

Experimento Phyphox

Belky Valentina Giron Lopez
est.belky.giron@unimilitar.edu.co
 Docente: José De Jesús Rúgeles

Resumen— En este informe se capturaron distintas señales periódicas (senoidales, cuadradas, triangulares y pulsos) empleando un osciloscopio digital, registrando su comportamiento en el dominio del tiempo y de la frecuencia. Se utilizaron las funciones MATH/FFT del instrumento para obtener los espectros y posteriormente se analizaron los primeros armónicos tanto de forma experimental como teórica, aplicando las series de Fourier. Además, se desarrollaron programas en Matlab y Python para visualizar señales, calcular la FFT e IFFT y reconstruir las formas de onda a partir de armónicos teóricos y experimentales. Finalmente, se compararon las señales reconstruidas con las originales, evaluando la precisión de los métodos empleados y destacando la importancia del análisis espectral en el procesamiento de señales para comunicaciones.

Abstract— In this report, different periodic signals (sine, square, triangular, and pulse) were captured using a digital oscilloscope, recording their behavior in the time and frequency domains. The instrument's MATH/FFT functions were used to obtain the spectra, and the first harmonics were subsequently analyzed both experimentally and theoretically, applying Fourier series. In addition, programs in Matlab and Python were developed to visualize signals, calculate the FFT and IFFT, and reconstruct waveforms from theoretical and experimental harmonics. Finally, the reconstructed signals were compared with the originals, evaluating the accuracy of the methods used and highlighting the importance of spectral analysis in signal processing for communications.

I. INTRODUCCIÓN

En este informe el procesamiento de señales es muy importante, ya que nos permite analizar, modificar y representar la información de manera eficiente tanto en el dominio del tiempo como en el de la frecuencia. Este laboratorio nos permitió familiarizarnos con los instrumentos, softwares de programación y técnicas necesarias para el estudio de señales, utilizando el generador de señales para transmitir la señal y el osciloscopio para la captura y análisis de las diferentes formas de onda. Asimismo, se abordó la aplicación de la Transformada Rápida de Fourier (FFT) y su inversa (IFFT), así como el cálculo teórico y experimental de armónicos a través de las series de Fourier.

Frecuencia	2.5 KHz
Vp	3

Tabla. 1. Características de la señal.

Para los experimentos se utilizaron las características

mencionadas en la tabla 1. Esto para la generación de las señales.

II. DESARROLLO DE EXPERIMENTOS

1. Señal seno 1

Primero configuramos la señal en el generados de señales con las características mencionadas en la Tabla 1. Y luego trasmitimos la señal al osciloscopio mediante la sonda, esto se puede observar en la siguiente ilustración:

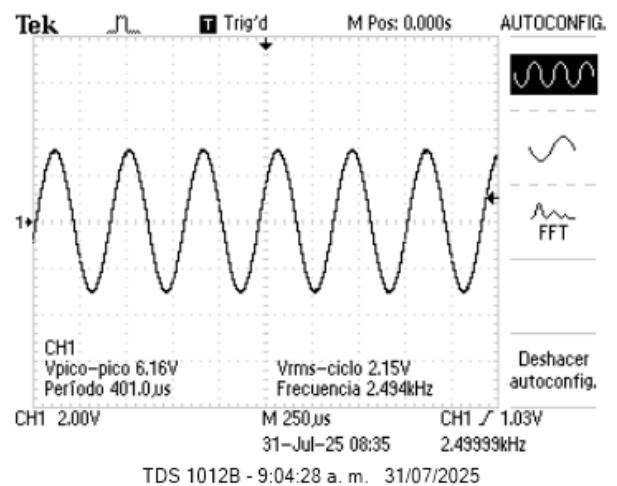


Fig. 1. Señal en dominio del tiempo.

Teniendo en cuenta la fig. 1 logramos identificar las siguientes características de la señal:

Vpp	6.16 V
T	401 µs
Vrms	2.15 V
f	2.49 KHz
M	250 µs

Tabla. 2. Características de la señal dominio del tiempo.

Luego configuramos el osciloscopio para obtener la señal en el dominio de la frecuencia, como se muestra a continuación:

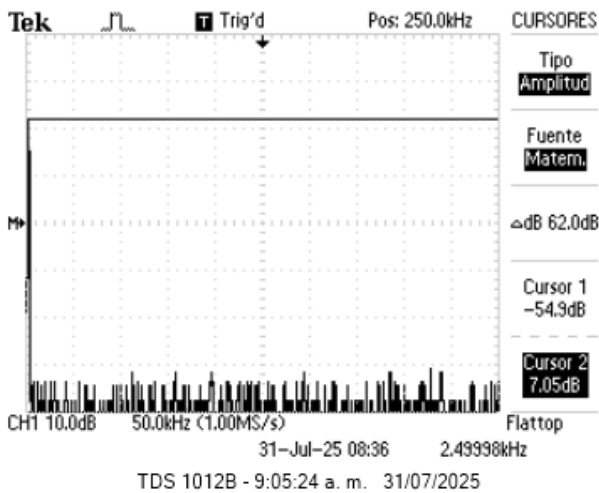


Fig. 2. Señal en el dominio de la frecuencia.

Teniendo en cuenta la fig. 2 logramos identificar las siguientes características de la señal:

Tipo	Amplitud
Fuente	Math
Δ dB	62.0 dB
f	2.49 KHz
fmax	50.0 KHz
Tasa de muestreo	100MS/s

Tabla. 3. Características de la señal dominio de la frecuencia.

Después de obtener estos datos exportaremos los archivos .csv de la señal en dominio del tiempo y de la frecuencia, estos datos nos permitirán mediante código de Python y Matlab hallar la fft, ifft, reconstruir la señal y además nos permitirá comparar con datos teóricos.

```

1 % Obtener señal en el tiempo de archivo .csv
2 a0 = senoitiempo.E1;
3 t0 = senoitiempo.E3;
4 % Obtener fft de la señal en tiempo
5 L = length(a0); % Longitud de la señal
6 A0 = fft(a0);
7 E2 = abs(A0/L); % Espectro de dos lados
8 E1 = E2(1:L/2+1); % Espectro de un solo lado
9 E1(2:end-1) = 2*E1(2:end-1); % Ajustar para simetria
10 f0 = (0:(L/2))/L; % vector de frecuencia
11 %Obtener señal en frecuencia de archivo .csv
12 A1 = senoifft.VarNames;
13 f1 = senoifft.VarName4;
14 %Reconstrucción de la señal en tiempo
15 A1V = sqrt(2)*10.^(A1/20); %Conversión de dB a V
16 N = length(f1);
17 A1_aprox = A1V .* exp(1j * zeros(size(N,1))); % fase cero
18 A1_full = [A1_aprox; flipud(conj(A1_aprox(2:end-1)))]; % hacer simétrica
19 a1 = ifft(A1_full, 'symmetric');
20 fs = max(f1)*2;
21 t1 = (0:length(A1_full)-1)/fs;
22 %Graficar
23 subplot(4,1,1);
24 plot(t0,a0);
25 title 'Señal en Tiempo (.csv)'
26 grid on
27 xlabel 'Tiempo [s]'
28 ylabel 'Amplitud [V]'
29
30 subplot(4,1,2);
31 plot(f0, E1);
32 title 'Señal en Frecuencia (Construida)'
33 grid on
34 xlabel 'Frecuencia [Hz]'
35 ylabel 'Amplitud'
36
37 subplot(4,1,3);
38 plot(f1,A1);
39 title 'Señal en Frecuencia (.csv)'
40 grid on
41 xlabel 'Frecuencia [Hz]'
42 ylabel 'Amplitud [dB]'
43
44 subplot(4,1,4);
45 plot(t1,a1);
46 title 'Señal en Tiempo (Reconstruida)'
47 grid on
48 xlabel 'Tiempo'
49 ylabel 'Amplitud'

```

Fig. 3. Código matlab.

Según la fig. 3 implementamos un código en Matlab para la visualización de la señal en dominio del tiempo, de la frecuencia, la señal construida y reconstruida.

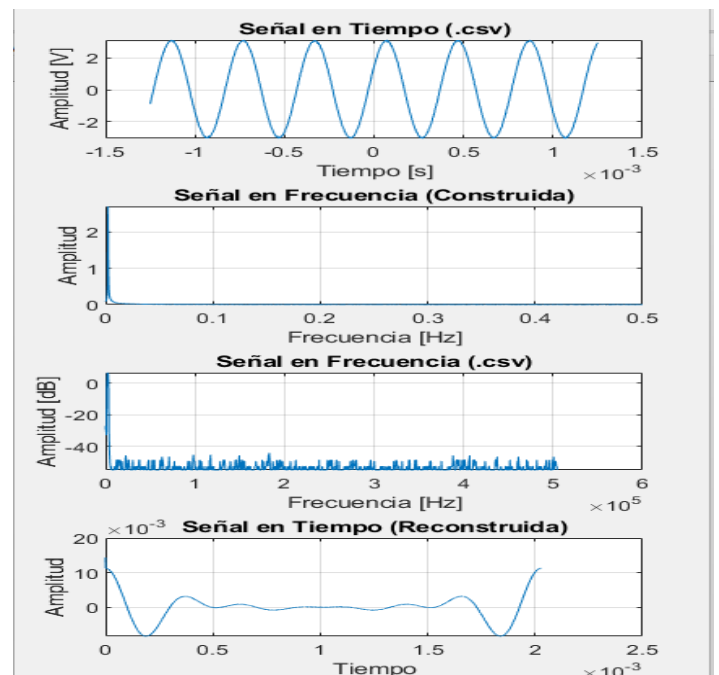


Fig. 4. Resultado.

Luego de la ejecución del código se muestra en la fig. 4 el resultado de este, podemos observar que las gráficas .csv son similares al resultado obtenido de manera experimental en el osciloscopio, la señal construida hace referencia a la transformada rápida de Fourier, la cual va de acuerdo a los datos obtenidos, por otro lado la señal reconstruida que sería la IFFT sale de manera errónea ya que no están los datos suficientes para calcularla.

Luego realizamos la señal seno en Python mediante el siguiente código:

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

# Ruta al archivo
archivo = 'seno1_tiempo.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]
voltaje = datos.iloc[:, 4]

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

Fig. 5. Código Python señal seno.

Al ejecutar el código obtenemos la siguiente gráfica:

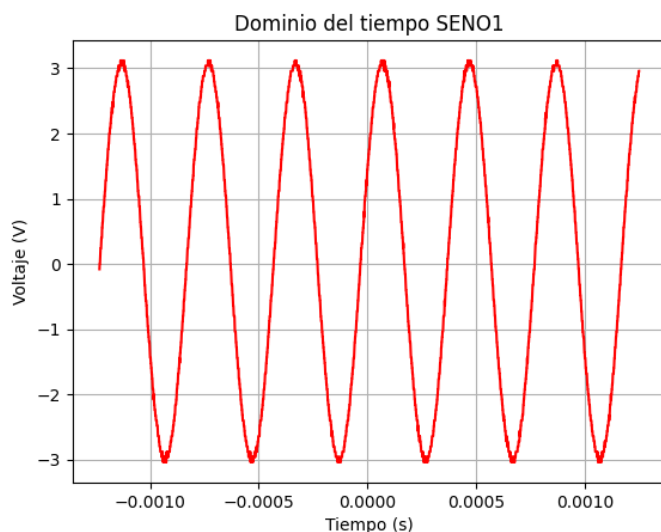


Fig. 6. Resultado Python señal seno.

Podemos concluir que la grafica obtenida se asemeja a la obtenida de manera experimental en el osciloscopio, junto con los valores asignados y vistos.

Ahora mediante este código graficaremos la señal en dominio de la frecuencia:

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

# Ruta al archivo
archivo = '/content/seno1_fft.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
frecuencia = datos.iloc[:, 3]
magnitud = datos.iloc[:, 4]

# Graficar
plt.plot(frecuencia, magnitud, color='blue')
plt.title('Espectro de la señal .csv')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('Magnitud (dB)')
plt.grid(True)
plt.show()
```

Fig. 7. Código Python señal seno.

Al ejecutar el código obtenemos la siguiente gráfica:

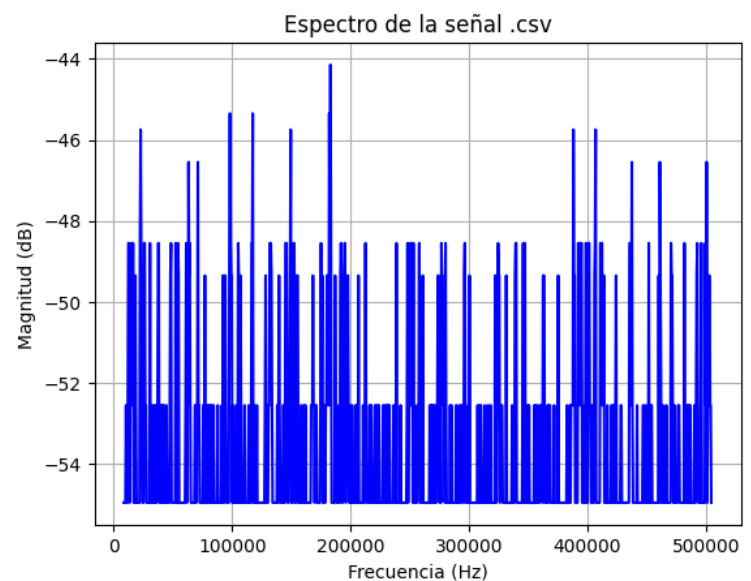


Fig. 8. Resultado Python señal seno.

El espectro de la señal se muestra en los rangos obtenidos experimentalmente y además al compararla con la de Matlab nos damos cuenta de que son similares.

Por último, graficamos la señal construida mediante el archivo .csv en el código realizamos la transformada rápida de Fourier.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft, fftfreq

# Cargar datos
datos = pd.read_csv('seno1_tiempo.csv', skiprows=17, header=None, encoding='latin1')
t, v = datos.iloc[:, 3].values, datos.iloc[:, 4].values # Tiempo (s), Voltaje (V)

# Calcular FFT
N = len(v)
dt = t[1] - t[0]
fs = 1/dt
freqs = fftfreq(N, dt)[:N//2]
fft_val = fft(v)[:N//2]
magnitud = np.abs(fft_val)/N # Normalizada
magnitud[1:] *= 2 # Compensar frecuencias negativas (excepto DC)

# Graficar FFT
plt.figure(figsize=(10, 4))
plt.plot(freqs, magnitud)
plt.title('Señal de Frecuencia (construida)')
plt.xlabel('Frecuencia (Hz)')
plt.ylabel('Amplitud (V)')
plt.grid(True)
plt.xlim(0, fs/2)
plt.tight_layout()
plt.show()

```

Fig. 9. Código Python señal seno 1.

De acuerdo con el código anterior al ejecutarlo nos arroja la siguiente gráfica:

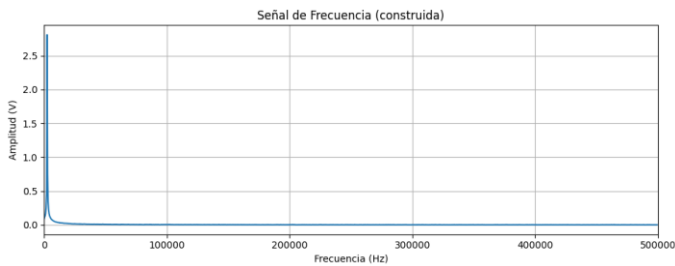


Fig. 10. Resultado Python señal seno 1.

La gráfica muestra un espectro de frecuencia, donde el eje X representa la frecuencia (Hz) y el eje Y la amplitud (V).

En este caso, se observa que:

- La mayor amplitud se concentra en las frecuencias bajas, con un pico inicial cercano a los 2.7 V.
- A medida que la frecuencia aumenta, la amplitud decae rápidamente y se mantiene cercana a cero.
- Esto indica que la señal está compuesta principalmente por componentes de baja frecuencia, mientras que las altas frecuencias tienen una contribución mínima.

2. Señal Seno 2

Frecuencia	2.5 KHz
Vp	3
DC	1

Tabla. 4. Características de la señal.

Primero configuramos la señal en el generados de señales con las características mencionadas en la Tabla 4. Y luego transmitimos la señal al osciloscopio mediante la sonda, esto se puede observar en la siguiente ilustración:

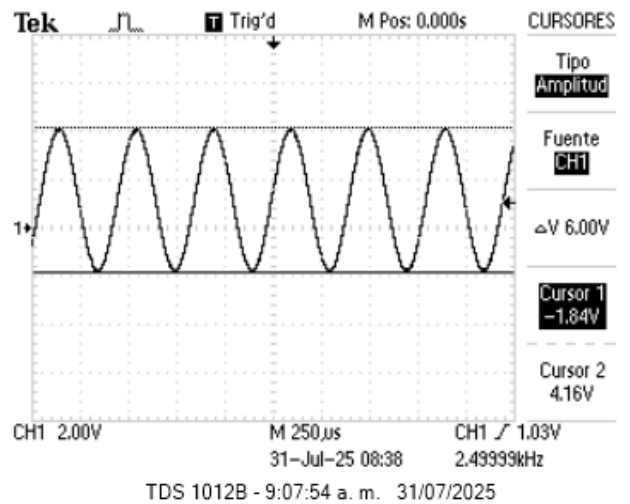


Fig. 11. Osciloscopio señal seno 2.

Teniendo en cuenta la fig. 11 logramos identificar las siguientes características de la señal:

Vpp	6 V
T	400 µs
f	2.49 KHz
M	250 µs

De acuerdo con las posiciones de los cursores nos damos cuenta de que el valor mínimo es de - 1.84 V y el máximo es de 4.16 V lo cual hace referencia a nuestro nivel de D.C mediante la siguiente operación:

$$\frac{4,16 + (-1,84)}{2} \approx 1,16 \text{ V}$$

La diferencia de 0.16 V en el nivel de DC se puede deber a la precisión de lectora del osciloscopio, a la tolerancia de la sonda o ruido.

Luego configuramos el osciloscopio para obtener la señal en el dominio de la frecuencia, como se muestra a continuación:

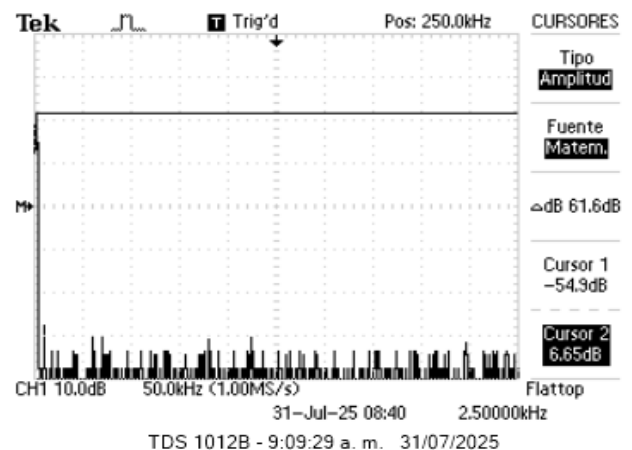


Fig. 12. Osciloscopio dominio de la frecuencia señal seno 2.

En la figura 12 se muestra el espectro de la señal (modo FFT del osciloscopio).

En la imagen se observa:

- Eje horizontal (frecuencia)

Va desde 0 Hz hasta aproximadamente 2,5 MHz, también se muestra cada pico de frecuencia el cual esta presente en la señal.

- Eje vertical (amplitud en dB)

Está en una escala logarítmica de decibelios (dB) y el valor más alto en la gráfica es de 61,6 dB, mostrado en la parte superior (Cursor 2).

- Pico en DC (0 Hz)

El pico que aparece al inicio (cerca de 0 Hz) corresponde al componente DC de la señal.

- Otros picos pequeños

Son armónicos o ruido, en la señal senoidal pura deben ser muy bajos, como se percibe en la figura 12.

Después de obtener estos datos exportaremos los archivos .csv de la señal en dominio del tiempo y de la frecuencia, estos datos nos permitirán mediante código de Python y Matlab hallar la fft, ifft, reconstruir la señal y además nos permitirá comparar con datos teóricos.

```

1 % Obtener señal en el tiempo de archivo .csv
2 a0 = seno2tiempo.E1;
3 t0 = seno2tiempo.E3;
4 % Obtener fft de la señal en tiempo
5 L = length(a0); % Longitud de la señal
6 A0 = fft(a0);
7 E2 = abs(A0/L); % Espectro de dos lados
8 E1 = E2(1:L/2+1); % Espectro de un solo lado
9 E1(2:end-1) = 2*E1(2:end-1); % Ajustar para simetría
10 f0 = (0:(L/2))/L; % vector de frecuencia
11 % Obtener señal en frecuencia de archivo .csv
12 A1 = seno2fft.VarName5;
13 f1 = seno2fft.VarName4;
14 % Reconstrucción de la señal en tiempo
15 A1V = sqrt(2)*10.^(A1/20); % Conversión de dB a V
16 N = length(f1);
17 A1_aprox = A1V .* exp(1j * zeros(size(N,1))); % fase cero
18 A1_full = [A1_aprox; flipud(conj(A1_aprox(2:end-1)))]; % hacer simétrica
19 a1 = ifft(A1_full, 'symmetric');
20 fs = max(f1)*2;
21 t1 = (0:length(A1_full)-1)/fs;
22 % Graficar
23 subplot(4,1,1);
24 plot(t0,a0);
25 title('Señal en Tiempo (.csv)');
26 grid minor
27 xlabel('Tiempo [s]');
28 ylabel('Amplitud [V]');
29
30 subplot(4,1,2);
31 plot(f0,E1);
32 title('Señal en Frecuencia (Construida)');
33 grid minor
34 xlabel('Frecuencia [Hz]');
35 ylabel('Amplitud');
36
37 subplot(4,1,3);
38 plot(f1,A1);
39 title('Señal en Frecuencia (.csv)');
40 grid minor
41 xlabel('Frecuencia [Hz]');
42 ylabel('Amplitud [dB]');
43
44 subplot(4,1,4);
45 plot(t1,a1);
46 title('Señal en Tiempo (Reconstruida)');
47 grid minor
48 xlabel('Tiempo');
49 ylabel('Amplitud');

```

Fig. 13. Código implementado en Matlab para la graficación.

A partir de correr este código obtendremos las siguientes graficas:

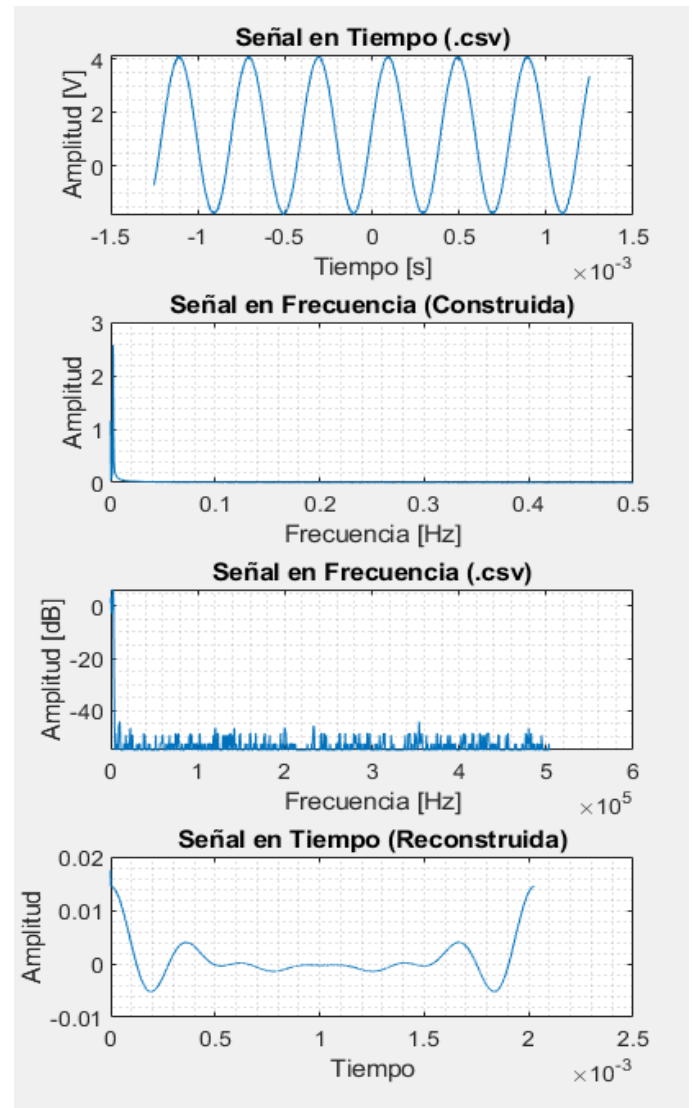


Fig. 14. Gráficas Matlab señal seno 2.

Luego de la ejecución del código se muestra en la fig. 14 el resultado de este, podemos observar que las gráficas .csv son similares al resultado obtenido de manera experimental en el osciloscopio, la señal construida hace referencia a la transformada rápida de Fourier, la cual va de acuerdo con los datos obtenidos, por otro lado, la señal reconstruida que sería la IFFT sale de manera errónea ya que no están los datos suficientes para calcularla.

Luego realizamos la señal seno en Python mediante el siguiente código:

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

# Ruta al archivo
archivo = 'seno2_tiempo.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]
voltaje = datos.iloc[:, 4]

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

Fig. 15. Código implementado en Python para la graficar.

Al ejecutar el código obtenemos la siguiente gráfica:

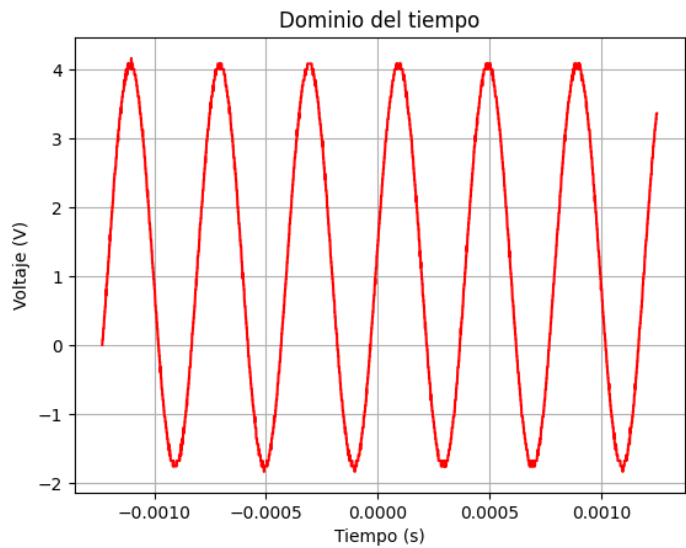


Fig. 16. Grafica en Python señal seno 2.

Podemos concluir que la gráfica obtenida se asemeja a la obtenida de manera experimental en el osciloscopio, junto con los valores asignados y vistos.

Para graficar el espectro en dominio de la frecuencia usamos el mismo código del dominio del tiempo y lo único que cambiamos es el archivo copiamos la ruta del archivo fft de la señal en .csv

Obtuvimos la siguiente gráfica:

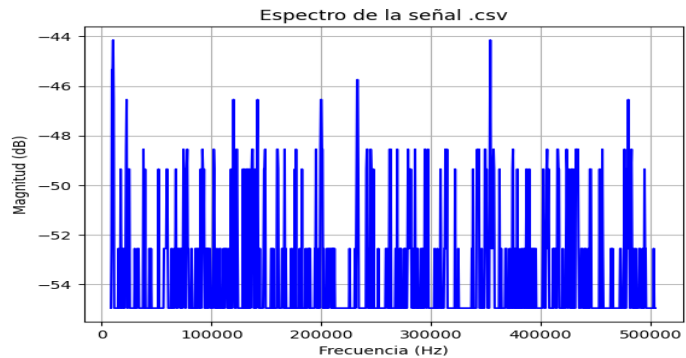


Fig. 17. Grafica en Python señal seno 2 dominio de la frecuencia.

De acuerdo con el código de la figura 9. Realizamos el cambio del archivo en este caso sería el de la señal seno 2 en el dominio del tiempo y aparece la siguiente gráfica:

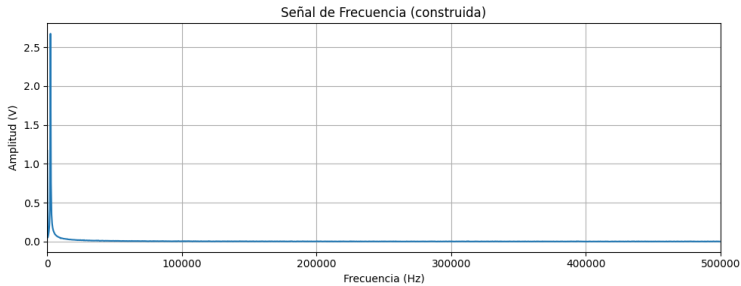


Fig. 18. Grafica en Python señal seno 2 FFT.

espectro de la señal se muestra en los rangos obtenidos experimentalmente y además al compararla con la de Matlab nos damos cuenta de que son similares.

- Eje X (Frecuencia en Hz)

Va de 0 Hz a 500 kHz y el 1 primer punto (cerca de 0 Hz) es donde aparece el componente DC.

- Eje Y (Amplitud en V)

Muestra la magnitud de cada componente de frecuencia.

- Comportamiento

Después del pico inicial, la amplitud cae rápidamente y se mantiene muy baja en el resto del espectro.

3. Señal Triangular 1

Frecuencia	2.5 KHz
Vp	3
DC	0

Tabla. 5. Características de la señal.

Fk	2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5
dB	4.7	-14.5	-22.5	-29.3	-33.7	-35.7	-39.3	-39.3
V	2.42	0.266	0.106	0.048	0.029	0.023	0.015	0.015

Tabla. 6. Características de la señal dominio de la frecuencia.

A partir de la figura 20. Encontramos los datos de la tabla 6. Además, a partir de ello podemos calcular los coeficientes teóricos de la serie de Fourier y así poder comparar los resultados obtenidos, esto nos permite reconstruir la señal mediante esta fórmula: Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

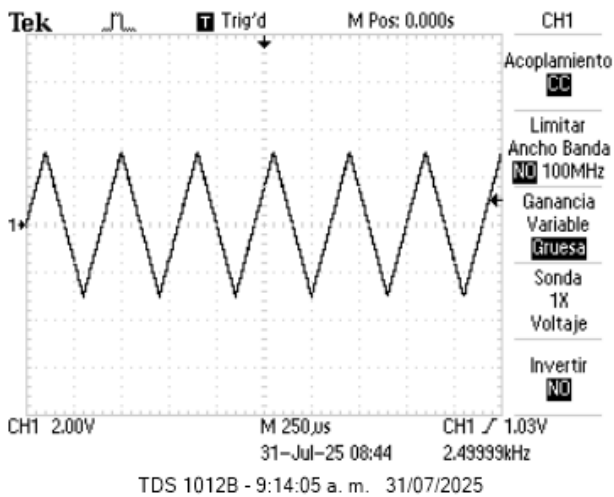


Fig. 19. Grafica en osciloscopio señal triangular 1.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

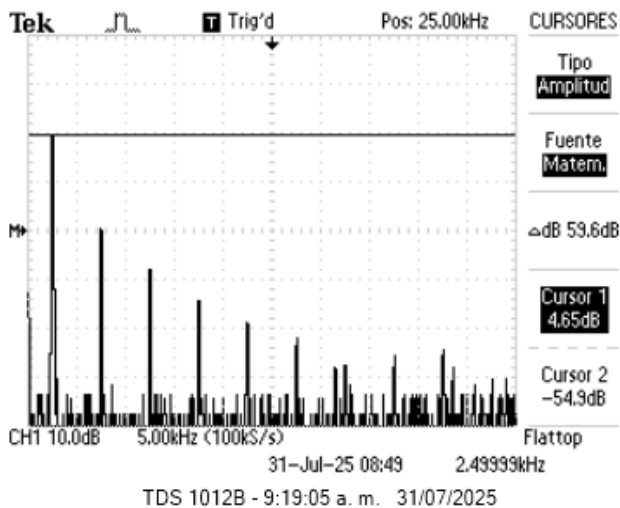


Fig. 20. Grafica en osciloscopio señal triangular 1 dominio de la frecuencia.

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

Luego de obtener los valores teoricos realizamos, un codigo en matlab para visualizar la señal tringular, su espectro en el dominio de la frecuencia, la FFT y la reconstruccion de la señal a partir de los datos que calculamos.

```

1 % Reconstrucción de la señal con series de fourier
2 f = 2.5e3;
3 T = 1/f;
4 t = -1e-3:T/100:1.5e-3;
5 x1 = 2.432;
6 x3 = 0.27;
7 x5 = 9.727e-2;
8 x7 = 4.963e-2;
9 x9 = 3e-2;
10 x11 = 2.01e-2;
11 x13 = 1.439e-2;
12 x15 = 1.081e-2;
13 V1 = x1*cos(2*pi*f*t);
14 V3 = x3*cos(3*2*pi*f*t);
15 V5 = x5*cos(5*2*pi*f*t);
16 V7 = x7*cos(7*2*pi*f*t);
17 V9 = x9*cos(9*2*pi*f*t);
18 V11 = x11*cos(11*2*pi*f*t);
19 V13 = x13*cos(13*2*pi*f*t);
20 V15 = x15*cos(15*2*pi*f*t);
21 vt = V1+V3+V5+V7+V9+V11+V13+V15;
22
23 D = length(vt); % Longitud de la señal
24 VT = fft(vt);
25 VT2 = abs(VT/D); % Espectro de dos lados
26 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
27 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
28 fvt = (0:(D/2))/D; % vector de frecuencia

```

Fig. 21. Código de Matlab señal triangular 1.

Luego de ejecutar el código obtenemos la siguiente gráfica:

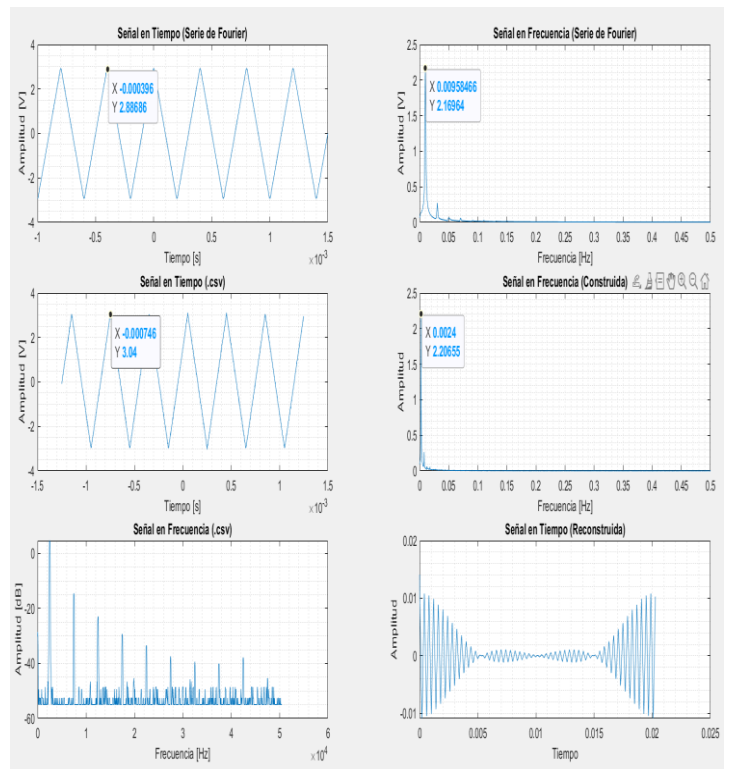


Fig. 22. Graficas obtenidas en Matlab.

Las graficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

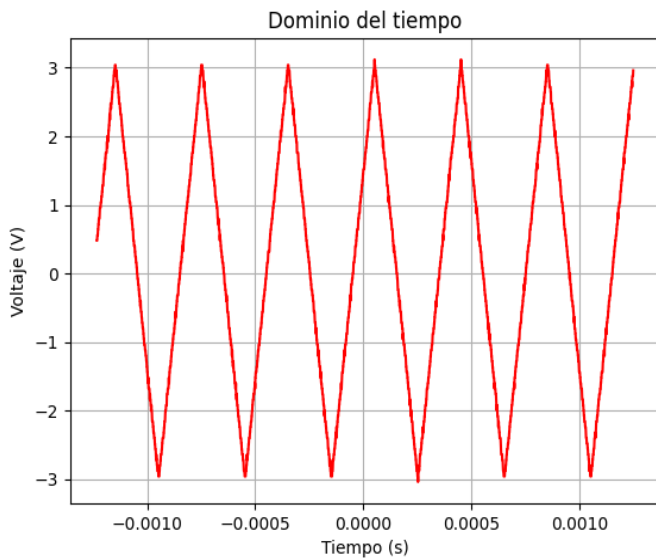


Fig. 23. Grafica Python dominio del tiempo señal triangular 1.

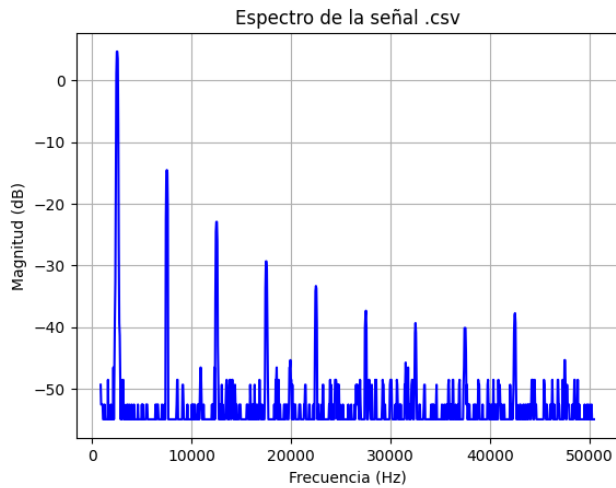


Fig. 24. Grafica Python dominio de la frecuencia señal triangular 1.

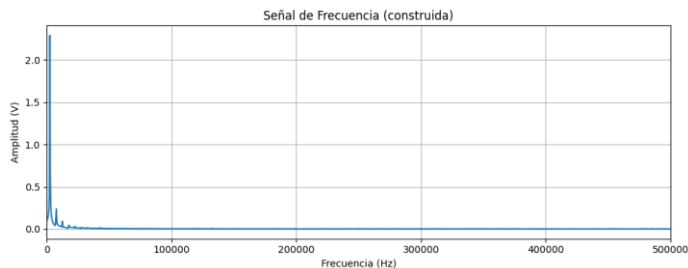


Fig. 25. Grafica Python FFT señal triangular 1.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y los datos exportados nos permiten compararla de manera teórica.

4. Señal Triangular 2

Frecuencia	2.5 KHz
Vp	3
DC	1.5

Tabla. 7. Características de la señal.

Fk	2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5
dB	4.7	-14.5	-22.5	-28.9	-34.9	-38.1	-39.4	-41.7
V	2.42	0.266	0.106	0.051	0.025	0.018	0.015	0.012

Tabla. 8. Características de la señal dominio de la frecuencia experimental.

A partir de los datos experimentales obtenidos realizamos el cálculo teórico mediante el software de Excel:

Componente Exp	2,429	0,266	0,106	0,051	0,025	0,018	0,015	0,012
Voltaje	3	3	3	3	3	3	3	3
n	1	3	5	7	9	11	13	15
pi*n	3,1415927	9,424778	15,70796	21,99115	28,27433	34,55752	40,8407	47,12389
Componente teori	2,432	0,270	0,097	0,050	0,030	0,020	0,014	0,011

Fig. 26. Calculo teórico.

Podemos concluir que entre los datos experimentales y teóricos existe un porcentaje de error pequeño y esto se puede deber a la tolerancia de los instrumentos usados.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

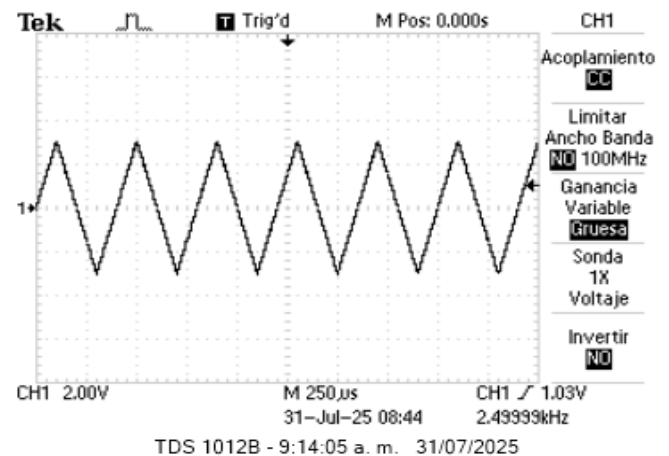


Fig. 27. Grafica en osciloscopio señal triangular 2.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

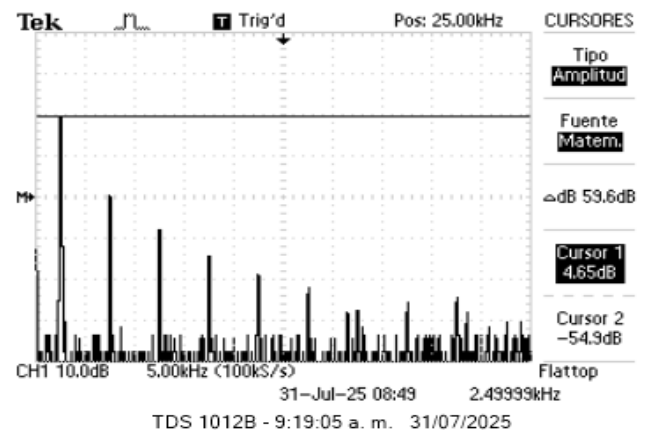


Fig. 28. Grafica en osciloscopio señal triangular 2 dominio de la frecuencia.

Luego de obtener los valores teóricos realizamos, un código en matlab para visualizar la señal triangular, su espectro en el dominio de la frecuencia, la FFT y la reconstrucción de la señal a partir de los datos que calculamos

```

1  % Reconstrucción de la señal con series de fourier
2  f = 2.5e3;
3  T = 1/f;
4  t = -1e-3:T/100:1.5e-3;
5  x1 = 2.432;
6  x3 = 0.27;
7  x5 = 9.727e-2;
8  x7 = 4.963e-2;
9  x9 = 3e-2;
10 x11 = 2.01e-2;
11 x13 = 1.439e-2;
12 x15 = 1.081e-2;
13 V1 = x1*cos(2*pi*f*t);
14 V3 = x3*cos(3*2*pi*f*t);
15 V5 = x5*cos(5*2*pi*f*t);
16 V7 = x7*cos(7*2*pi*f*t);
17 V9 = x9*cos(9*2*pi*f*t);
18 V11 = x11*cos(11*2*pi*f*t);
19 V13 = x13*cos(13*2*pi*f*t);
20 V15 = x15*cos(15*2*pi*f*t);
21 DC = 1.5;
22 vt = V1+V3+V5+V7+V9+V11+V13+V15+DC;
23
24 D = length(vt); % Longitud de la señal
25 VT = fft(vt);
26 VT2 = abs(VT/D); % Espectro de dos lados
27 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
28 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
29 fvt = (0:(D/2))/D; % vector de frecuencia

```

Fig. 29. Código Matlab.

Luego de ejecutar el código obtenemos la siguiente gráfica:

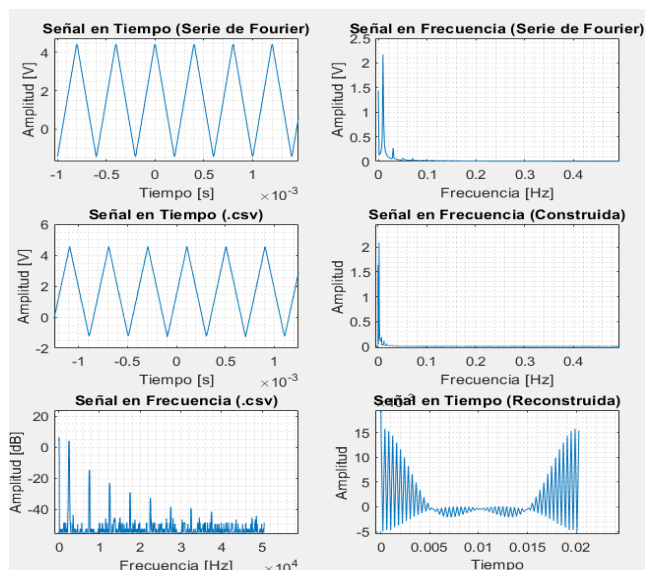


Fig. 30. Gráfica Matlab señal triangular 2.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

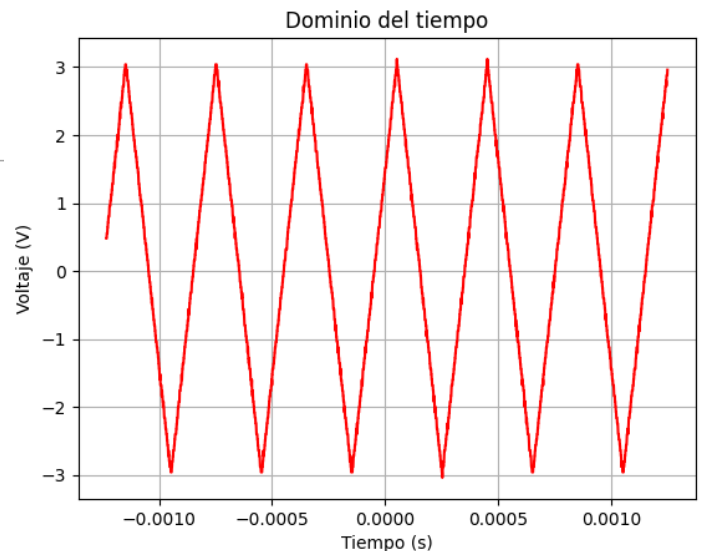


Fig. 31. Gráfica Python dominio del tiempo señal triangular 2.

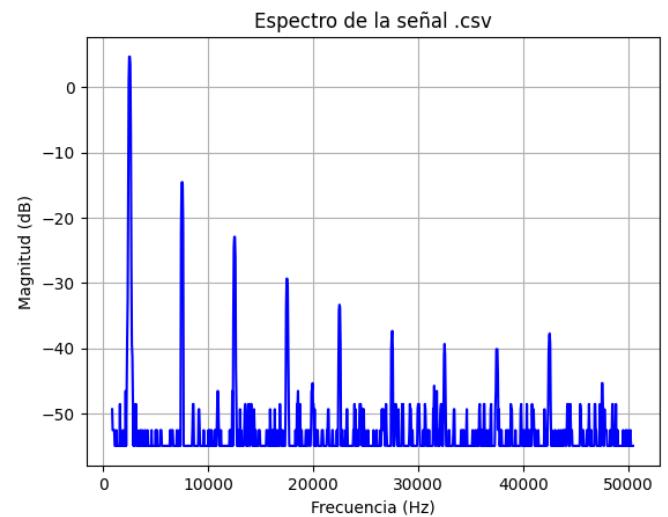


Fig. 32. Gráfica Python dominio de la frecuencia señal triangular 2.



Fig. 33. Gráfica Python FFT señal triangular 2.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y los datos exportados nos permiten compararla de manera teórica.

5. Señal Cuadrada 1.

Frecuencia	2.5 KHz
Vp	3
DC	0

Tabla. 9. Características de la señal.

Fk	2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5
dB	8.7	-0.9	-5.3	-8.1	-10.5	-11.7	-14.1	-14
V	3.85	1.275	0.768	0.557	0.422	0.368	0.279	0.2

Tabla. 10. Características de la señal dominio de la frecuencia experimental.

A partir de los datos experimentales obtenidos realizamos el cálculo teórico mediante el software de Excel:

Componente Exp	3.85	1.275	0.768	0.557	0.422	0.368	0.279	0
Voltaje	3	3	3	3	3	3	3	
n	1	3	5	7	9	11	13	
pi*n	3,1415927	9,424778	15,70796	21,99115	28,27433	34,55752	40,8407	47,1
Componente teori	3,820	1,273	0,764	0,546	0,424	0,347	0,294	0

Fig. 32. Calculo teórico.

Podemos concluir que entre los datos experimentales y teóricos existe un porcentaje de error pequeño y esto se puede deber a la tolerancia de los instrumentos usados.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

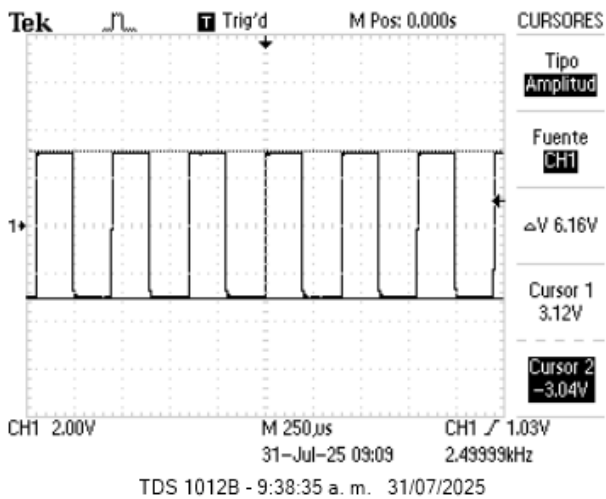


Fig. 34. Grafica en osciloscopio señal cuadrada 1.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

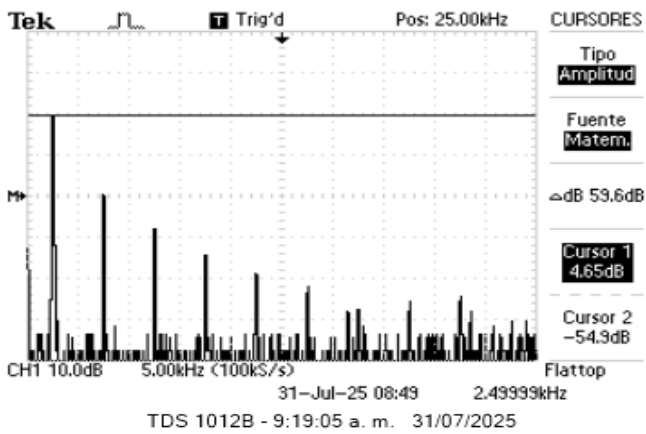


Fig. 35. Grafica en osciloscopio señal cuadrada 1 dominio de la frecuencia.

Luego de obtener los valores teoricos realizamos, un código en matlab para visualizar la señal cuadrada, su espectro en el dominio de la frecuencia, la FFT y la reconstruccion de la señal a partir de los datos que calculamos

Luego implementaremos este código en Python para visualizar la gráfica de la señal:

```
1 % Reconstrucción de la señal con series de fourier
2 f = 2.5e3;
3 T = 1/f;
4 t = -1e-3:T/100:1.5e-3;
5 x1 = 3.82;
6 x3 = 1.273;
7 x5 = 0.764;
8 x7 = 0.546;
9 x9 = 0.424;
10 x11 = 0.347;
11 x13 = 0.294;
12 x15 = 0.255;
13 V1 = x1*sin(2*pi*f*t);
14 V3 = x3*sin(3*2*pi*f*t);
15 V5 = x5*sin(5*2*pi*f*t);
16 V7 = x7*sin(7*2*pi*f*t);
17 V9 = x9*sin(9*2*pi*f*t);
18 V11 = x11*sin(11*2*pi*f*t);
19 V13 = x13*sin(13*2*pi*f*t);
20 V15 = x15*sin(15*2*pi*f*t);
21 vt = V1+V3+V5+V7+V9+V11+V13+V15;
22
23 D = length(vt); % Longitud de la señal
24 VT = fft(vt);
25 VT2 = abs(VT/D); % Espectro de dos lados
26 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
27 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
28 fvt = (0:(D/2))/D; % vector de frecuencia
```

Fig. 36. Código Matlab.

Luego de ejecutar el código obtenemos la siguiente gráfica:

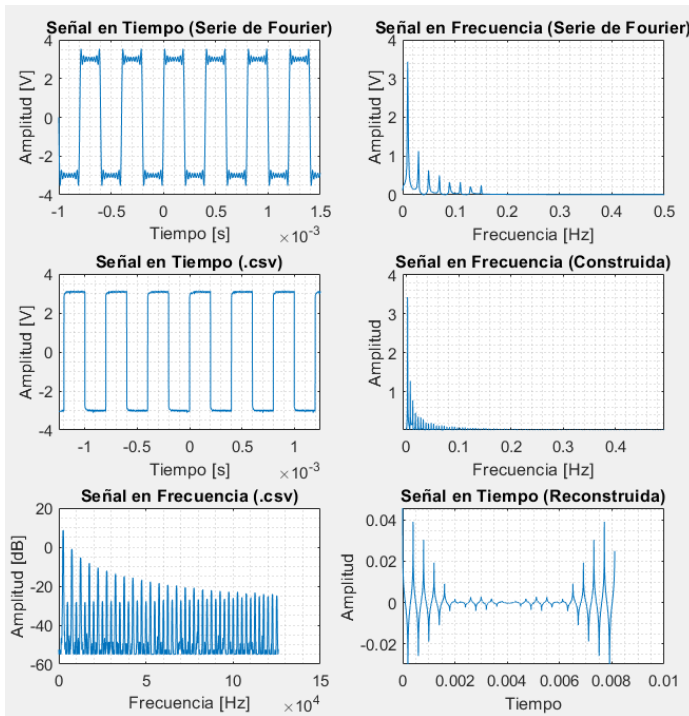


Fig. 37. Gráfica Matlab señal cuadrada 1.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

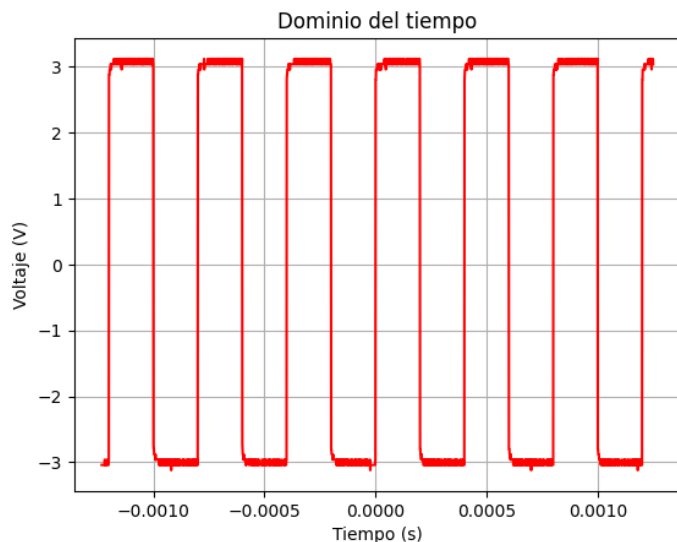


Fig. 38. Gráfica Python dominio del tiempo señal cuadrada 1.

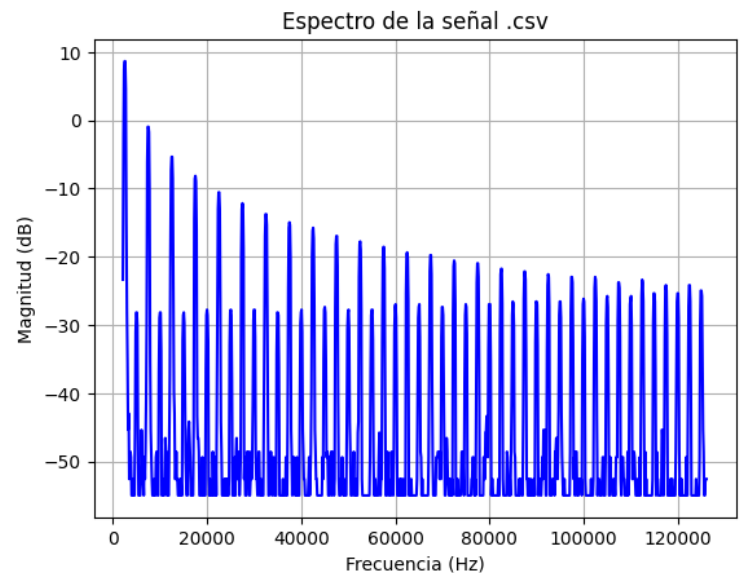


Fig. 39. Gráfica Python dominio de la frecuencia señal cuadrada 1.

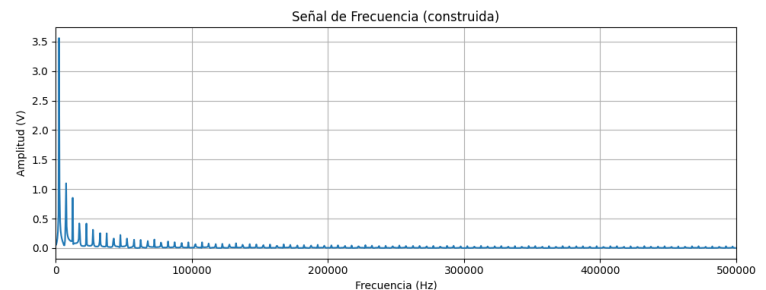


Fig. 40. Gráfica Python FFT señal cuadrada 1.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y los datos exportados nos permiten compararla de manera teórica.

6. Señal Cuadrada 2.

Frecuencia	2.5 KHz
Vp	3
DC	1.25

Tabla. 11. Características de la señal.

Fk	2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5
dB	8.3	-1.3	-5.7	-8.5	-10.5	-12.1	-13.7	-14.9
V	3.677	1.218	0.734	0.532	0.422	0.351	0.292	0.254

Tabla. 12. Características de la señal dominio de la frecuencia experimental.

A partir de los datos experimentales obtenidos realizamos el cálculo teórico mediante el software de Excel:

Componente Exp	3,677	1,218	0,734	0,532	0,422	0,351	0,292	0,254
Voltaje	3	3	3	3	3	3	3	3
n	1	3	5	7	9	11	13	15
pi*n	3,1415927	9,424778	15,70796	21,99115	28,27433	34,55752	40,8407	47,12389
Componente teori	3,820	1,273	0,764	0,546	0,424	0,347	0,294	0,255

Fig. 41. Cálculo teórico.

Podemos concluir que entre los datos experimentales y teóricos existe un porcentaje de error pequeño y esto se puede deber a la tolerancia de los instrumentos usados.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

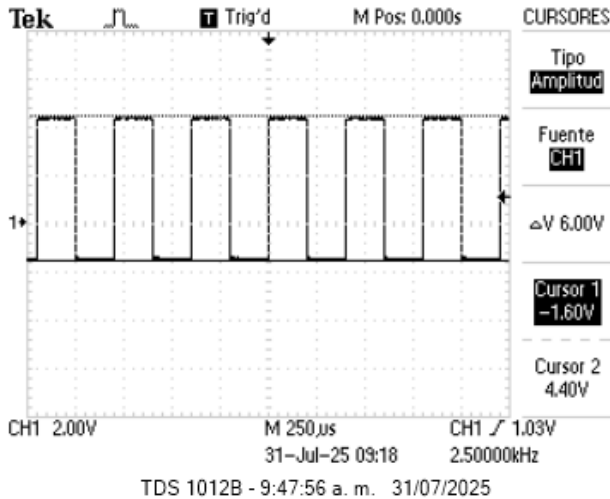


Fig. 42. Grafica en osciloscopio señal cuadrada 2.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

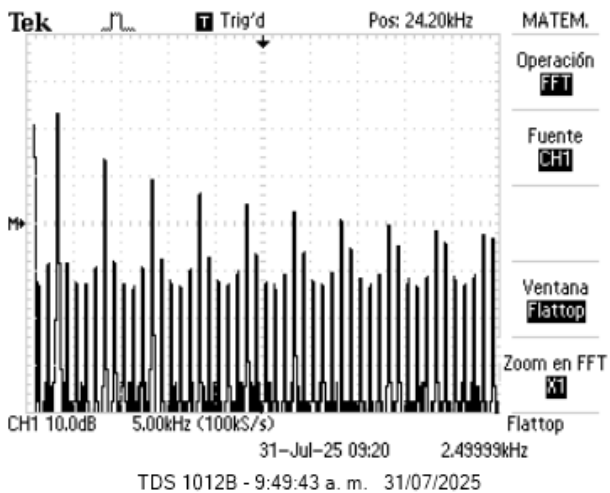


Fig. 43. Grafica en osciloscopio señal cuadrada 2 dominio de la frecuencia.

Luego de obtener los valores teóricos realizamos, un código en matlab para visualizar la señal cuadrada, su espectro en el dominio de la frecuencia, la FFT y la reconstrucción de la señal a partir de los datos que calculamos

Luego implementaremos este código en Python para visualizar la gráfica de la señal:

```
1 % Reconstrucción de la señal con series de fourier
2 f = 2.5e3;
3 T = 1/f;
4 t = -1e-3:T/100:1.5e-3;
5 x1 = 3.82;
6 x3 = 1.273;
7 x5 = 0.764;
8 x7 = 0.546;
9 x9 = 0.424;
10 x11 = 0.347;
11 x13 = 0.294;
12 x15 = 0.255;
13 V1 = x1*sin(2*pi*f*t);
14 V3 = x3*sin(3*2*pi*f*t);
15 V5 = x5*sin(5*2*pi*f*t);
16 V7 = x7*sin(7*2*pi*f*t);
17 V9 = x9*sin(9*2*pi*f*t);
18 V11 = x11*sin(11*2*pi*f*t);
19 V13 = x13*sin(13*2*pi*f*t);
20 V15 = x15*sin(15*2*pi*f*t);
21 DC = 1.25;
22 vt = V1+V3+V5+V7+V9+V11+V13+V15+DC;
23
24 D = length(vt); % Longitud de la señal
25 VT = fft(vt);
26 VT2 = abs(VT/D); % Espectro de dos lados
27 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
28 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
29 fvt = (0:(D/2))/D; % vector de frecuencia
```

Fig. 44. Código Matlab.

Luego de ejecutar el código obtenemos la siguiente gráfica:

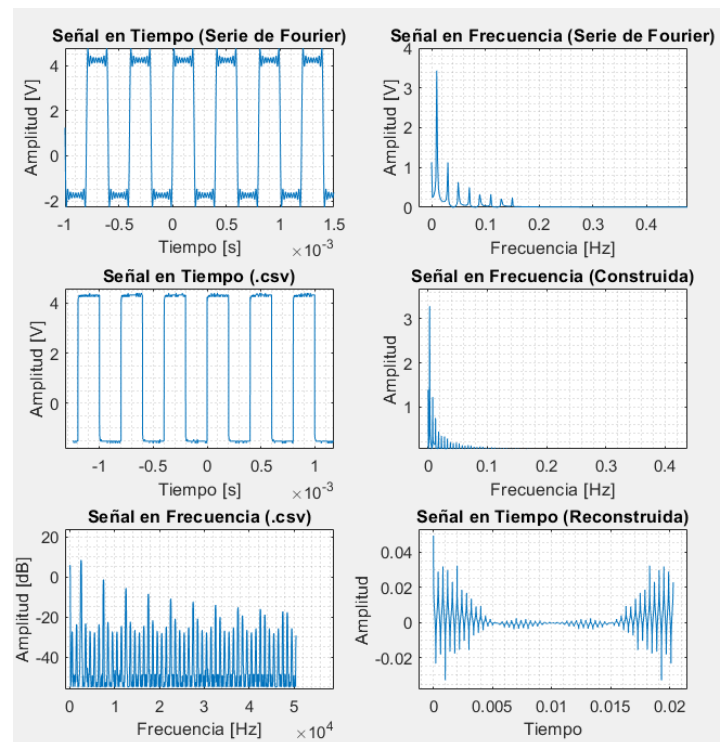


Fig. 45. Grafica Matlab señal cuadrada 1.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

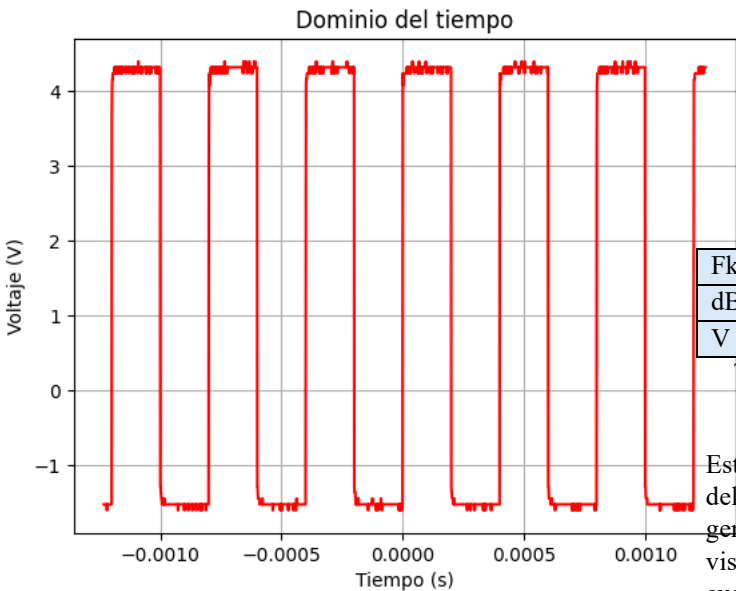


Fig. 46. Grafica Python dominio del tiempo señal cuadrada 2.

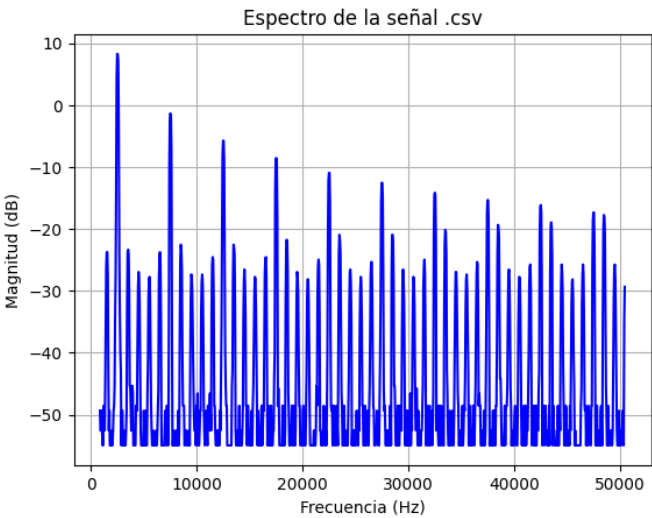


Fig. 47. Grafica Python dominio de la frecuencia señal cuadrada 2.

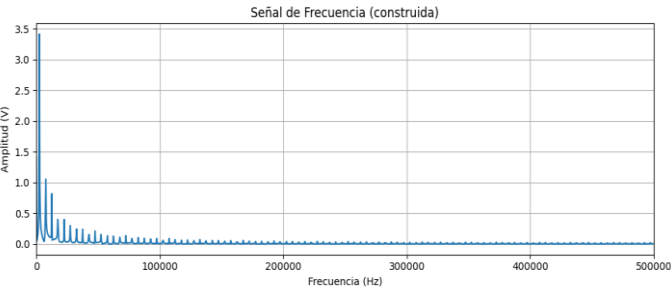


Fig. 48. Grafica Python FFT señal cuadrada 2.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y en ellas para esta señal observamos el DC en las gráficas, los datos exportados nos permiten compararla de manera teórica.

7. Señal pulso 1.

Frecuencia	2.5 KHz
Vp	3
DC	0
Ciclo util	50

Tabla. 13. Características de la señal.

Fk	2.5	7.5	12.5	17.5	22.5	27.5	32.5	37.5
dB	8.7	-0.5	-5.3	-8.1	-10.5	-12.1	-13.3	-14.9
V	3.85	1.33	0.76	0.55	0.422	0.351	0.306	0.266

Tabla. 14. Características de la señal dominio de la frecuencia experimental.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

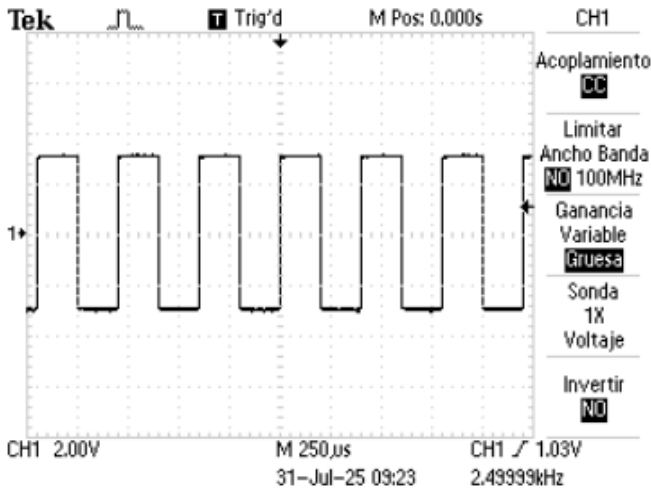


Fig. 49. Grafica en osciloscopio señal pulso 1.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

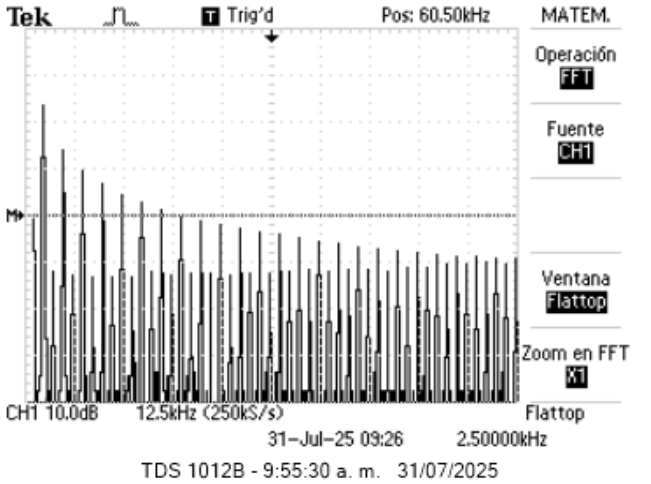


Fig. 50. Grafica en osciloscopio señal pulso 1 dominio de la frecuencia.

Luego de obtener los valores teóricos realizamos, un código en matlab para visualizar la señal pulso, su espectro en el dominio de la frecuencia, la FFT y la reconstrucción de la señal a partir de los datos que calculamos, la grafica obtenida en python es:

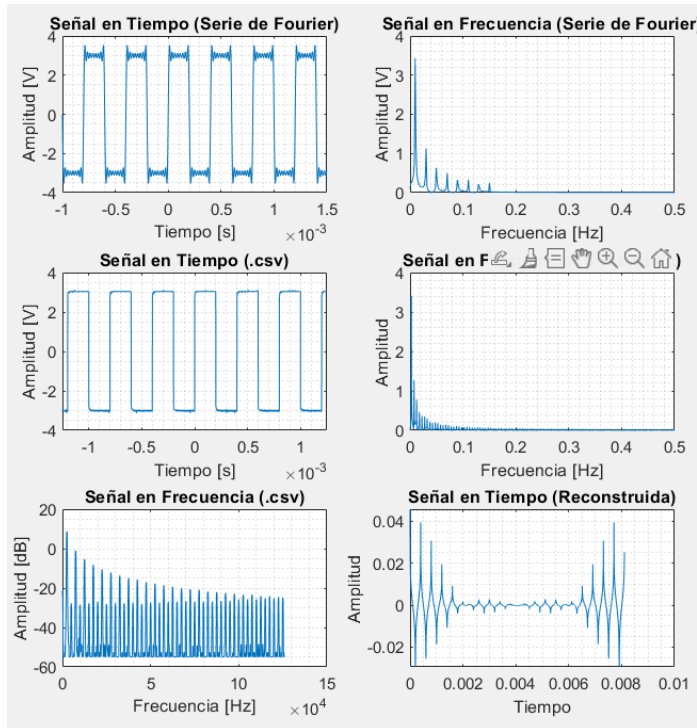


Fig. 51. Grafica Matlab señal pulso 1

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

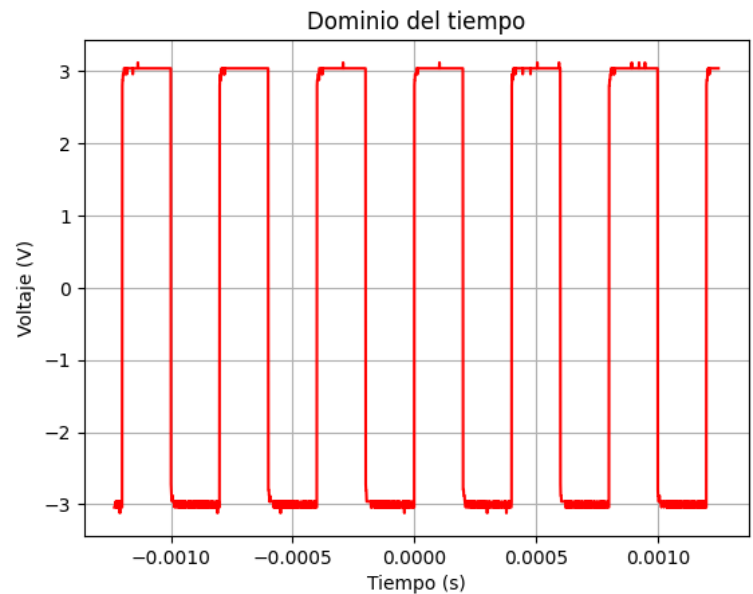


Fig. 52. Grafica Python dominio del tiempo señal pulso 1.

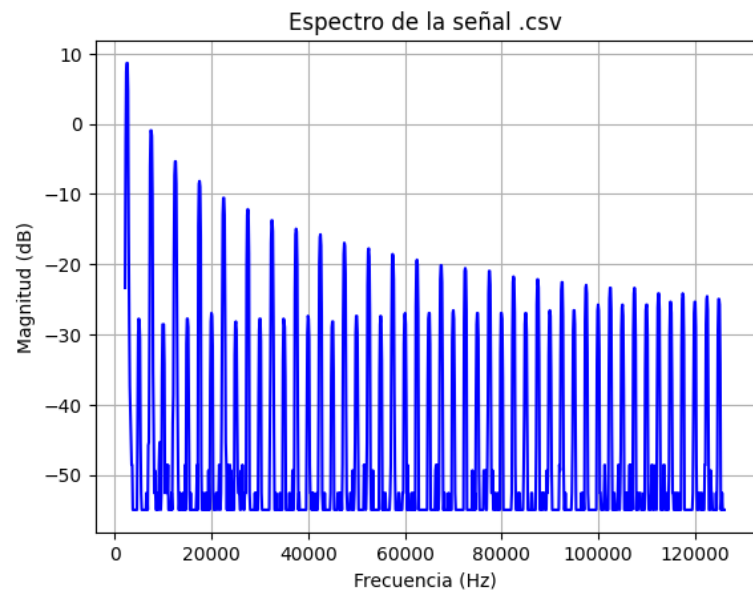


Fig. 53. Grafica Python dominio de la frecuencia señal pulso 1.

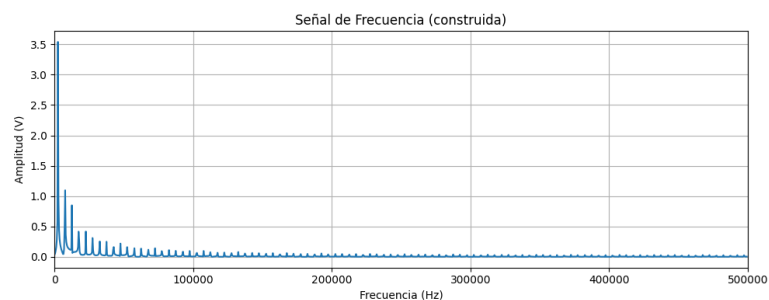


Fig. 54. Grafica Python FFT señal pulso 1.

Podemos concluir que ambos códigos se asemejan a la parte experimental, en este caso el ciclo útil hace ver la señal de manera normal incluso parecida a la cuadrada.

8. Señal pulso 2.

Frecuencia	2.5 KHz
Vp	3
DC	0
Ciclo útil	20

Tabla. 13. Características de la señal.

Fk	2.5	5	7.5	12.5	17.5	22.5	27.5	32.5
dB	4.3	2.7	-1.3	-7.7	0	-11.3	-8.9	-9.3
V	2.32	1.93	1.21	0.58	1.414	0.385	0.508	0.485

Tabla. 14. Características de la señal dominio de la frecuencia experimental.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

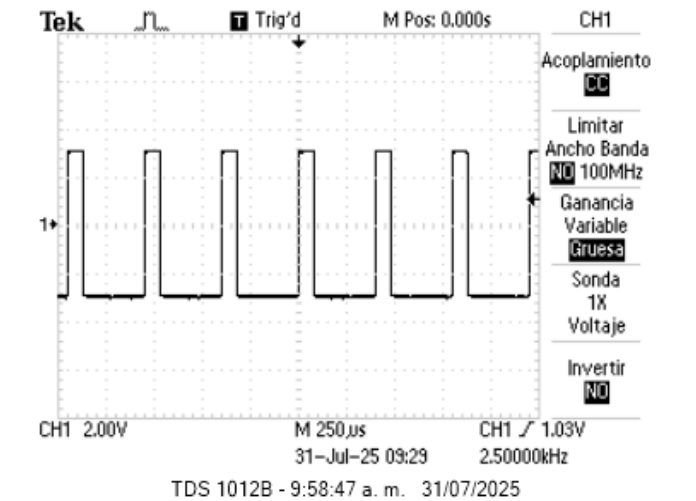


Fig. 55. Grafica en osciloscopio señal pulso 2.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

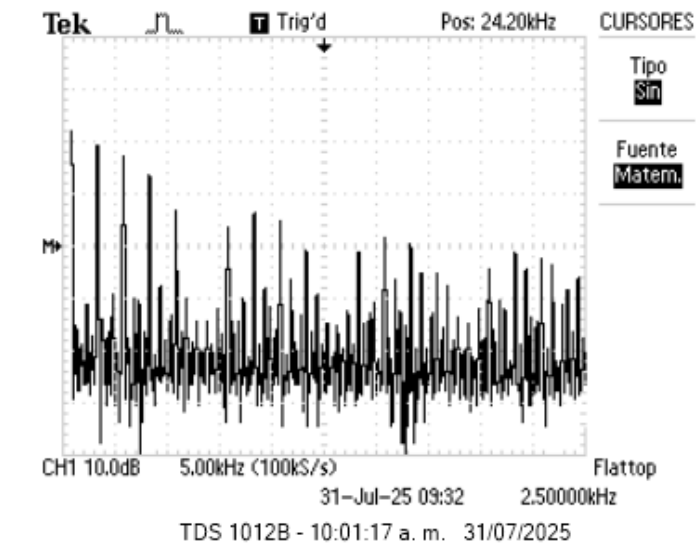


Fig. 56. Grafica en osciloscopio señal pulso 2 dominio de la frecuencia.

Luego de obtener los valores teoricos realizamos, un codigo en matlab para visualizar la señal pulso, su espectro en el dominio de la frecuencia, la FFT y la reconstruccion de la señal a partir de los datos que calculamos.

Empleamos el siguiente codigo en matlab:

```
1 % Reconstrucción de la señal con series de fourier
2 f = 2.5e3;
3 T = 1/f;
4 w = 2*pi*f;
5 CT = 0.2;
6 tau = CT*T;
7 x = pi*(tau/T);
8 V = 3;
9 v = (V*tau)/T;
10 t = -1e-3:T/100:1e-3;
11 vt = v;
12
13 for n = 1:1:10
14     vt = vt + (2*v)*(sin(n*x)/(n*x))*cos(n*w*t);
15 end
16
17 D = length(vt); % Longitud de la señal
18 VT = fft(vt);
19 VT2 = abs(VT/D); % Espectro de dos lados
20 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
21 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
22 fvt = (0:(D/2))/D; % vector de frecuencia
23 % Obtener señal en el tiempo de archivo .csv
24 a0 = pulso2.VarName5;
25 t0 = pulso2.E3;
```

Fig. 57. Código Matlab señal pulso 2.

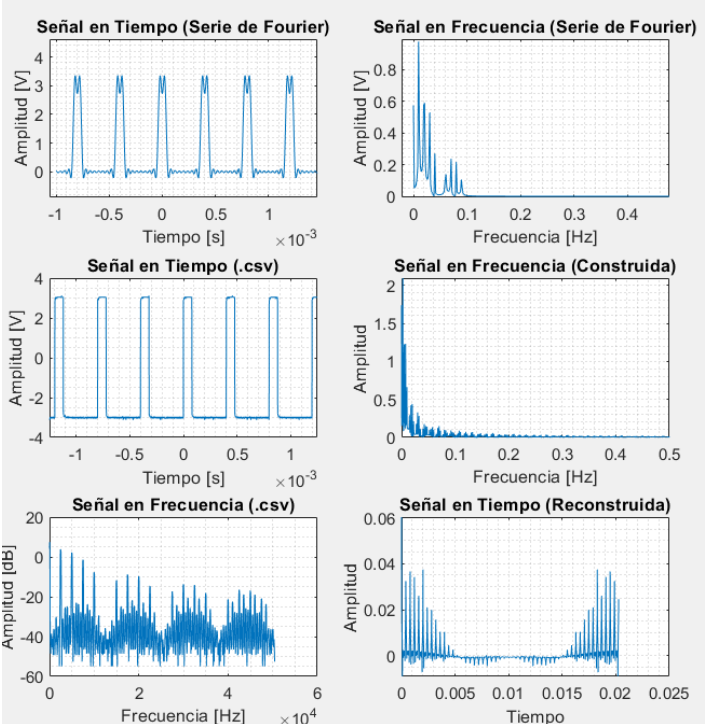


Fig. 57. Grafica Matlab señal pulso 2.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Luego realizamos los cálculos teóricos mediante el software de Excel, en donde también graficaremos la reconstrucción de la señal y podemos visualizar el lóbulo de la señal teórica:

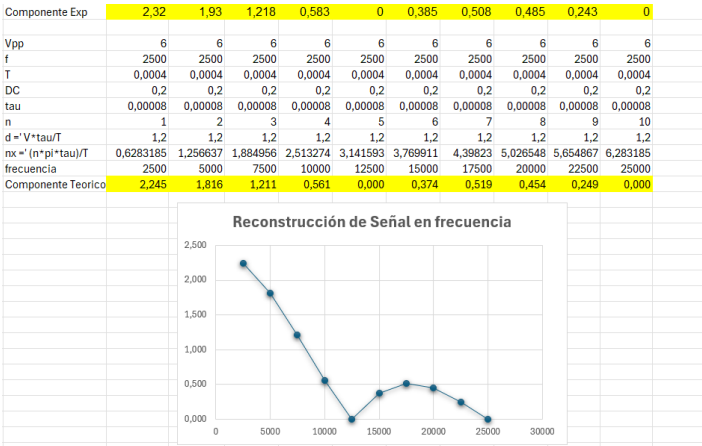


Fig. 58. Calculo teórico y grafico en Excel señal pulso 2.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

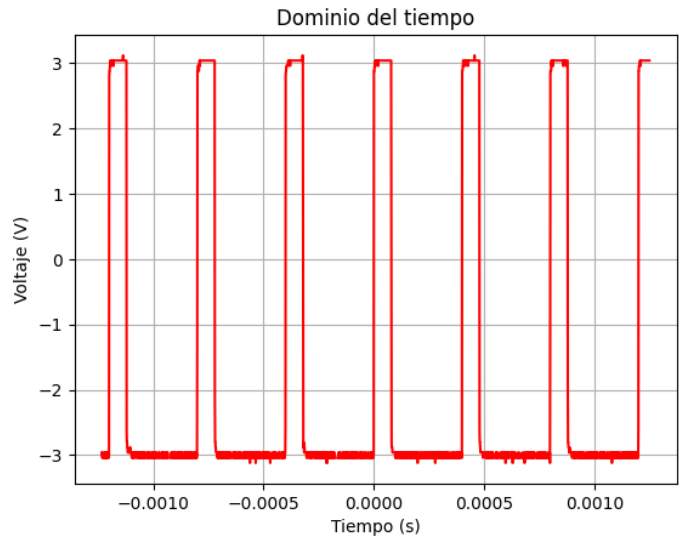


Fig. 59. Grafica Python dominio del tiempo señal pulso 2.

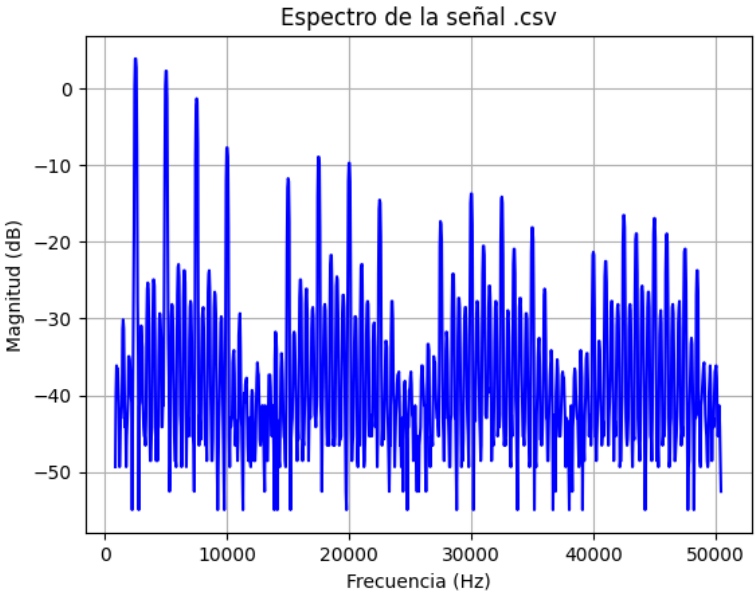


Fig. 60. Grafica Python dominio de la frecuencia señal pulso 2.

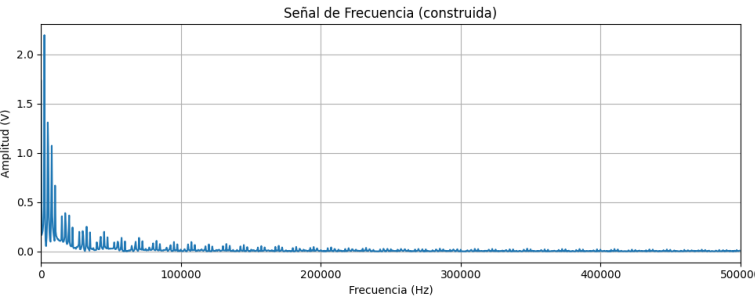


Fig. 61. Grafica Python FFT señal pulso 2.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y en ellas para esta señal observamos el ciclo útil genera un recorte en la señal, hace que los pulsos sean mas angostos, esto se debe a que es el porcentaje de tiempo que la señal está en nivel alto dentro de un periodo completo de la señal.

9. Señal pulso 3.

Frecuencia	2.5 KHz
Vp	3
DC	0
Ciclo util	30

Tabla. 15. Características de la señal.

Fk	2.5	5	7.5	12.5	17.5	22.5	27.5	32.5
dB	7.1	2.3	-10.9	-8.1	-5.3	-11.3	-17.7	-9.7
V	3.20	1.84	0.403	0.55	0.76	0.385	0.184	0.463

Tabla. 16. Características de la señal dominio de la frecuencia experimental.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:

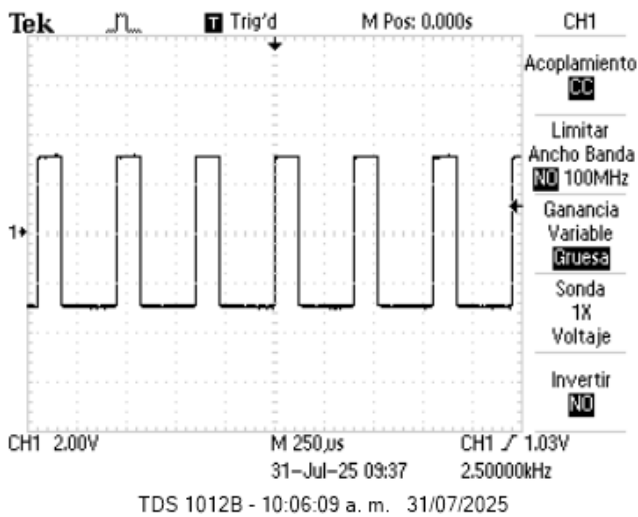


Fig. 62. Grafica en osciloscopio señal pulso 3.

```

1  % Reconstrucción de la señal con series de fourier
2  f = 2.5e3;
3  T = 1/f;
4  w = 2*pi*f;
5  CT = 0.3;
6  tau = CT*T;
7  x = pi*(tau/T);
8  V = 3;
9  v = (V*tau)/T;
10 t = -1e-3:T/100:1e-3;
11 vt = v;
12
13 for n = 1:1:10
14     vt = vt + (2*v)*(sin(n*x)/(n*x))*cos(n*w*t);
15 end
16
17 D = length(vt); % Longitud de la señal
18 VT = fft(vt);
19 VT2 = abs(VT/D); % Espectro de dos lados
20 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
21 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
22 fvt = (0:(D/2))/D; % vector de frecuencia
23 % Obtener señal en el tiempo de archivo .csv
24 a0 = pulso3.VarName5;
25 t0 = pulso3.E3;

```

Fig. 64. Código Matlab señal pulso 3.

Luego pasamos el osciloscopio a la función math para poder visualizar el espectro en función de la frecuencia.

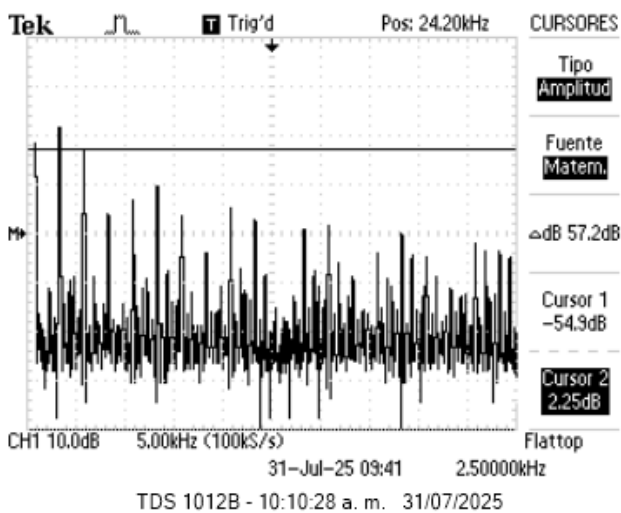


Fig. 63. Grafica en osciloscopio señal pulso 3 dominio de la frecuencia.

Luego de obtener los valores teóricos realizamos, un código en matlab para visualizar la señal pulso, su espectro en el dominio de la frecuencia, la FFT y la reconstrucción de la señal a partir de los datos que calculamos.

Empleamos el siguiente código en matlab:

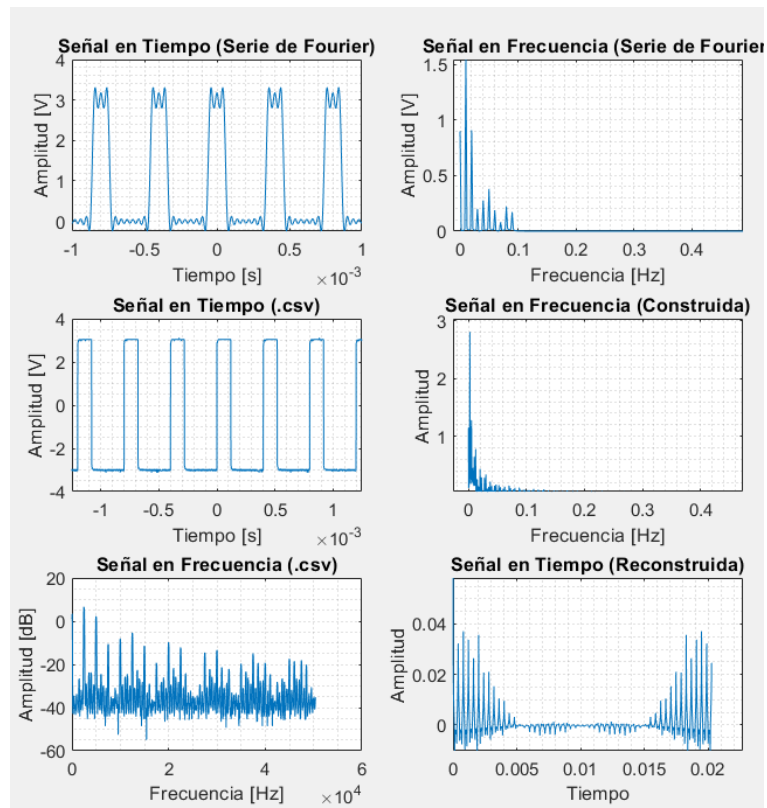


Fig. 65. Grafica Matlab señal pulso 3.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Luego realizamos los cálculos teóricos mediante el software de Excel, en donde también graficaremos la reconstrucción de la señal y podemos visualizar el lóbulo de la señal teórica:

Componente Exp	3,203	1,843	0,403	0,557	0,768	0,385	0,184	0,463	0,351	0
Vpp	6	6	6	6	6	6	6	6	6	6
f	2500	2500	2500	2500	2500	2500	2500	2500	2500	2500
T	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004	0,0004
DC	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3
tau	0,00012	0,00012	0,00012	0,00012	0,00012	0,00012	0,00012	0,00012	0,00012	0,00012
n	1	2	3	4	5	6	7	8	9	10
d = V*tau/T	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8	1,8
rx = (n*pi*tau)/T	0,9424778	1,884956	2,827433	3,769911	4,712389	5,654867	6,597345	7,539822	8,4823	9,424778
frecuencia	2500	5000	7500	10000	12500	15000	17500	20000	22500	25000
Componente Teorico	3,090	1,816	0,393	0,561	0,764	0,374	0,169	0,454	0,343	0,000

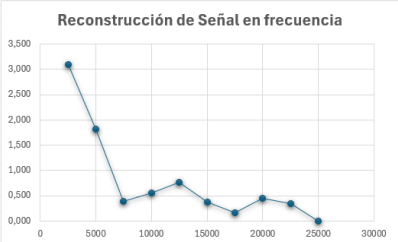


Fig. 66. Calculo teórico y grafico en Excel señal pulso 3.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

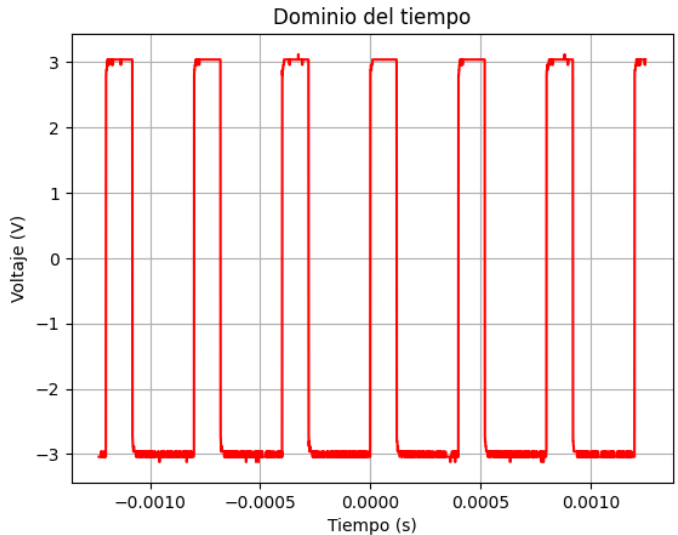


Fig. 67. Grafica Python dominio del tiempo señal pulso 3.

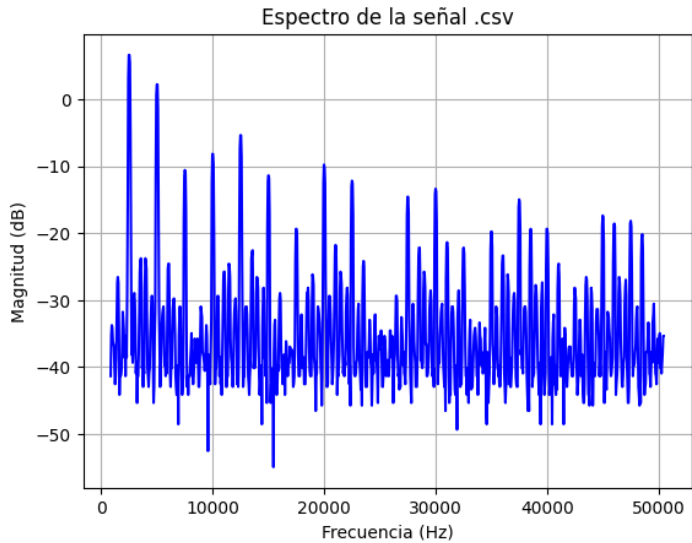


Fig. 68. Grafica Python dominio de la frecuencia señal pulso 3.

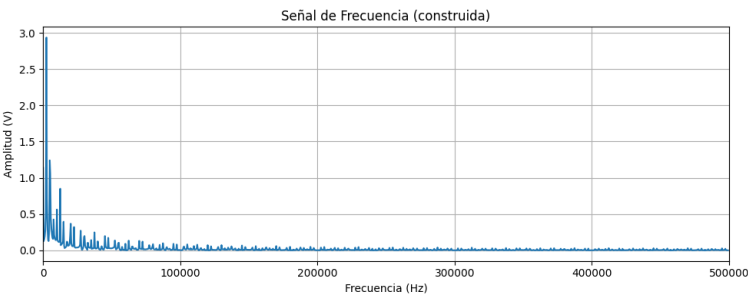


Fig. 69. Grafica Python FFT señal pulso 3.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y en ellas para esta señal observamos el ciclo útil genera un recorte en la señal, hace que los pulsos sean mas angostos, esto se debe a que es el porcentaje de tiempo que la señal está en nivel alto dentro de un periodo completo de la señal.

10. Señal pulso 4.

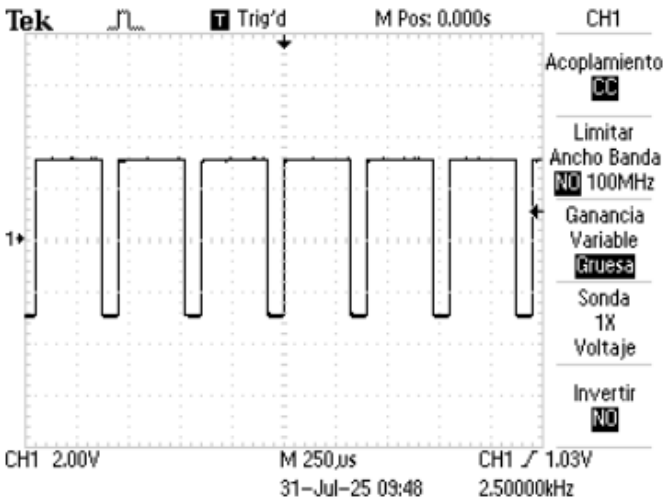
Frecuencia	2.5 KHz
Vp	3
DC	0
Ciclo util	80

Tabla. 17. Características de la señal.

Fk	2.5	5	7.5	12.5	17.5	22.5	27.5	32.5
dB	10.3	8.3	4.7	-1.3	0	-5.3	-2.5	-3.7
V	2.31	1.83	1.214	0.609	0.003	0.384	0.530	0.462

Tabla. 18. Características de la señal dominio de la frecuencia experimental.

Estos son los datos que obtuvimos experimentalmente a partir del espectro en el dominio de la frecuencia, pero para empezar generamos la señal en el generador de señales y la visualizamos en el osciloscopio en el dominio del tiempo lo cual se ve así:



TDS 1012B - 10:17:49 a. m. 31/07/2025

Fig. 70. Grafica en osciloscopio señal pulso 4.

Luego pasamos el osciloscopio a la función math para poder

visualizar el espectro en función de la frecuencia.

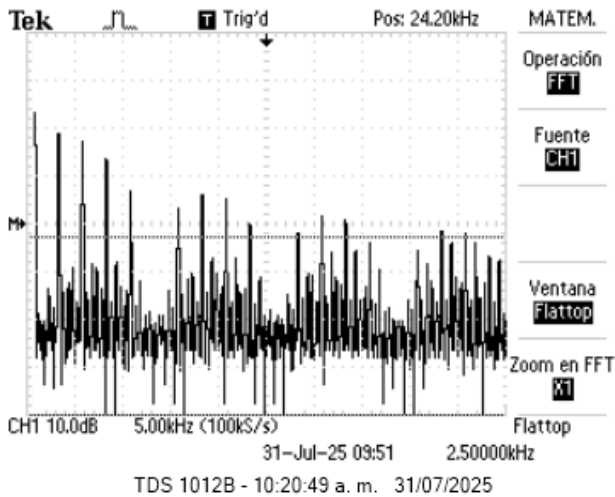


Fig. 71. Gráfica en osciloscopio señal pulso 4 dominio de la frecuencia.

Luego de obtener los valores teóricos realizamos, un código en matlab para visualizar la señal pulso, su espectro en el dominio de la frecuencia, la FFT y la reconstrucción de la señal a partir de los datos que calculamos.

Empleamos el siguiente código en matlab:

```
1 % Reconstrucción de la señal con series de fourier
2 f = 2.5e3;
3 T = 1/f;
4 w = 2*pi*f;
5 CT = 0.8;
6 tau = CT*T;
7 x = pi*(tau/T);
8 V = 3;
9 v = (V*tau)/T;
10 t = -1e-3:T/100:1e-3;
11 vt = v;
12
13 for n = 1:1:10
14     vt = vt + (2*v)*(sin(n*x)/(n*x))*cos(n*w*t);
15 end
16
17 D = length(vt); % Longitud de la señal
18 VT = fft(vt);
19 VT2 = abs(VT/D); % Espectro de dos lados
20 VT1 = VT2(1:fix(D/2)+1); % Espectro de un solo lado
21 VT1(2:end-1) = 2*VT1(2:end-1); % Ajustar para simetria
22 fvt = (0:(D/2))/D; % vector de frecuencia
23 % Obtener señal en el tiempo de archivo .csv
24 a0 = pulso4.VarName5;
25 t0 = pulso4.E3;
```

Fig. 72. Código Matlab señal pulso 4.

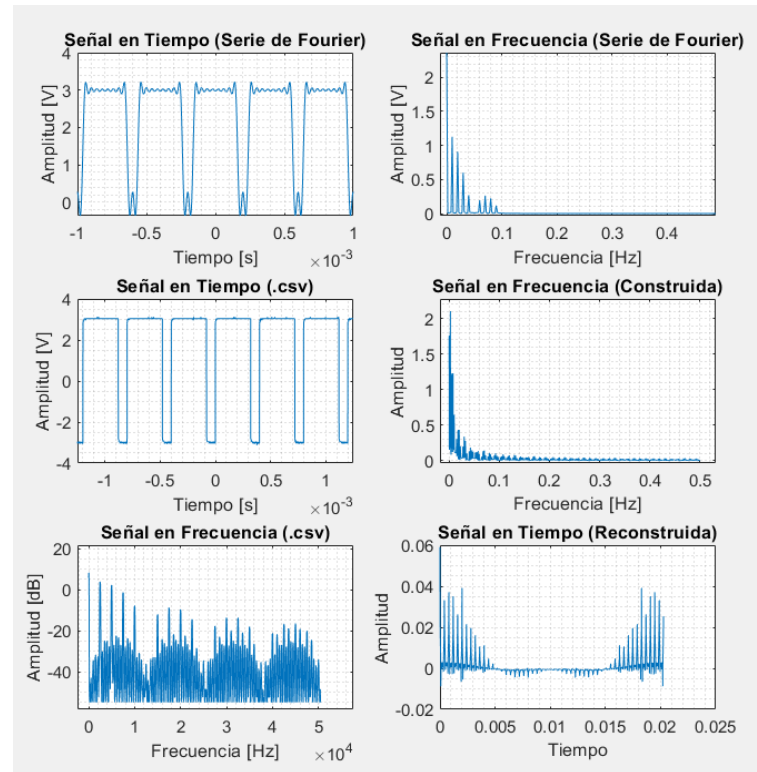


Fig. 73. Gráfica Matlab señal pulso 4.

Las gráficas obtenidas son correctas en comparación con lo teórico y experimental lo único que no se puede percibir de manera correcta es la IFFT, ya que nos faltarían datos para que sea graficada de manera correcta.

Luego realizamos los cálculos teóricos mediante el software de Excel, en donde también graficaremos la reconstrucción de la señal y podemos visualizar el lóbulo de la señal teórica:

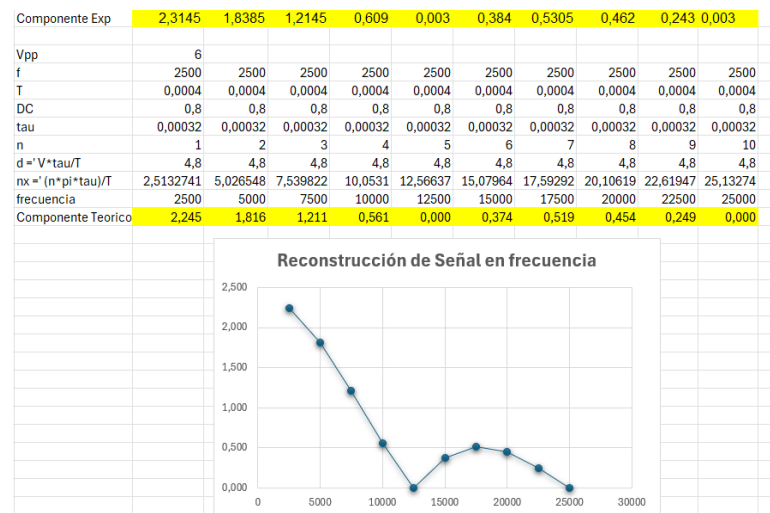


Fig. 74. Calculo teórico y grafico en Excel señal pulso 4.

Continuamos con la graficacion en Python en la cual se usan los códigos usados en la señal seno ya que es estándar solo modificamos el archivo .csv de donde se van a extraer los datos.

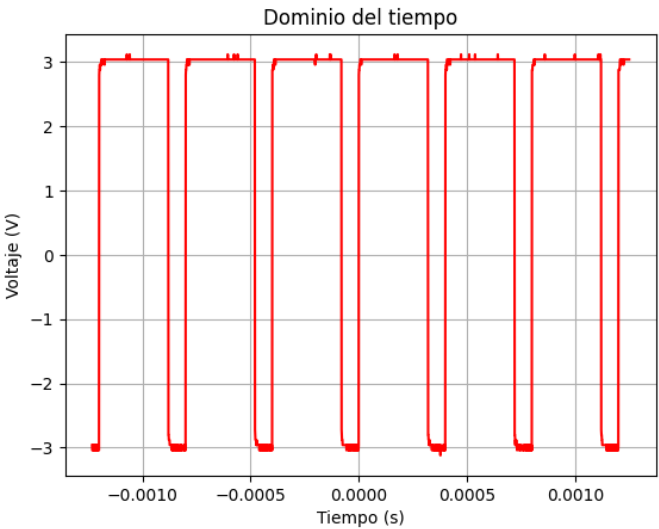


Fig. 75. Grafica Python dominio del tiempo señal pulso 4.

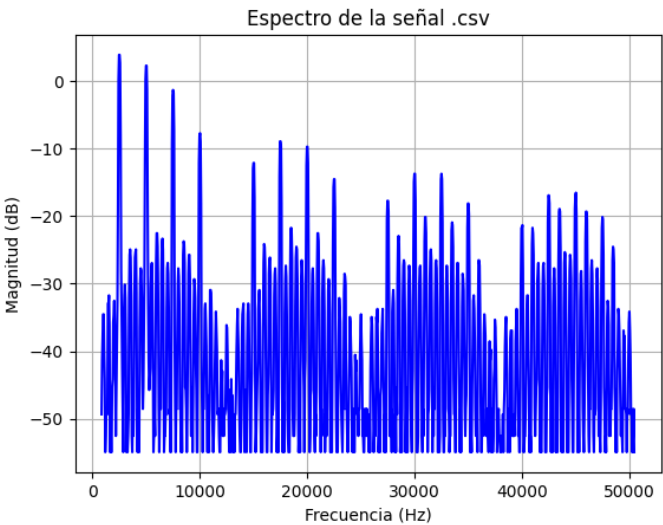


Fig. 76. Grafica Python dominio de la frecuencia señal pulso 4.

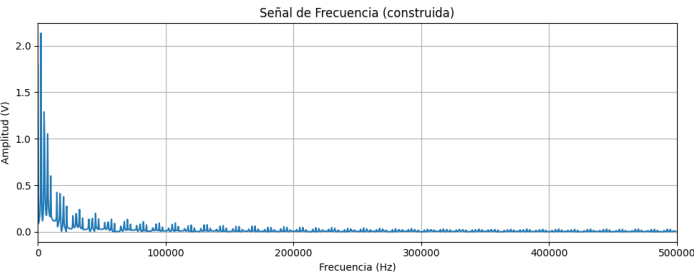


Fig. 77. Grafica Python FFT señal pulso 4.

Podemos concluir que ambos códigos se asemejan a la parte experimental, y en ellas para esta señal observamos el ciclo útil hace que los pulsos sean más anchos, esto se debe a que es el porcentaje de tiempo que la señal está en nivel alto dentro de un periodo completo de la señal.

Para concluir con este informe y practica vamos a responder las siguientes preguntas:

- Active la función MATH/FFT en el osciloscopio digital

y ajuste los controles de tiempo para poder observar los primeros armónicos de la señal. Almacene un archivo CSV de la señal en el dominio de la frecuencia. Ábralo con el block de notas, con Matlab y Python para verificar la estructura del archivo. ¿Además de los datos del muestreo que otra información está incluida en el archivo?

Campo	Significado
Record Length, 2500	La adquisición contiene 2500 puntos de datos.
Sample Interval, 1.000000E-6 s	Cada muestra está separada por 1 μs (1 millón de muestras por segundo).
Trigger Point, 1.250000E3 Samples	El disparo (trigger) ocurrió en la muestra 1250 aproximadamente, lo que significa, en el centro de la captura.
Source, CH1	La señal se capturó en el canal 1.
Vertical Units, Volts	Las unidades verticales son voltios.
Vertical Scale, 2.0000000	Cada división vertical equivale a 2 V.
Vertical Offset, 0	No se aplicó desplazamiento vertical.
Horizontal Units, s	Las unidades horizontales son segundos.
Horizontal Scale, 249.9999937E-6	Cada división horizontal equivale a 250 μs.
Pt Fmt, Y	El formato de puntos está en Y (valores de voltaje).
Yzero, 0.0	Nivel de referencia de 0 V.
Probe Atten, 1.000000	La atenuación de la sonda es 1× (medida directa).
Note, TDS 1012B - 9:03:49 a. m. 31/07/2025	Fecha y hora de la captura.

- Analice los parámetros técnicos del instrumento empleado para la captura de las señales. ¿Cuál es la tasa de muestreo del osciloscopio utilizado? Revise los datos de tiempo del archivo .CSV.

Intervalo de muestreo (Sample Interval)
1.000000×10−6 seg = 1 us

La tasa de muestreo seria:

$$f_s = \frac{1}{T_s} = \frac{1}{1 \times 10^{-6} \text{ s}} = 1,000,000 \text{ muestras/segundo}$$

Link repositorio GitHub:
<https://github.com/belkyvalentina11/Comunicaci-n-digital-.git>

REFERENCIAS

[1] Notes, E. (s. f.). *Electronic Waveforms & Signals: sine, square, triangle . . . » Electronics Notes*.

https://www.electronics-notes.com/articles/basic_concepts/electronic-electrical-waveforms/waveform-types-basics-sine-square-triangle-ramp.php

[2] 2. *Dominio de la Frecuencia | PySDR: A Guide to SDR and DSP using Python*. (s. f.).

https://pysdr.org/es/content-es/frequency_domain.html