0.127$ ;  
 $A_6 = 0.106$ ;  $A_7 = 0.090$ ;  $A_8 = 0.079$ ;  $A_9 = 0.070$ ;  $A_{10} = 0.063$ ;  
 $A_{11} = 0.057$ ;

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

$v_1 = A_1 * \sin(1 * \pi * D) * \cos(2 * \pi * 1 * f * t)$ ;  
 $v_2 = A_2 * \sin(2 * \pi * D) * \cos(2 * \pi * 2 * f * t)$ ;  
 $v_3 = A_3 * \sin(3 * \pi * D) * \cos(2 * \pi * 3 * f * t)$ ;  
 $v_4 = A_4 * \sin(4 * \pi * D) * \cos(2 * \pi * 4 * f * t)$ ;  
 $v_5 = A_5 * \sin(5 * \pi * D) * \cos(2 * \pi * 5 * f * t)$ ;  
 $v_6 = A_6 * \sin(6 * \pi * D) * \cos(2 * \pi * 6 * f * t)$ ;  
 $v_7 = A_7 * \sin(7 * \pi * D) * \cos(2 * \pi * 7 * f * t)$ ;  
 $v_8 = A_8 * \sin(8 * \pi * D) * \cos(2 * \pi * 8 * f * t)$ ;  
 $v_9 = A_9 * \sin(9 * \pi * D) * \cos(2 * \pi * 9 * f * t)$ ;  
 $v_{10} = A_{10} * \sin(10 * \pi * D) * \cos(2 * \pi * 10 * f * t)$ ;  
 $v_{11} = A_{11} * \sin(11 * \pi * D) * \cos(2 * \pi * 11 * f * t)$ ;

$VT = v_1 + v_2 + v_3 + v_4 + v_5 + v_6 + v_7 + v_8 + v_9 + v_{10} + v_{11}$ ;

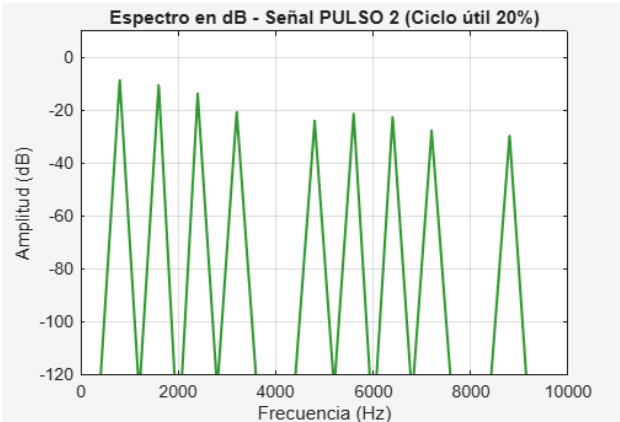
```
% === Gráfica de la señal en el tiempo ===
figure;
plot(t, VT);
xlabel('Tiempo (s)');
ylabel('Amplitud');
title('Señal PULSO - CICLO ÚTIL 20%');
grid on;

% === 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

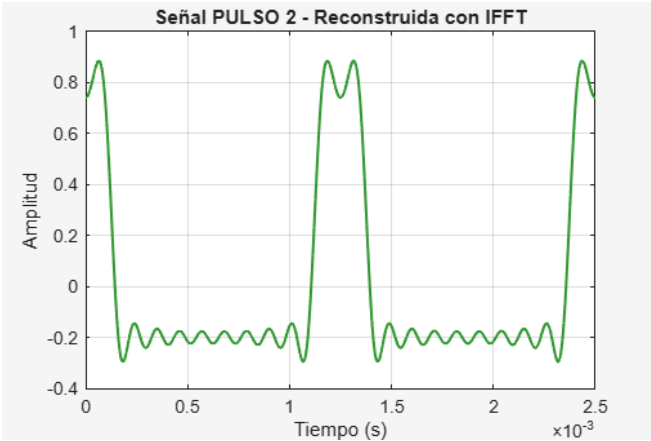
% === Conversión a decibelios ===
Y_dB = 20 * log10(Y_mag + eps); % eps evita log(0)

% === 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
```



**Ilustración 74 pulso 2 en frecuencia mediante FFT**  
La amplitud en dB decrece con la frecuencia, mostrando un contenido espectral amplio, típico de una señal de pulso angosto (duty cycle pequeño). La presencia de más armónicos con amplitudes significativas es una consecuencia directa del ciclo útil reducido al 20%, ya que pulsos más estrechos requieren más armónicos para ser representados con precisión. Esta gráfica refleja el comportamiento esperado: más armónicos, incluyendo pares, y mayor dispersión espectral debido al pulso angosto



**Ilustración 75 pulso 2 en dominio de tiempo mediante IFFT**

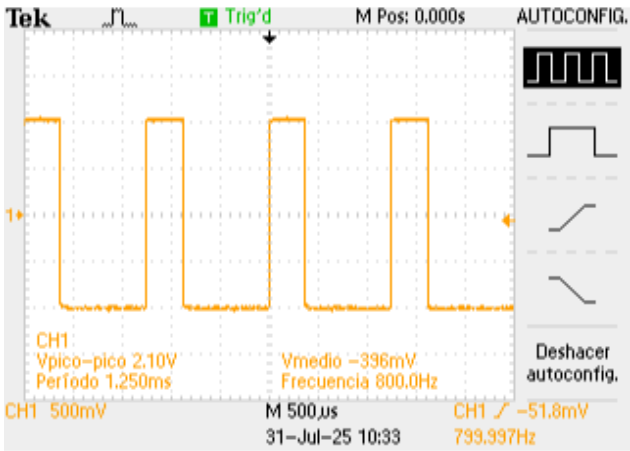
Se observa claramente un pulso angosto, de corta duración y amplitud cercana a 0.8 V, apareciendo de forma periódica. El resto del tiempo la señal permanece cerca de 0 o en valores bajos negativos, como corresponde a un ciclo útil de solo el 20%. El efecto Gibbs es visible alrededor de los flancos del pulso (oscilaciones pequeñas), indicando que la reconstrucción se hizo con un número limitado de armónicos.

**9. SEÑAL PULSO 3 CICLO UTIL 30%**

Se llevara el mismo proceso que en el Puslo 1 y 2, por lo cual la tabla del pulso 3 es la siguiente:

FRECUENCIA		800 Hz							
Vp		1							
DC		0							
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO		
				d.B	Vrms	VP[V]	[2°/a/n°π]		
PULSO3	30%	1	800	-4,59	0,5895	0,8337	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		

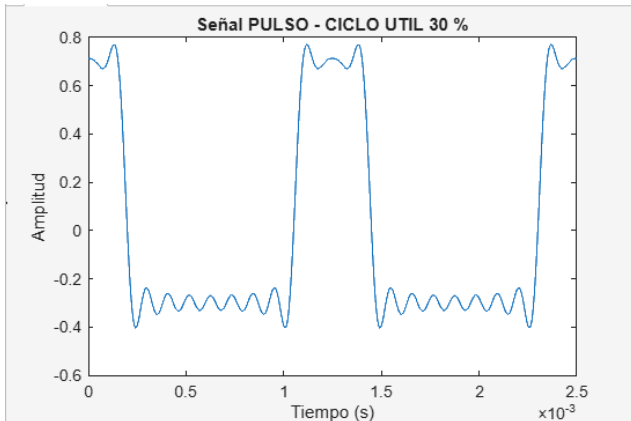
Osciloscopio:



TDS 2012B - 10:36:41 a. m. 31/07/2025  
**Ilustración 76 señal pulso 3 dominio tiempo - osciloscopio**

La forma de onda capturada en el osciloscopio muestra pulsos angostos pero más anchos que en PULSO2 (20%), con una amplitud pico a pico de 2.10 V, periodo de 1.25 ms, y un valor medio de  $-396$  mV. El pulso tiene una forma claramente rectangular, aparece de forma periódica y se mantiene apagado más tiempo del que está encendido, lo cual concuerda con un duty cycle del 30%.

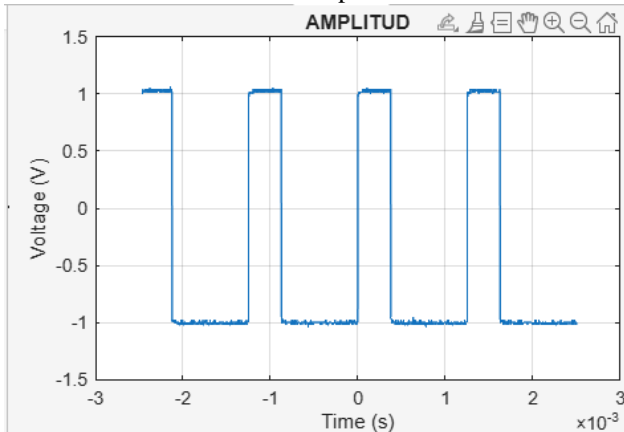
Grafica de Matlab con datos teoricos



**Ilustración 77 señal pulso 3 dominio del tiempo- Matlab**

La señal reconstruida muestra claramente un **pulso periódico angosto**, con amplitud aproximada de **0.7 V**, y oscilaciones leves en los flancos (efecto Gibbs). Aunque no es una onda cuadrada simétrica, la forma de la señal es clara y permite identificar el ciclo útil del 30% por el tiempo activo relativo a cada periodo.

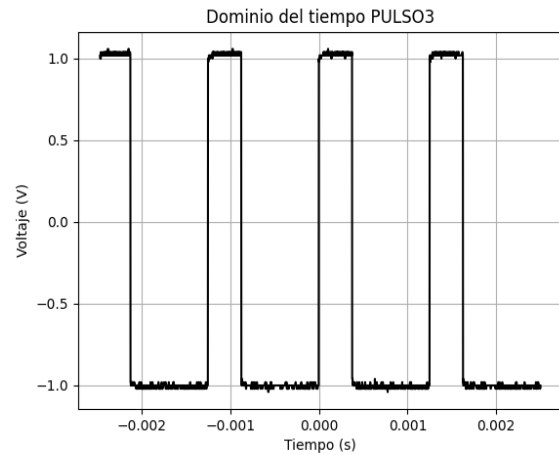
Grafica de Matlab con datos exportados



**Ilustración 78 señal pulso3 de archivo .CSV tiempo, Matlab**

Los niveles de tensión y el periodo coinciden correctamente. Se observa una buena coincidencia en la duración del pulso y su forma, lo que valida la calidad de la medición y la reconstrucción desde datos reales.

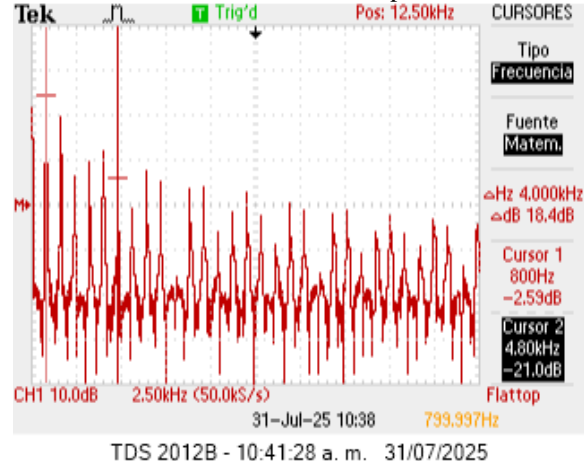
Phyton



**Ilustración 79 señal pulso 3 tiempo- Python**

La señal PULSO3 con ciclo útil del 30% representa un pulso asimétrico, más angosto que una onda cuadrada (50%) pero más ancho que PULSO2 (20%). La asimetría en el tiempo provoca la presencia de armónicos pares e impares en el espectro, como lo confirma la tabla de datos.

Dominio de la frecuencia osciloscopio



**Ilustración 80 señal pulso3 dominio frecuencia- osciloscopio**

Se observa un pico principal a 800 Hz (frecuencia fundamental), con una amplitud de  $-2.59$  dB, lo cual es coherente con los datos experimentales. A diferencia de señales cuadradas (duty 50%), aquí se presentan armónicos pares e impares, lo cual es esperable debido al ciclo útil diferente de 50%, que rompe la simetría temporal. La distribución de los armónicos es más densa y rica, con componentes destacadas en múltiplos de la fundamental (1600 Hz, 2400 Hz, etc.), y una caída progresiva en amplitud hacia frecuencias más altas.

Grafico con datos exportados a Matlab.

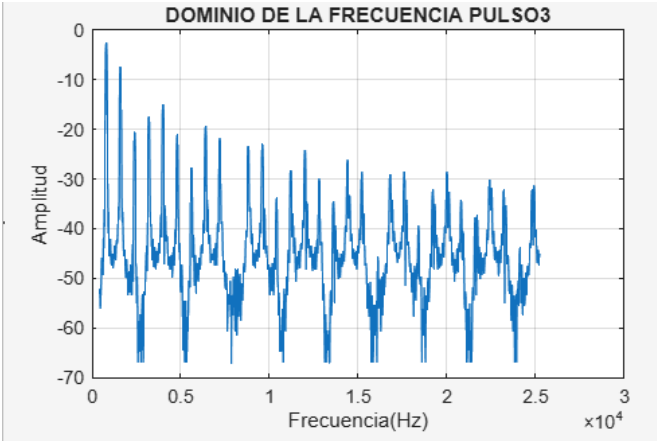


Ilustración 81 pulso3 en frecuencia desde .CSV-Matlab

Se observan picos armónicos bien definidos tanto pares como impares, lo que valida la pérdida de simetría típica de una señal cuadrada al usar un duty cycle  $\neq 50\%$ . La amplitud decrece progresivamente en cada armónico, pero no de forma uniforme, ya que los pulsos no son perfectamente cuadrados ni continuos. El fondo espectral (ruido o componentes menores) está por debajo de los  $-40$  dB, como en señales reales.

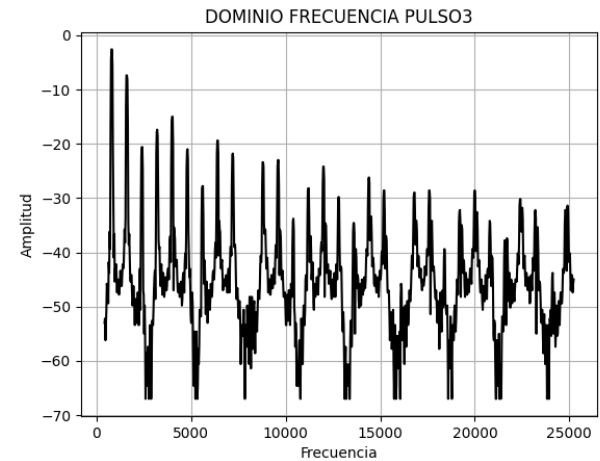


Ilustración 82 pulso3 en frecuencia desde .CSV-Python

La señal PULSO3 (duty cycle 30%) presenta un contenido espectral más complejo que señales cuadradas puras. Luego FFT

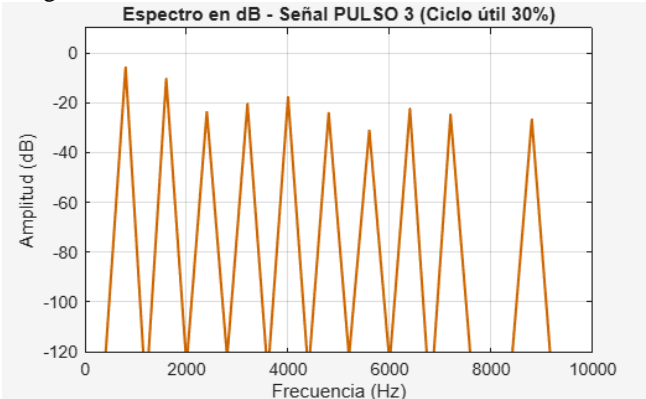


Ilustración 83 pulso 3 en frecuencia mediante FFT

A diferencia de una señal cuadrada pura, aquí se presentan armónicos pares e impares, lo cual confirma la asimetría de la señal (el ciclo útil no es 50%). La amplitud de los armónicos

disminuye progresivamente, aunque con menos regularidad, debido a la forma más estrecha y asimétrica del pulso. El espectro se extiende hasta los 10 kHz, mostrando una rica cantidad de armónicos que son necesarios para representar una señal pulsada con duty cycle  $\neq 50\%$ .

IFFT

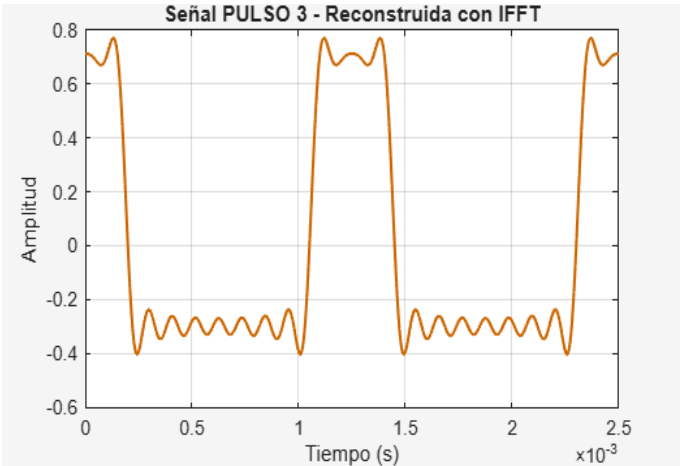


Ilustración 84 pulso 3 en dominio de tiempo mediante IFFT

La duración del pulso corresponde a un ciclo útil del 30% del periodo (1.25 ms), es decir, el pulso permanece encendido aproximadamente 0.375 ms. Se observan oscilaciones (efecto Gibbs) antes y después de cada pulso, lo cual es normal al reconstruir con una cantidad limitada de armónicos. El resto del tiempo la señal se mantiene en un valor cercano a  $-0.5$  V, reflejando el bajo promedio del resto del ciclo.

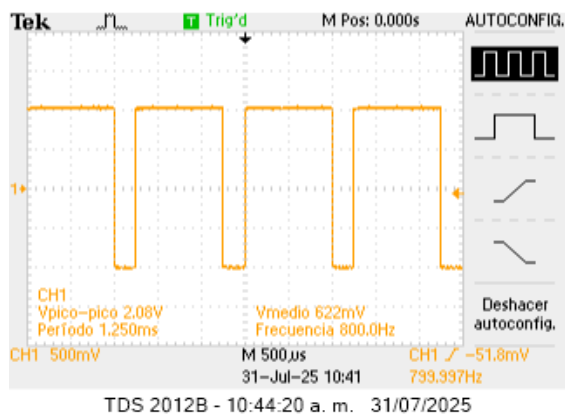
10. SEÑAL PULSO 4 80 % CICLO UTIL

Se llevara el mismo proceso que en el Puslo 1 , pulso2 y pulso 3 por lo cual la tabla del pulso 4 es la siguiente:

FRECUENCIA	800 Hz						
Vp	1						
DC	0						
SEÑAL	CICLO UTIL	ARMONICA	FRECUENCIA(HZ)	A			TEORICO
				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

Osciloscopio:

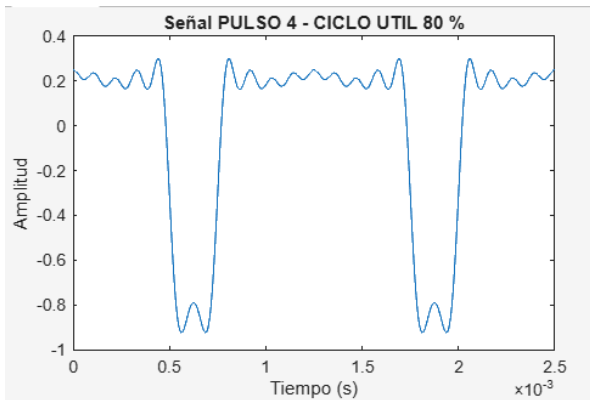




**Ilustración 85 señal pulso 4 dominio tiempo - osciloscopio**

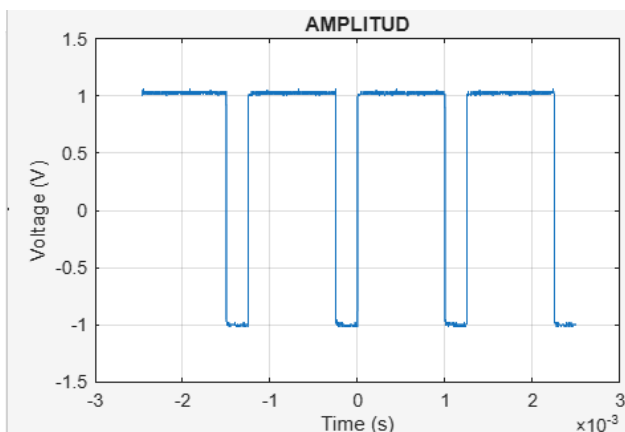
El tiempo de subida y bajada es casi instantáneo, típico de un generador de pulsos. Tiene  $V_{pico-pico}$  de 2.08 V, lo que concuerda con los cálculos de amplitud realizados en la tabla anterior.

Datos teóricos a Matlab



**Ilustración 86 señal pulso 4 dominio del tiempo- Matlab**

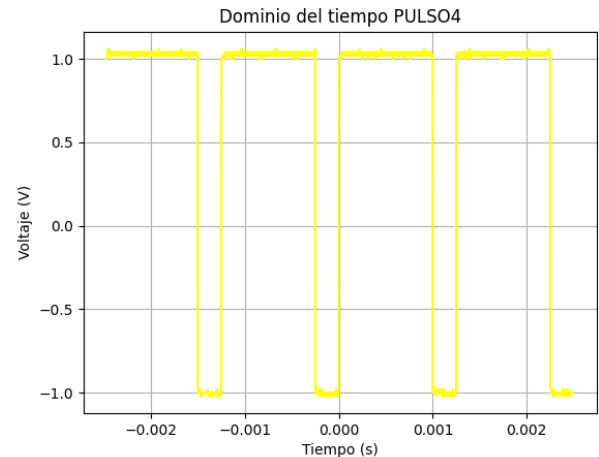
La simetría y espaciamento entre picos ratifican que la señal es periódica y su forma se aproxima a una onda cuadrada asimétrica. Datos exportados a Matlab



**Ilustración 87 señal pulso4 de archivo .CSV tiempo, Matlab**

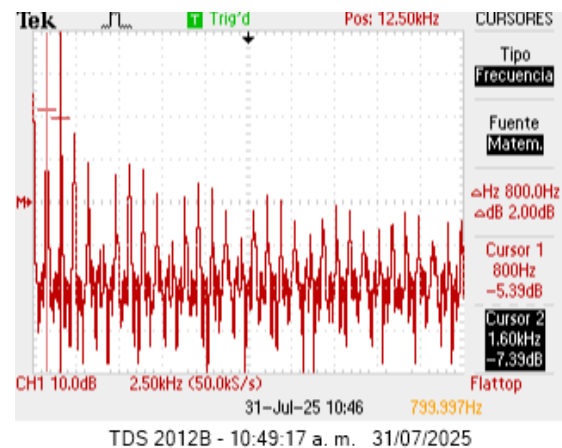
El nivel alto es más largo (80% del tiempo), y el nivel bajo es más corto (20%), lo cual concuerda con el ciclo útil del 80%. Las oscilaciones visibles cerca de los bordes son coherentes con el número limitado de armónicos utilizados para reconstruir la señal.

Phyton



**Ilustración 88 señal pulso4 de archivo .CSV tiempo- Python**

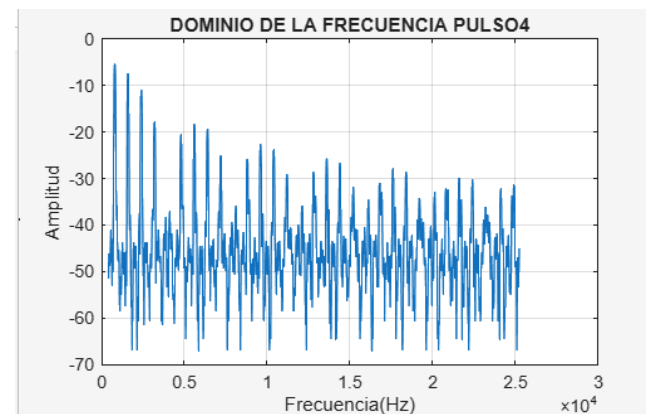
Osciloscopio en el dominio de la frecuencia



**Ilustración 89 señal pulso4 dominio frecuencia- osciloscopio**

El pico más alto corresponde a la frecuencia fundamental (800 Hz), como se esperaba. La diferencia de niveles entre la fundamental y los armónicos indica una forma de onda más cuadrada, típica de señales con ciclos útiles altos.

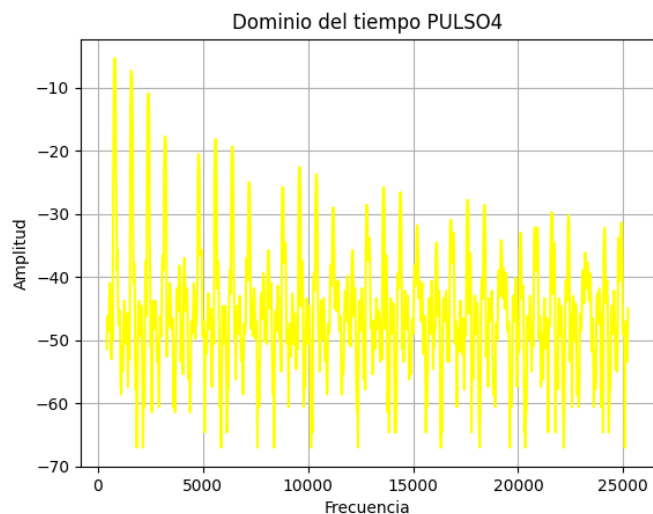
Matlab en el dominio de la frecuencia



**Ilustración 90 pulso 4 en frecuencia desde .CSV-Matlab**

Se observa una menor caída en la amplitud entre armónicos

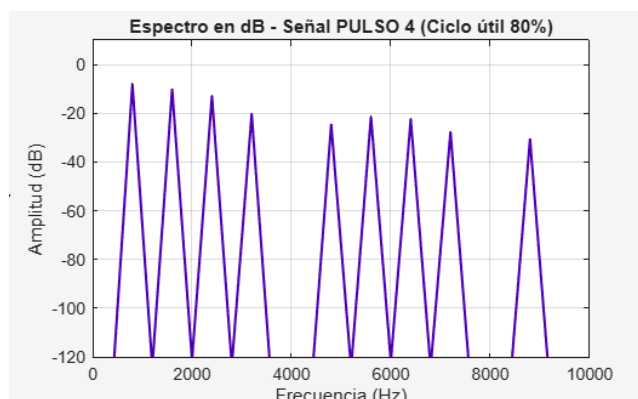
en comparación con señales con menor ciclo útil (como la del 20% o 30%). La densidad espectral es mayor, indicando más contenido de alta frecuencia, lo cual es típico en señales con cambios abruptos (bordes muy rectos).



**Ilustración 91 pulso 4 en frecuencia-Python**

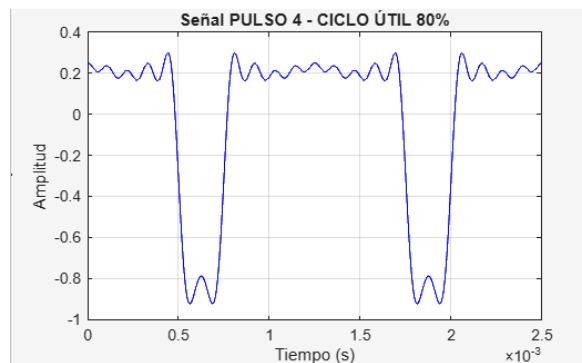
Las 3 gráficas coinciden en mostrar la estructura armónica típica de una señal de tipo pulso con un ciclo útil del 80%.

FFT



**Ilustración 92 pulso 4 en frecuencia mediante FFT**

La **frecuencia fundamental** es de 800 Hz, y los armónicos son múltiplos impares y pares (porque no es una onda simétrica). Las amplitudes no siguen una disminución perfecta como en el ciclo útil 50%, ya que el ciclo 80% provoca una distribución diferente del contenido armónico. A diferencia de un pulso cuadrado simétrico, en esta señal se presentan más **armónicos pares**, lo que indica que hay componentes de **asimetría en el tiempo**, producto del ciclo útil alto.



**Ilustración 93 pulso 4 en dominio de tiempo mediante IFFT**

La señal reconstruida muestra una onda cuadrada con un ciclo útil del 80%, es decir, el pulso permanece en nivel alto (positivo) durante el 80% del periodo y en bajo (negativo) el 20% restante. La forma de la señal es consistente con una reconstrucción por IFFT, reflejando el contenido armónico limitado: no es perfectamente. Esta señal es inversa a la de un ciclo útil del 20%: aquí el nivel alto domina el periodo.

Durante el análisis de las señales PULSO 1, PULSO 2, PULSO 3 y PULSO 4, se observaron diferencias importantes tanto en el dominio del tiempo como en el dominio de la frecuencia, originadas principalmente por el valor del ciclo útil de cada señal. A través del estudio de sus gráficas, espectros de frecuencia y reconstrucciones mediante IFFT, fue posible identificar el impacto que tiene el ciclo útil sobre el contenido armónico y la forma final de la señal.

La señal PULSO 1, con un ciclo útil del 20%, presentó pulsos angostos y transiciones muy rápidas en el tiempo. Como consecuencia, su espectro de frecuencia mostró un alto contenido de armónicos, especialmente en frecuencias altas, lo cual indica una señal con gran riqueza espectral. La reconstrucción con IFFT reveló una forma de onda con mucho ripple y oscilaciones, típicas de señales con transiciones abruptas.

En contraste, PULSO 2 con un ciclo útil del 50%, mostró una señal cuadrada perfectamente simétrica. Esta configuración generó un espectro con solo armónicos impares y una distribución armónica predecible. Es el caso más "ideal" de onda cuadrada, lo que se reflejó en una reconstrucción suave y clara por IFFT, con transiciones bien definidas y menos distorsión.

La señal PULSO 3, con un ciclo útil del 30%, se comportó como un punto intermedio entre los casos anteriores. Su forma en el dominio del tiempo mostró una parte alta más corta y una parte baja más prolongada. En frecuencia, hubo presencia de armónicos pares e impares, con una amplitud que disminuía de forma menos simétrica que en el pulso 2. La reconstrucción evidenció un nivel moderado de distorsión, resultado del equilibrio parcial entre la duración del pulso y la energía espectral.

Finalmente, PULSO 4, con un ciclo útil del 80%, representó una señal en la que el tiempo en estado alto predominaba ampliamente sobre el tiempo en estado bajo. Esta asimetría se reflejó en un espectro con armónicos tanto pares como impares, y con una distribución de amplitud más compleja. La

IFFT mostró una forma de onda con más componente continua, menor contenido en alta frecuencia comparado con el pulso 1, pero aún con bordes marcados. Su forma se asemeja a una onda cuadrada invertida respecto al pulso 1.

## CONCLUSION

Durante el desarrollo del laboratorio se cumplió integralmente con los objetivos planteados en la guía, al analizar diez señales periódicas —senoidales, triangulares, cuadradas y de pulso— desde las perspectivas del tiempo y la frecuencia, utilizando herramientas prácticas como el osciloscopio digital y computacionales como MATLAB y Python (Thonny). Se capturaron señales con distintos valores de componente continua (DC) y ciclos útiles, exportando los datos en formato .csv y activando la función MATH/FFT del osciloscopio para visualizar los primeros armónicos, tal como indicaba el procedimiento. A partir de estos datos, se implementaron programas para graficar las señales en ambos dominios y reconstruirlas mediante IFFT, comparando los resultados con las señales generadas teóricamente usando coeficientes de Fourier. En este proceso se observaron características particulares en cada señal: las senoidales mostraron un espectro limpio con un solo armónico (SENO 1) o con desplazamiento vertical al tener componente DC (SENO 2); las triangulares presentaron armónicos impares decrecientes y efecto Gibbs en la reconstrucción (TRIANGULAR 1 y 2); las cuadradas evidenciaron armónicos impares con y sin desplazamiento vertical (CUADRADA 1 y 2); y las señales de pulso variaron su complejidad espectral dependiendo del ciclo útil —la PULSO 1 fue simétrica con armónicos impares, mientras que las PULSO 2, 3 y 4 mostraron espectros más ricos en armónicos pares e impares, dependiendo de su asimetría y duración del pulso—. En conjunto, este laboratorio permitió confirmar los fundamentos del procesamiento digital de señales, evidenciar la relación entre la forma de onda, el contenido espectral y los parámetros de adquisición, y validar la fidelidad de las reconstrucciones aplicando correctamente la FFT y la IFFT.

## REFERENCIAS

- [1] Banzhaf, W. (s.f.). *Lab 3 - Using the Agilent 54621A Digital Oscilloscope as a Spectrum Analyzer*. University of Hartford, Ward College of Technology.  
Recuperado del documento: *Análisis de señales (1) (1).pdf*
- [2] University of Rhode Island, Department of Electrical and Computer Engineering. (s.f.). *FFT Tutorial - ELE 436: Communication Systems*.  
Recuperado del documento: *fft\_matlab.pdf*
- [3] MathWorks. (2024). *MATLAB Documentation*. The MathWorks, Inc.  
<https://www.mathworks.com/help/matlab/>
- [4] [4] Oppenheim, A. V., & Schaffer, R. W. (2009). *Discrete-Time Signal Processing* (3.<sup>a</sup> ed.). Prentice Hall.  
Recuperado del documento: *INFORMES DE LABORATORIOS.pdf*
- [5] [5] Lathi, B. P. (2005). *Linear Systems and Signals* (2.<sup>a</sup> ed.). Oxford University Press.  
Referencia citada en el desarrollo teórico de señales periódicas.
- [6] [6] Universidad Nacional de Colombia. (2022). *Análisis espectral de señales periódicas usando series de Fourier y transformadas*. Curso de Señales y Sistemas.

Disponible en:  
<https://www.virtual.unal.edu.co/cursos/ingenieria/4025051/pdf>  
[7] Rawat, T. K. (2015). *Signal Processing and Linear Systems*. Oxford University Press.  
Citado como bibliografía base para el análisis de señales en frecuencia.

## REPOSITORIO GITHUB

- <https://github.com/bejaranoximena12/SEGUNDO-INFORME-COMUNICACIONES-DIGITALES.git>