

LABORATORIO DE PROCESAMIENTO DE SEÑALES

Vallentina Diaz Valbuena

est.vallentina.diaz@unimilitar.edu.co

Docente: José de Jesús Rugeles

I. RESUMEN

Este informe presenta un análisis detallado del comportamiento espectral de diferentes tipos de señales periódicas (senoidales, triangulares, cuadradas y de pulso) mediante el estudio de sus armónicos en los dominios del tiempo y la frecuencia. El trabajo combina análisis teórico con mediciones experimentales realizadas en un osciloscopio digital Tektronix TDS 2012B, complementadas con procesamiento computacional utilizando MATLAB.

Se generaron y analizaron múltiples señales con diferentes parámetros: voltaje pico de 1V, distintos valores de componente DC (0, 1, 1.25 y 1.5V), y para señales de pulso, diferentes ciclos útiles (20%, 30%, 50% y 80%). El estudio incluyó la reconstrucción de señales a partir de sus armónicos teóricos y experimentales, la comparación de espectros obtenidos mediante FFT e IFFT de datos temporales con mediciones espectrales directas del osciloscopio, y los intentos de reconstrucción inversa del dominio temporal a partir de datos espectrales.

Los resultados muestran una excelente concordancia entre valores teóricos y experimentales, especialmente para armónicos de mayor orden, con porcentajes de error que disminuyen conforme aumenta el número del armónico.

II. ABSTRACT

This report presents a detailed analysis of the spectral behavior of different types of periodic signals (sinusoidal, triangular, square, and pulse) by studying their harmonics in the time and frequency domains. The work combines theoretical analysis with experimental measurements performed on a Tektronix TDS 2012B digital oscilloscope, complemented by computational processing using MATLAB.

Multiple signals were generated and analyzed with different parameters: a peak voltage of 1 V, various DC component values (0, 1, 1.25, and 1.5 V), and for pulse signals, different duty cycles (20%, 30%, 50%, and 80%). The study included signal reconstruction from their theoretical and experimental harmonics, comparison of spectra obtained using FFT and IFFT of time-based data with direct oscilloscope spectral

measurements, and attempts at inverse time-domain reconstruction from spectral data. The results show excellent agreement between theoretical and experimental values, especially for higher-order harmonics, with error percentages decreasing as the harmonic number increases.

III. DESARROLLO EXPERIMENTAL

El desarrollo experimental de este laboratorio se centró en la adquisición, análisis y reconstrucción de diferentes tipos de señales periódicas utilizando un osciloscopio digital y el software MatLab para el procesamiento de datos. El procedimiento experimental se estructuró en varias fases clave:

Inicialmente, se configuró el osciloscopio para capturar las señales en el dominio del tiempo, cuando se activa la función MATH/FFT en el osciloscopio digital y se ajustan los controles de tiempo para poder observar los primeros armónicos de la señal. Al almacenar el archivo CSV de la señal en el dominio de la frecuencia, contiene dos tipos de información:

- *Metadatos de configuración de la medición:*

Record Length: 1024 puntos (longitud total de la grabación).

Sample Interval: 1.2330334 Hz (frecuencia de muestreo, equivalente a un intervalo de tiempo de ~0.811 segundos entre puntos).

Trigger Point: 0.0 muestras (punto donde se disparó la adquisición).

Source: MATH (la señal proviene de un cálculo matemático interno del instrumento).

Vertical Units: dB (unidades de la amplitud en decibelios).

Vertical Scale: 10.0000001 dB por división.

Vertical Offset: -26.9897 dB (valor de referencia o desplazamiento en el eje vertical).

Horizontal Units: Hz (unidades del eje horizontal, frecuencia).

Horizontal Scale: 126.2626221 Hz por división.

Pt Fmt: Y (formato del punto, indica que los datos son valores Y).

Yzero: 0.0 (valor de referencia para el cálculo del formato Y).

Note: TDS 2012B - 9:27:54 a.Ám. 30/07/2025 (Modelo del osciloscopio - Tektronix TDS 2012B - y marca de tiempo de la

captura: 9:27:54 AM del 30 de Julio de 2025. El "a.Ám." parece un artefacto de codificación, probablemente "a.m.")

- *Datos de la señal muestreada:*

La parte principal del archivo (después de los metadatos y las primeras líneas de datos dispersos) consiste en 1024 líneas (o pares de columnas), cada una representando un punto de la señal.

Cada línea contiene dos valores separados por coma:

Primer valor: Frecuencia (en Hz, eje horizontal).

Segundo valor: Amplitud (en dB, eje vertical).

En resumen, la información adicional a los datos de muestreo (frecuencia y amplitud) es toda la metadata que describe cómo se configuró y realizó la medición en el osciloscopio Tektronix TDS 2012B.

Ahora bien, la **tasa de muestreo** se calcula como el inverso del intervalo de muestreo el cual se encuentra en la línea de Sample Interval: 2.0000000E-6 s, por medio de la siguiente ecuación:

$$\text{Tasa de muestreo} = \frac{1}{2 \times 10^{-6} \text{s}} = 500\,000 \text{ muestras/s}$$

Señales reconstruidas (armónicos teóricos)

- Senoidal ($V_p=1$, $DC=0$)

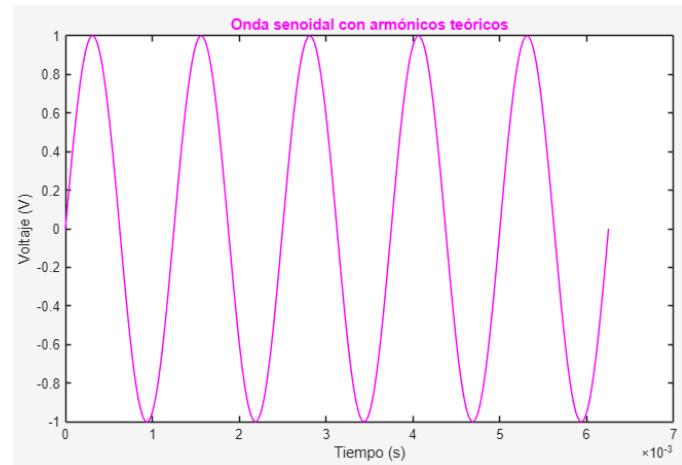
Para generar la reconstrucción de una señal senoidal se debe considerar que esta solo contiene el armónico fundamental, es decir el primer armónico, ya que en la serie de Fourier los armónicos superiores ($n=3,5,7,\dots$) tienen amplitud cero. Esto contrasta con ondas cuadradas o triangulares que requieren múltiples armónicos.

Por lo tanto, se implementa el siguiente código en MatLab para lograr visualizar la gráfica que se genera de la onda senoidal a partir de su armónico teórico con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
T=0.00125;
f=1/T;
A1=1;
t=0:T/100:5*T;
V1= A1*sin(2*pi*f*t);
plot(t,V1,Color='m');
```

```
xlabel("Tiempo (s)")
ylabel("Voltaje (V)")
title("Onda senoidal con armónicos teóricos", 'Color', 'm')
```

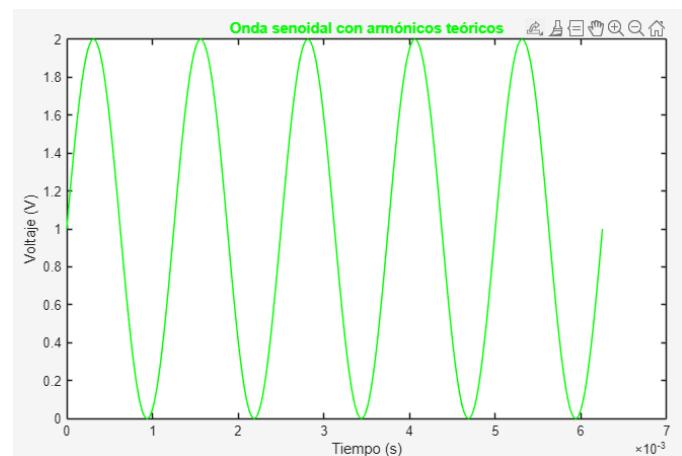
A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica senoidal.



Gráfica 3. Senoidal en MatLab

- Senoidal ($V_p=1$, $DC=1$)

Ahora bien, se implementa un código similar al anterior en MatLab para lograr visualizar la gráfica que se genera de la onda senoidal a partir de su armónico teórico con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título, pero cuando se le agrega un offset de 1V, se añade una variable con este valor y se suma en el voltaje 1. A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica triangular.



Gráfica 3. Senoidal en MatLab



- Triangular ($V_p=1$, $DC=0$)

Para generar las magnitudes de los armónicos de una señal triangular se emplea la siguiente ecuación:

$$A_n = \frac{8 \times V_p}{(n \times \pi)^2}$$

Donde:

V_p =Voltaje pico de la señal

n =Número de armónico

Para una señal triangular se usan únicamente armónicos impares, como se observa en la siguiente tabla:

Armónico	Frecuencia (Hz)	Magnitud (V)
1	800	0,81
3	2400	0,09
5	4000	0,03
7	5600	0,016
9	7200	0,01
11	8800	0,006

Se implementa el siguiente código en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda triangular a partir de sus armónicos teóricos con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
T=0.00125;
f=1/T;
A1=0.81;
A3=0.09;
A5=0.03;
A7=0.016;
A9=0.01;
A11=0.006;
t=0:T:5*T;
V1= A1*cos(2*pi*f*t);
V3= A3*cos(2*pi*3*f*t);
```

$V5= A5\cos(2\pi\cdot5\cdot f\cdot t);$

$V7= A7\cos(2\pi\cdot7\cdot f\cdot t);$

$V9= A9\cos(2\pi\cdot9\cdot f\cdot t);$

$V11= A11\cos(2\pi\cdot11\cdot f\cdot t);$

$VT=V1+V3+V5+V7+V9+V11;$

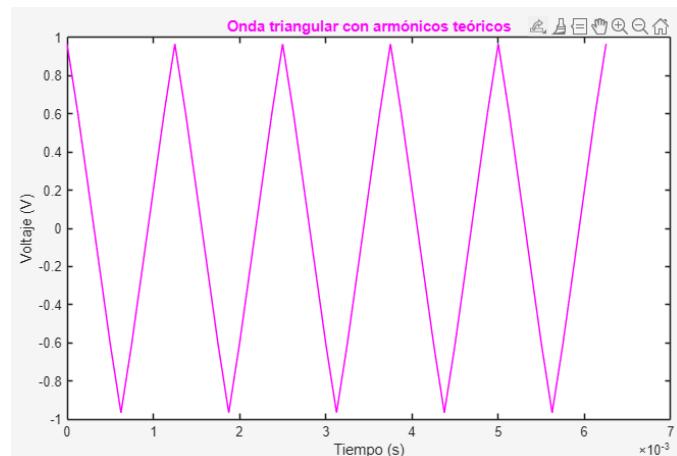
`plot(t, VT, Color='m');`

`xlabel("Tiempo (s)")`

`ylabel("Voltaje (V)")`

`title("Onda triangular con armónicos teóricos", 'Color', 'm')`

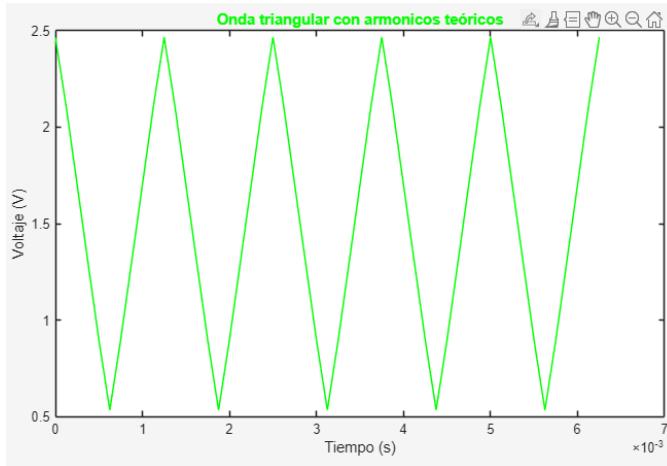
A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica triangular.



Gráfica 3. Triangular en MatLab

- Triangular ($V_p=1$, $DC=1.5$)

Ahora bien, se implementa un código similar al anterior en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda triangular a partir de sus armónicos teóricos con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título, pero cuando se le agrega un offset de 1.5V, se añade una variable con este valor y se suma en el voltaje total. A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica triangular.



Gráfica 3. Triangular en MatLab

- Cuadrada ($V_p=1$, $DC=0$)

Para generar las magnitudes de los armónicos de una señal cuadrada se emplea la siguiente ecuación:

$$A_n = \frac{4 \times V_p}{n \times \pi}$$

Donde:

V_p =Voltaje pico de la señal

n =Número de armónico

Para una señal triangular se usan únicamente armónicos impares, como se observa en la siguiente tabla:

Armónico	Frecuencia (Hz)	Magnitud (V)
1	800	1,27
3	2400	0,42
5	4000	0,25
7	5600	0,18
9	7200	0,14
11	8800	0,11

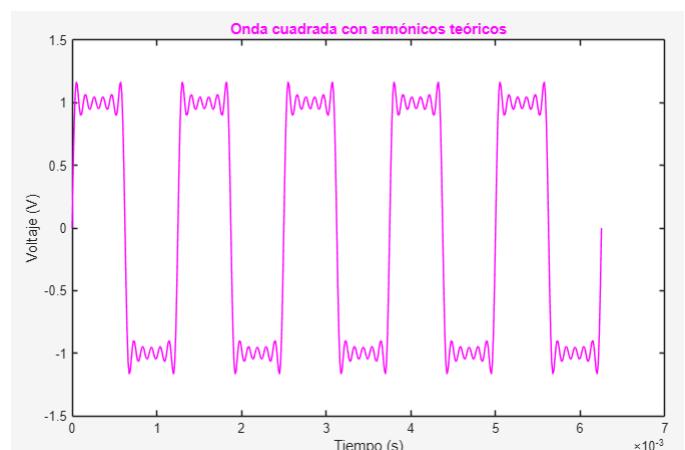
Se implementa el siguiente código en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda cuadrada a partir de sus armónicos teóricos con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

$T=0.00125;$

$f=1/T;$

```
A1=1.27;
A3=0.42;
A5=0.25;
A7=0.18;
A9=0.14;
A11=0.11;
t=0:T/100:5*T;
V1= A1*sin(2*pi*f*t);
V3= A3*sin(2*pi*3*f*t);
V5= A5*sin(2*pi*5*f*t);
V7= A7*sin(2*pi*7*f*t);
V9= A9*sin(2*pi*9*f*t);
V11= A11*sin(2*pi*11*f*t);
VT=V1+V3+V5+V7+V9+V11;
plot(t,VT,Color='m');
xlabel("Tiempo (s)")
ylabel("Voltaje (V)")
title("Onda cuadrada con armónicos teóricos", 'Color', 'm')
```

A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica cuadrada.

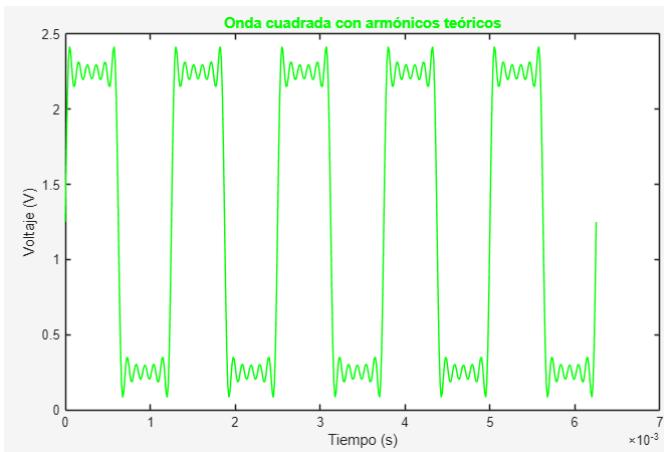


Gráfica 3. Cuadrada en MatLab



- Cuadrada ($V_p=1$, $DC=1.25$)

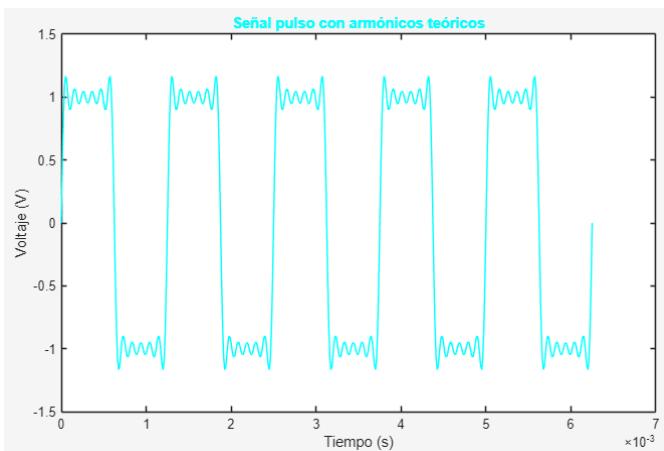
Ahora bien, se implementa un código similar al anterior en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda cuadrada a partir de sus armónicos teóricos con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título, pero cuando se le agrega un offset de 1.25V, se añade una variable con este valor y se suma en el voltaje total. A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal periódica triangular.



Gráfica 3. Cuadrada en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 50%)

Para generar las magnitudes de los armónicos de una señal pulso rectangular con un ciclo útil del 50%, es similar a los armónicos de la onda cuadrada, hallados anteriormente, por lo tanto a continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal pulso.



Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 20%)

Para generar las magnitudes de los armónicos de una señal pulso se emplea la siguiente ecuación:

$$A_n = 2 \times V_p \left(\frac{\sin(n\pi \times \tau)}{n\pi} \right)$$

Donde:

V_p =Voltaje pico de la señal

n =Número de armónico

τ =Ancho de pulso

Para una señal pulso se usan armónicos pares e impares, como se observa en la siguiente tabla:

Armónico	Frecuencia (Hz)	Magnitud (V)
1	800	0,37
2	1600	0,3
3	2400	0,2
4	3200	0,093
5	4000	0
6	4800	-0,062
7	5600	-0,086
8	6400	-0,075
9	7200	-0,041

Se implementa el siguiente código en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal pulso a partir de sus armónicos teóricos con un voltaje pico de 1V, un ciclo útil del 20% y una frecuencia de 800Hz, con sus respectivos ejes y título.

$T = 0.00125;$

$f = 1/T;$

$A1 = 0.37;$

$A2 = 0.3;$

$A3 = 0.2;$

$A4 = 0.093;$

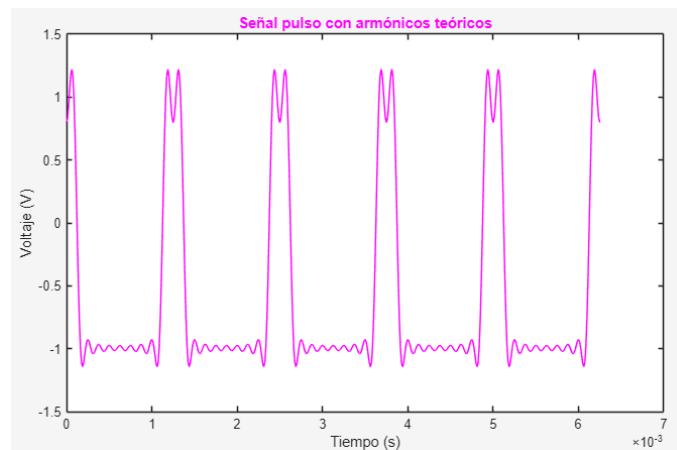
$A5 = 0;$

```

A6 = -0.062;
A7 = -0.086;
A8 = -0.075;
A9 = -0.041;
t = 0:T/100:5*T;
DC_bipolar = 2*(0.2) - 1;
V1 = 2 * A1 * cos(2*pi*f*t);
V2 = 2 * A2 * cos(2*pi*2*f*t);
V3 = 2 * A3 * cos(2*pi*3*f*t);
V4 = 2 * A4 * cos(2*pi*4*f*t);
V5 = 2 * A5 * cos(2*pi*5*f*t);
V6 = 2 * A6 * cos(2*pi*6*f*t);
V7 = 2 * A7 * cos(2*pi*7*f*t);
V8 = 2 * A8 * cos(2*pi*8*f*t);
V9 = 2 * A9 * cos(2*pi*9*f*t);
VT = DC_bipolar + V1 + V2 + V3 + V4
+ V5 + V6 + V7 + V8 + V9;
plot(t, VT, 'Color', 'm');
xlabel("Tiempo (s)");
ylabel("Voltaje (V)");
title("Señal pulso con armónicos teóricos", 'Color', 'm');

```

A continuación, en la Gráfica 3, se logra visualizar la reconstrucción de una señal pulso.



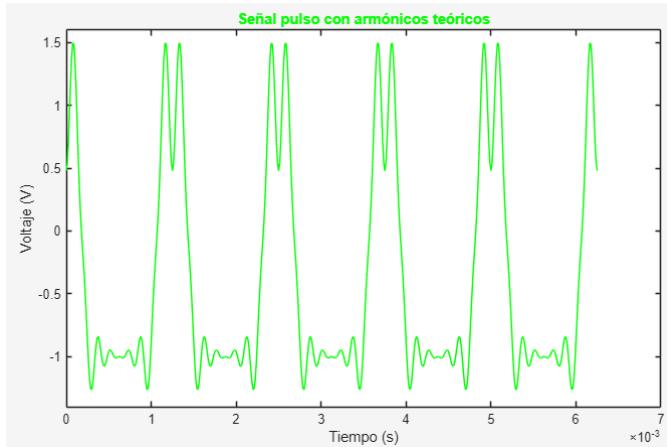
Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 30%)

Se emplea la misma ecuación para el pulso anterior y se tiene en cuenta que para una señal pulso se usan armónicos pares e impares, como se observa en la siguiente tabla:

Armónico	Frecuencia (Hz)	Magnitud (V)
1	800	0,51
2	1600	0,3
3	2400	0,065
4	3200	-0,093
5	4000	-0,12
6	4800	-0,062
7	5600	-0,028
8	6400	-0,075
9	7200	-0,057

Se implementa el mismo código en MatLab del pulso anterior, pero se le asignan las nuevas magnitudes para lograr visualizar la gráfica que se genera de la señal pulso a partir de sus armónicos teóricos con un voltaje pico de 1V, un ciclo útil del 30% y una frecuencia de 800Hz, con sus respectivos ejes y título.



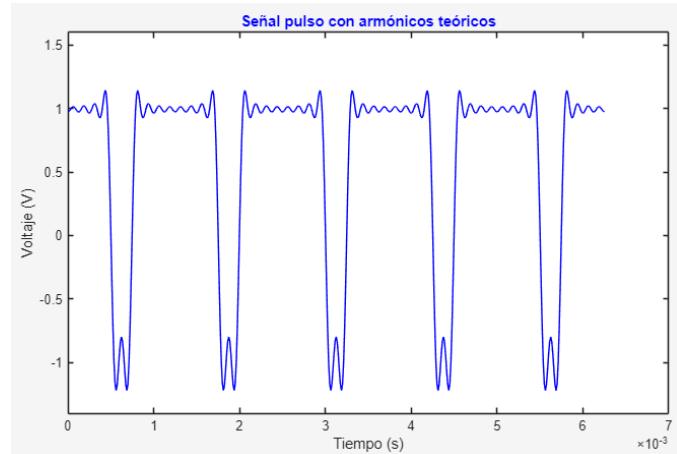
Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 80%)

Se emplea la misma ecuación para el pulso anterior y se tiene en cuenta que para una señal pulso se usan armónicos pares e impares, como se observa en la siguiente tabla:

Armónico	Frecuencia (Hz)	Magnitud (V)
1	800	0,37
2	1600	-0,3
3	2400	0,2
4	3200	-0,093
5	4000	0
6	4800	0,062
7	5600	-0,086
8	6400	0,075
9	7200	-0,041

Se implementa el mismo código en MatLab del pulso anterior, pero se le asignan las nuevas magnitudes para lograr visualizar la gráfica que se genera de la señal pulso a partir de sus armónicos teóricos con un voltaje pico de 1V, un ciclo útil del 80% y una frecuencia de 800Hz, con sus respectivos ejes y título.



Gráfica 3. Pulso en MatLab

Señales reconstruidas (armónicos experimentales)

Por medio de la activación de la función MATH/FFT en el osciloscopio digital, se puede generar la reconstrucción de las señales a partir de los armónicos experimentales, para esto se implementó el uso de los cursores del osciloscopio, para lograr recopilar todos los armónicos relevantes con sus correspondientes frecuencias.

Para la señal senoidal se debe considerar que esta solo contiene el armónico fundamental, es decir el primer armónico, ya que en la serie de Fourier los armónicos superiores ($n=3,5,7,\dots$) tienen amplitud cero. Esto contrasta con ondas cuadradas o triangulares que requieren múltiples armónicos.

- Senoidal ($V_p=1$, $DC=0$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-2,99

- Senoidal ($V_p=1$, $DC=1$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-2,83

- Triangular ($V_p=1$, $DC=0$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-4,59
3	2400	-24,6



5	4000	-32,2
7	5600	-38,6
9	7200	-42,6
11	8800	-46,2

- Triangular ($V_p=1$, $DC=1.5$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-4,52
3	2400	-23,4
5	4000	-32,6
7	5600	-38,2
9	7200	-41,8
11	8800	-45,8

- Cuadrada ($V_p=1$, $DC=0$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-0,9
3	2400	-11
5	4000	-15,8
7	5600	-17,8
9	7200	-20,2
11	8800	-22,2

- Cuadrada ($V_p=1$, $DC=1.25$)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-0,88
3	2400	-10,6
5	4000	-15,4
7	5600	-17,4
9	7200	-19,8
11	8800	-21,8

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 50%)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-0,98
3	2400	-10,2
5	4000	-15
7	5600	-17,8
9	7200	-19,8
11	8800	-21,8

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 20%)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-5.39
2	1600	-7.39
3	2400	-10.6
4	3200	-17.4
5	4000	ruido
6	4800	-21
7	5600	-18.2
8	6400	-19.4
9	7200	-24.2

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 30%)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-2.59
2	1600	-7.39
3	2400	-20.2
4	3200	17.8
5	4000	-15
6	4800	-21
7	5600	-27.8



8	6400	-19.4
9	7200	-21.4

- Pulso (Vp=1, DC=0, Ciclo útil= 80%)

Armónico	Frecuencia (Hz)	Magnitud (dB)
1	800	-5.39
2	1600	-7.39
3	2400	-11
4	3200	-17.8
5	4000	ruido
6	4800	-21
7	5600	-18.2
8	6400	-19.4
9	7200	-25

Una vez tomado los datos de los armónicos experimentales, se implementa el siguiente código en MatLab para poder asignar las variables de la tabla a variables más simples y lograr visualizar las gráficas que se generan de las señales a partir de sus armónicos experimentales, con sus respectivos ejes y título.

```
T = 0.00125;
f = 1/T;
t = 0 : T/100 : 5*T;
Fs = 1/(t(2)-t(1));
senales = {
    'Cuadrada 1', [1 3 5 7 9 11],
    [-0.9 -11 -15.8 -17.8 -19.8 -22.2],
    0;
    'Cuadrada 2', [1 3 5 7 9 11],
    [-0.59 -10.6 -15.4 -17.8 -19.8 -21.8],
    1;
    'Pulso 1', [1 3 5 7 9 11],
    [-0.99 -10.2 -15 -17.8 -19.8 -21.8],
    0;
    'Pulso 2', [1:10],
    [-5.39 -7.39 -10.6 -17.4 0 -21 -18.2
    -19.4 -24.2 0], 0;
}
```

```
'Pulso 3', [1:10],
[-2.59 -7.39 -10.2 -20.2 17.8 -15 -21 -27.8
-19.4 -21.4 0 ...]

-23.4 -23 -35 -29 24.6 -30 -36 -26.2
27.8 0], 0;

'Pulso 4', [1:10],
[-5.39 -7.39 -11 -17.8 -56 -21 -18.2
-19.4 -25 -64], 0;

'Triangular 1', [1 3 5 7 9 11],
[-4.59 -24.6 -32.2 -38.6 -42.6 -46.2],
0;

'Triangular 2', [1 3 5 7 9 11],
[-4.19 -23.4 -32.6 -38.2 -41.8 -45.8],
1.25;

'Senoidal 1', [1],
[-2.59], 1;

'Senoidal 2', [1],
[-2.99], 0;

};

colores = lines(10);

figure('Position', [100, 100, 1200,
800]);

for k = 1:size(senales, 1)

    nombre = senales{k, 1};

    n_arm = senales{k, 2};

    dBs = senales{k, 3};

    offset = senales{k, 4};

    A = 10.^ (dBs (:)/20);

    A(dBs == 0) = 0;

    A(dBs > 0) = 0;

    vt = zeros(size(t));

    if contains(nombre, 'Triangular')

        for n = 1:length(n_arm)

            vt = vt + A(n) *
cos(2*pi*n_arm(n)*f*t);

        end

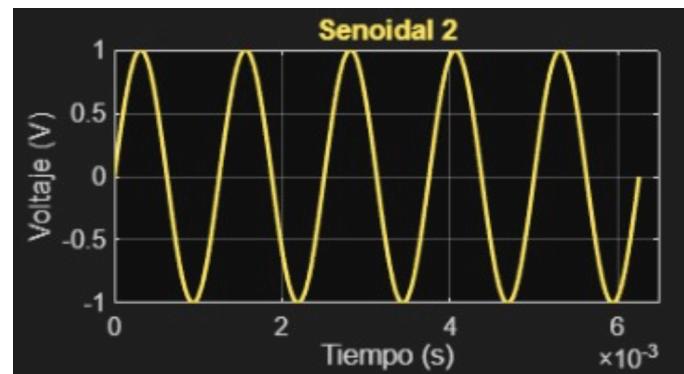
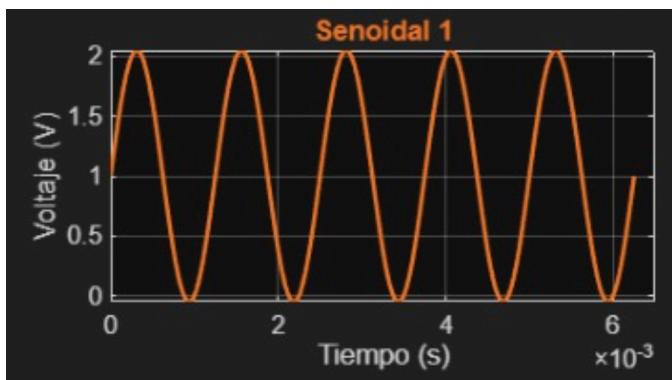
    else
```



```
for n = 1:length(n_arm)
    vt = vt + A(n) * sin(2*pi*n_arm(n)*f*t);
end
if offset ~= 0
    vt = offset + vt;
end
subplot(4, 3, k);
plot(t, vt, 'Color', colores(k,:),
'LineWidth', 1.5);
title(nombre, 'FontWeight', 'bold',
'Color', colores(k,:));
xlabel('Tiempo (s)');
ylabel('Voltaje (V)');
grid on;
xlim([0, 0.01]);
ylim auto;
end
sgtitle('Reconstrucción de señales a
partir de armónicas experimentales');
```

A continuación, se logran visualizar las reconstrucciones de las señales.

- Senoidal (Vp=1, DC=1)



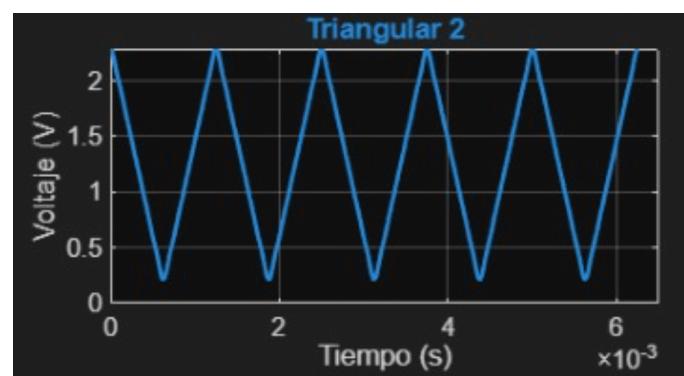
Gráfica 3. Senoidal en MatLab

- Triangular (Vp=1, DC=0)



Gráfica 3. Triangular en MatLab

- Triangular 2 (Vp=1, DC=1.5)



Gráfica 3. Triangular en MatLab

- Cuadrada (Vp=1, DC=0)

Gráfica 3. Senoidal en MatLab

- Senoidal 2 (Vp=1, DC=0)



Gráfica 3. Cuadrada en MatLab

- Cuadrada ($V_p=1$, DC=1.25)



Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, DC=0, Ciclo útil= 30%)



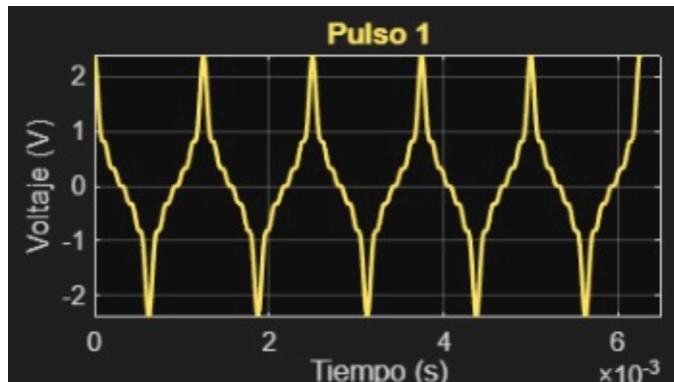
Gráfica 3. Cuadrada en MatLab

- Pulso 1 ($V_p=1$, DC=0, Ciclo útil= 50%)



Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, DC=0, Ciclo útil= 80%)



Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, DC=0, Ciclo útil= 20%)



Gráfica 3. Pulso en MatLab

Comparación de armónicos

A partir de las señales reconstruidas con los armónicos teóricos y experimentales, se pueden realizar comparaciones para poder verificar su precisión, exactitud y porcentaje de error, ya que las magnitudes teóricas están en unidades de



voltaje y las experimentales en decibeles, se debe realizar la respectiva conversión de los teóricos, por medio de las siguientes ecuaciones:

$$V_{rms} = \frac{V_p}{\sqrt{2}}$$

$$dB = 20 \times \log_{10}(V_{rms})$$

- Senoidal ($V_p=1$, DC=0)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-3,01	-2,99	0,6%

El error entre los valores teórico y experimental es muy bajo, esto indica una excelente coincidencia entre los resultados teóricos y experimentales.

- Senoidal ($V_p=1$, DC=1)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-3,01	-2,83	5,9%

La adición de un componente DC incrementa significativamente el error en la medición experimental del primer armónico de una señal senoidal.

- Triangular ($V_p=1$, DC=0)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-4,8	-4,59	4,3%
3	-23,92	-24,6	2,8%
5	-33,46	-32,2	3,7%
7	-38,92	-38,6	0,8%
9	-43,01	-42,6	0,95%
11	-46,61	-46,2	0,8%

La comparación de la tabla anterior muestra una buena concordancia entre los valores teóricos y experimentales, con una precisión que mejora para los armónicos de mayor orden.

- Triangular ($V_p=1$, DC=1.5)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-4,8	-4,52	5,8%
3	-23,92	-23,4	2,1%
5	-33,46	-32,6	2,5%
7	-38,92	-38,2	1,8%
9	-43,01	-41,8	2,8%
11	-46,61	-45,8	1,7%

La comparación de la tabla anterior muestra una buena concordancia entre los valores teóricos y experimentales, con una precisión que mejora para los armónicos de mayor orden, aunque el componente DC incrementa ligeramente el error en el primer armónico.

- Cuadrada ($V_p=1$, DC=0)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-0,93	-0,9	3,2%
3	-10,54	-11	4,3%
5	-15,05	-15,8	4,9%
7	-17,9	-17,8	0,5%
9	-20,08	-20,2	0,5%
11	-22,18	-22,2	0,1%

La comparación de la tabla anterior muestra una excelente concordancia entre los valores teóricos y experimentales, especialmente para los armónicos de mayor orden, con una precisión que mejora conforme aumenta el número del armónico.

- Cuadrada ($V_p=1$, DC=1.25)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-0,93	-0,88	5,3%
3	-10,54	-10,6	0,5%
5	-15,05	-15,4	2,3%
7	-17,9	-17,4	2,7%



9	-20,08	-19,8	1,3%
11	-22,18	-21,8	1,7%

La comparación de la tabla anterior muestra una buena concordancia entre los valores teóricos y experimentales, con una precisión que mejora para los armónicos de mayor orden, aunque el componente DC incrementa ligeramente el error en el primer armónico.

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 50%)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-0,93	-0,98	5,3%
3	-10,54	-10,2	3,2%
5	-15,05	-15	0,3%
7	-17,9	-17,8	0,5%
9	-20,08	-19,8	1,3%
11	-22,18	-21,8	1,7%

La comparación de la tabla anterior muestra una excelente concordancia entre los valores teóricos y experimentales, especialmente para los armónicos de mayor orden, con una precisión que mejora conforme aumenta el número del armónico.

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 20%)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-5,62	-5,39	4%
2	-7,44	-7,39	0,6%
3	-10,96	-10,6	3,2%
4	-17,62	-17,4	1,2%
5	0	ruido	
6	-21,14	-21	0,6%
7	-18,29	-18,2	0,4%
8	-19,48	-19,4	0,4%
9	-24,73	-24,2	2,1%

La comparación de la tabla anterior muestra una excelente concordancia entre los valores teóricos y experimentales, especialmente para los armónicos de mayor orden, con una precisión que mejora conforme aumenta el número del armónico.

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 30%)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-2,83	-2,67	5,6%
2	-7,44	-7,39	0,6%
3	-20,73	-20,2	2,5%
4	-17,62	17,8	1%
5	-15,4	-15	2,5%
6	-21,14	-21	0,6%
7	-28,04	-27,8	0,8%
8	-19,48	-19,4	0,4%
9	-21,87	-21,4	2,1%

La comparación de la tabla anterior muestra una excelente concordancia entre los valores teóricos y experimentales, especialmente para los armónicos de mayor orden, con una precisión que mejora conforme aumenta el número del armónico.

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 80%)

Armónico	Magnitud teórica (dB)	Magnitud experimental (dB)	%Error
1	-5,62	-5,39	4%
2	-7,44	-7,39	0,6%
3	-10,96	-11	0,3%
4	-17,62	-17,8	2,7%
5	0	ruido	
6	-21,14	-21	0,6%
7	-18,29	-18,2	0,4%
8	-19,48	-19,4	0,4%
9	-24,73	-25	1%



La comparación de la tabla anterior muestra una excelente concordancia entre los valores teóricos y experimentales, especialmente para los armónicos de mayor orden, con una precisión que mejora conforme aumenta el número del armónico.

Señales en tiempo (.csv)

Ya obtenidos los datos se van a exportar en formato CSV (Tabulator, decimal point) desde OpenChoiceDesk. Estos se van a descargar en el pc, para el uso de MatLab Drive y se realizan las siguientes acciones:



Imagen 3. Variable importada

Al seleccionar todos los datos, se le asigna un nombre corto a la tabla que los contiene, como se observa en la Imagen 3, esto con el propósito de poder identificar fácilmente la tabla de los datos a trabajar. Luego se selecciona la importación a realizar y se realiza una importación de datos, como se ilustra en la Imagen 5 a continuación.

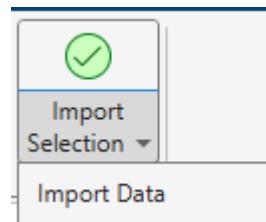


Imagen 5. Importar

Ya con las modificaciones mencionadas anteriormente, se debe observar la tabla de los datos como se ilustra en la imagen 6, cada columna con su variable correspondiente.

	A	B	C	D	E
	RecordLength	x2500	Points	x_0_0025	x0_14
1	Record Length	Number	2500 Points	-0.0025	0.14
2	Sample Interval	2e-06 s		-0.002498	0.16
3	Trigger Point	1250.0002	Samples	-0.002496	0.16
4				-0.002494	0.18

Imagen 7. Tabla de rodar

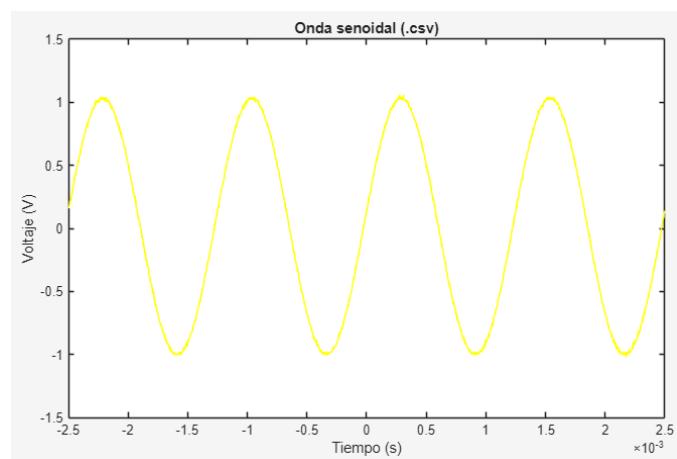
- Senoidal (Vp=1, DC=0)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda senoidal con un

voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=Q.x_0_0025;
v=Q.x0_14;
plot(t,v,Color='y')
xlabel("Tiempo (s)")
ylabel("Voltaje (V)")
title("Onda senoidal (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



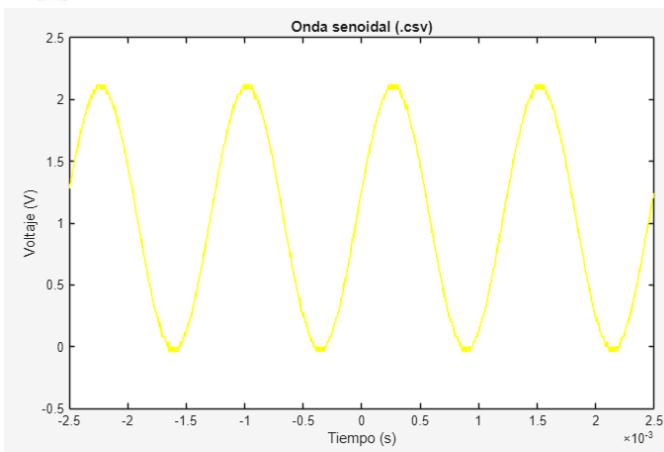
Gráfica 3. Senoidal en MatLab

- Senoidal (Vp=1, DC=1)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la onda senoidal con un voltaje pico de 1V, un offset de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=W.x_0_0025;
v=W.x1_28;
plot(t,v,Color='y')
xlabel("Tiempo (s)")
ylabel("Voltaje (V)")
title("Onda senoidal (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



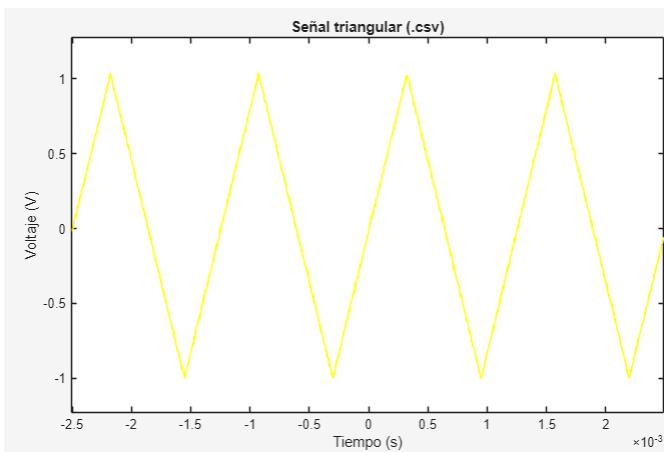
Gráfica 3. Senoidal en MatLab

- Triangular (Vp=1, DC=0)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal triangular con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=E.x_0_0025;  
  
v=E.x_0_04;  
  
plot(t,v,Color='y')  
  
xlabel("Tiempo (s)")  
  
ylabel("Voltaje (V)")  
  
title("Señal triangular (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



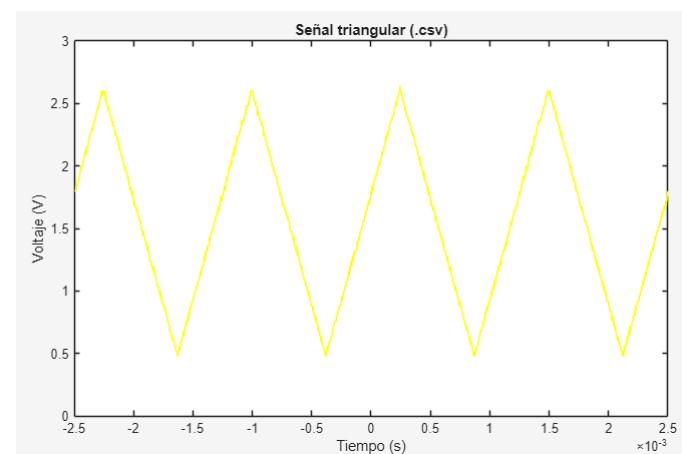
Gráfica 3. Triangular en MatLab

- Triangular (Vp=1, DC=1.5)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal triangular con un voltaje pico de 1V, un offset de 1.5V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=R.x_0_0025;  
  
v=R.x1_76;  
  
plot(t,v,Color='y')  
  
xlabel("Tiempo (s)")  
  
ylabel("Voltaje (V)")  
  
title("Señal triangular (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. Señal triangular (.csv)

- Cuadrada (Vp=1, DC=0)

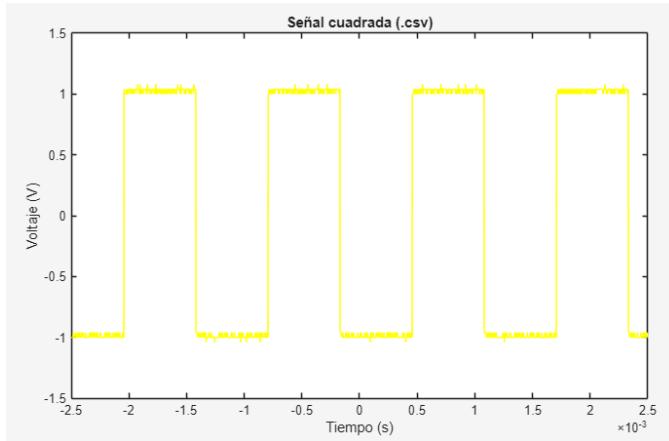
Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal cuadrada con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=T.x_0_0025;  
  
v=T.x_0_96;  
  
plot(t,v,Color='y')  
  
xlabel("Tiempo (s)")  
  
ylabel("Voltaje (V)")
```



```
title("Señal cuadrada (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



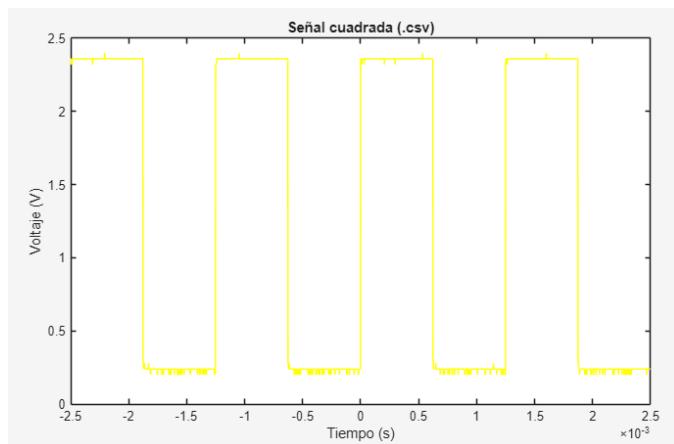
Gráfica 3. Cuadrada en MatLab

- Cuadrada ($V_p=1$, $DC=1.25$)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal cuadrada con un voltaje pico de 1V, un offset de 1.25V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=Y.x_0_0025;  
  
v=Y.x2_36;  
  
plot(t,v,Color='y')  
xlabel("Tiempo (s)")  
ylabel("Voltaje (V)")  
  
title("Señal cuadrada (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



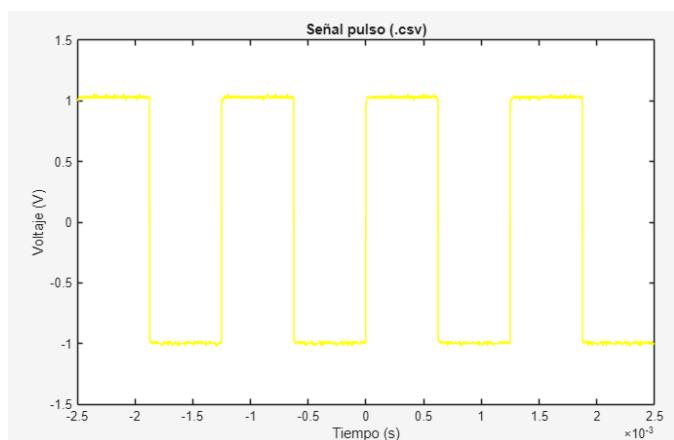
Gráfica 3. Cuadrada en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 50%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal pulso con un voltaje pico de 1V, un ciclo útil del 50% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=U.x_0_0025;  
  
v=U.x1;  
  
plot(t,v,Color='y')  
xlabel("Tiempo (s)")  
ylabel("Voltaje (V)")  
  
title("Señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. Pulso en MatLab

- Pulso ($V_p=1$, $DC=0$, Ciclo útil= 20%)

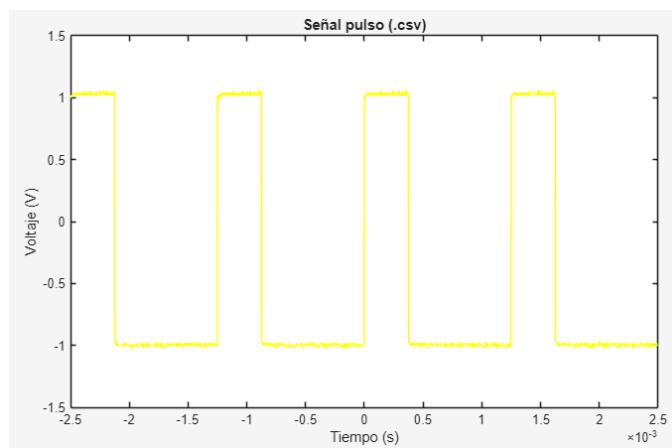


Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal pulso con un voltaje pico de 1V, un ciclo útil del 20% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=0.x_0_0025;  
  
v=0.x1_02;  
  
plot(t,v,Color='y')  
  
xlabel("Tiempo (s)")  
  
ylabel("Voltaje (V)")  
  
title("Señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.

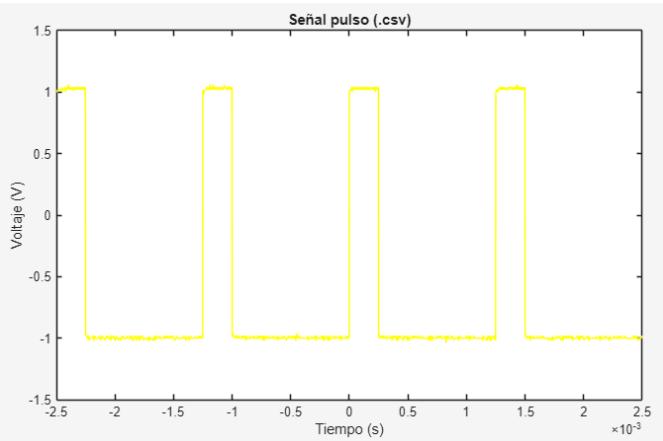
A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. Pulso en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 80%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal pulso con un voltaje pico de 1V, un ciclo útil del 80% y una frecuencia de 800Hz, con sus respectivos ejes y título.



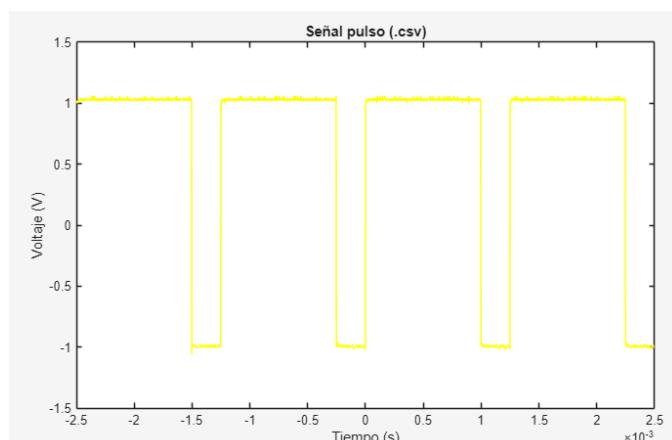
Gráfica 3. Pulso en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 30%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la señal pulso con un voltaje pico de 1V, un ciclo útil del 30% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
t=P.x_0_0025;  
  
v=P.x1;  
  
plot(t,v,Color='y')  
  
xlabel("Tiempo (s)")  
  
ylabel("Voltaje (V)")  
  
title("Señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.





Gráfica 3. Pulso en MatLab

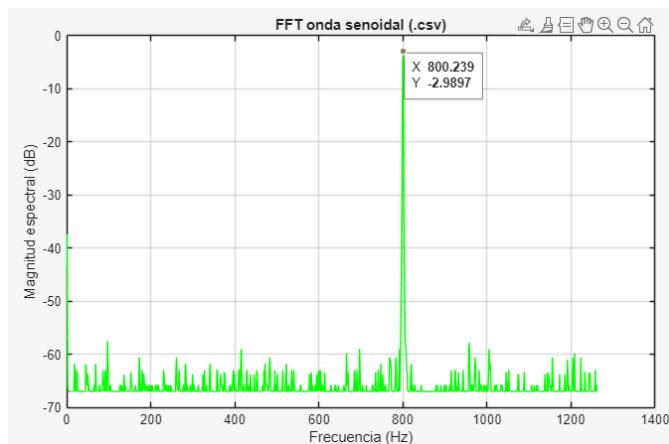
Señales en frecuencia (.csv)

- Senoidal ($V_p=1$, DC=0)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la onda senoidal con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=Q.x0;  
  
d=Q.x_34_1897;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT onda senoidal (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. FFT senoidal en MatLab

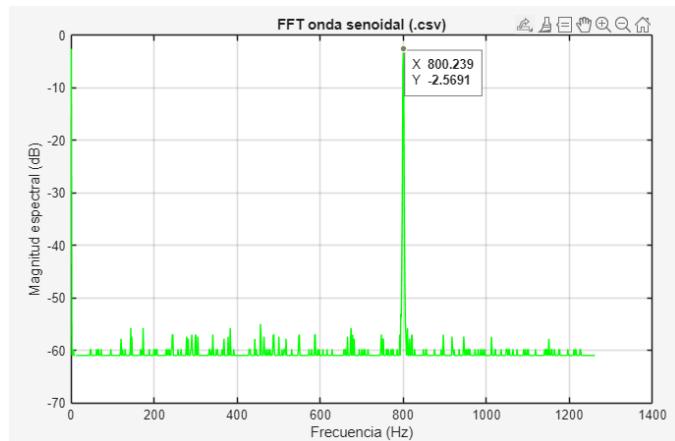
- Senoidal ($V_p=1$, DC=1)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la onda senoidal con un voltaje pico de 1V, un offset de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=W.x0;  
  
d=W.x0_2309;
```

```
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT onda senoidal (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



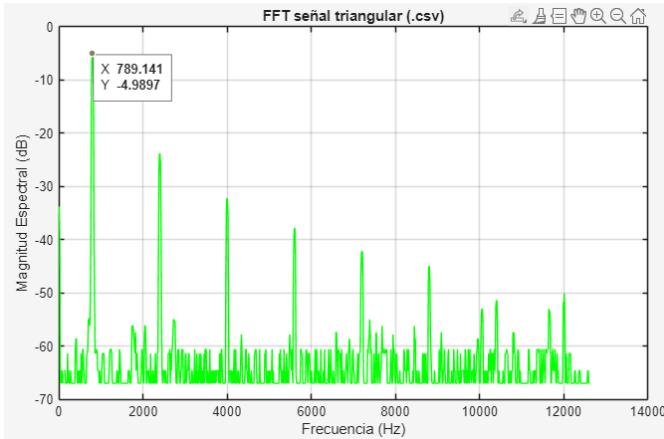
Gráfica 3. FFT senoidal en MatLab

- Triangular ($V_p=1$, DC=0)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal triangular con un voltaje pico de 1V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=E.x0;  
  
d=E.x_35_7897;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT señal triangular (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



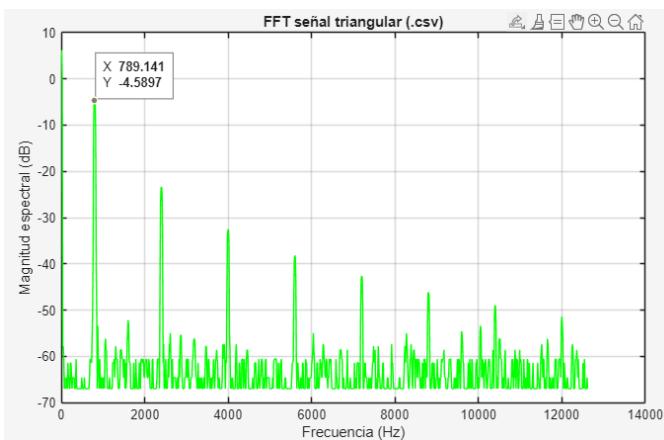
Gráfica 3. FFT triangular en MatLab

- Triangular ($V_p=1$, $DC=1.5$)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT señal triangular con un voltaje pico de 1V, un offset de 1.5V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=R.x0;
d=R.x3_8103;
plot(f,d,Color='g')
xlabel("Frecuencia (Hz)")
ylabel("Magnitud espectral (dB)")
title("FFT señal triangular (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



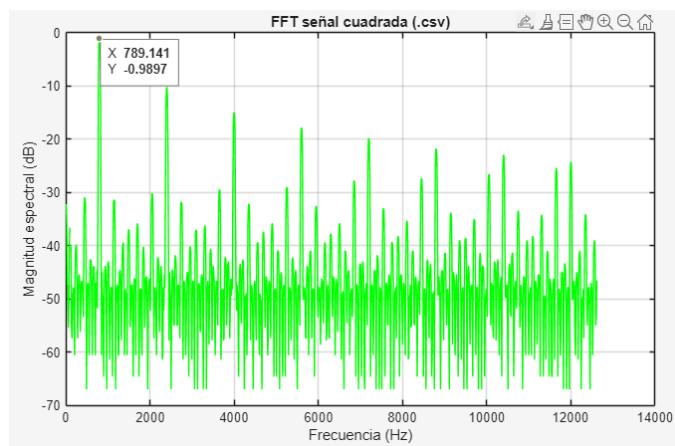
Gráfica 3. FFT triangular en MatLab

- Cuadrada ($V_p=1$, $DC=0$)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT señal cuadrada con un voltaje pico de 1V, un offset de 1.25V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=T.x0;
d=T.x_35_3897;
plot(f,d,Color='g')
xlabel("Frecuencia (Hz)")
ylabel("Magnitud espectral (dB)")
title("FFT señal cuadrada (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. FFT cuadrada en MatLab

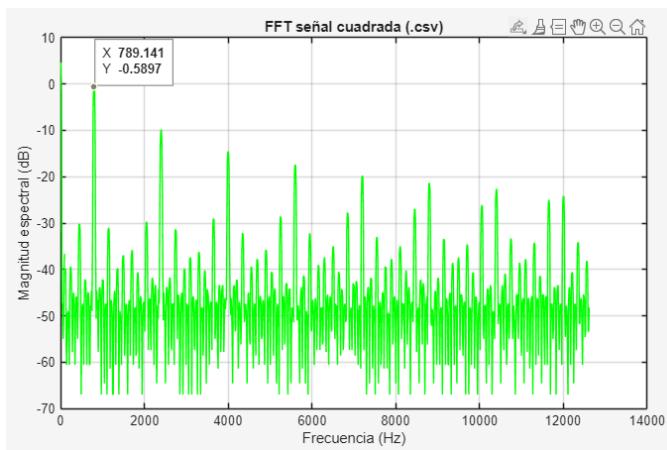
- Cuadrada ($V_p=1$, $DC=1.25$)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal cuadrada con un voltaje pico de 1V, un offset de 1.25V y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=Y.x0;
d=Y.x2_2103;
plot(f,d,Color='g')
xlabel("Frecuencia (Hz)")
ylabel("Magnitud espectral (dB)")
title("FFT señal cuadrada (.csv)")
```



A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



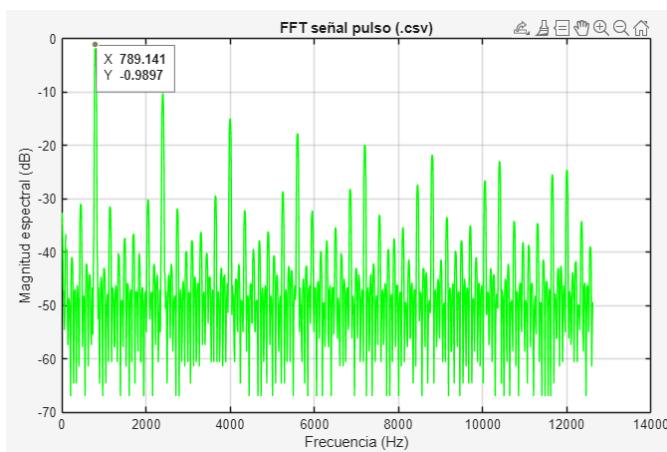
Gráfica 3. FFT cuadrada en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 50%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal pulso con un voltaje pico de 1V, un ciclo útil del 50% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=U.x0;  
  
d=U.x_34_9897;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



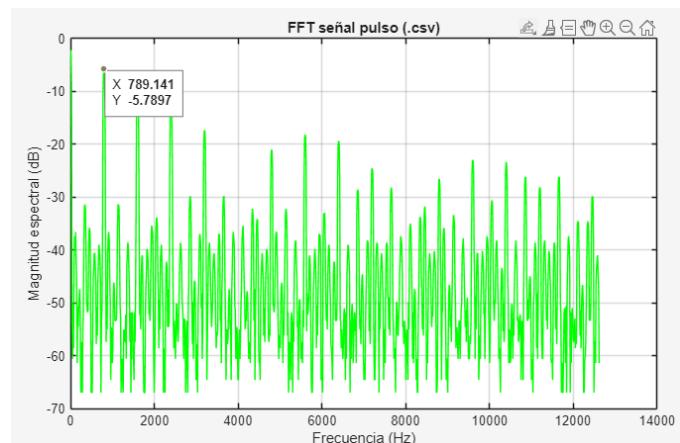
Gráfica 3. FFT pulso en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 20%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal pulso con un voltaje pico de 1V, un ciclo útil del 20% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=O.x0;  
  
d=O.x_4_5897;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. FFT pulso en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 30%)

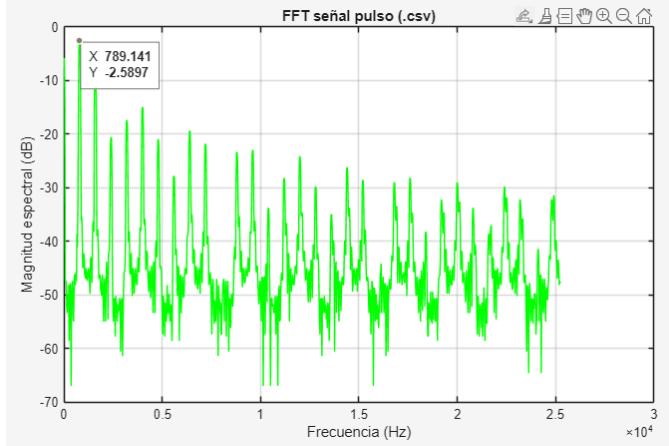
Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal pulso con un voltaje pico de 1V, un ciclo útil del 30% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=P.x0;  
  
d=P.x_8_189;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")
```



```
ylabel("Magnitud espectral (dB)")  
title("FFT señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



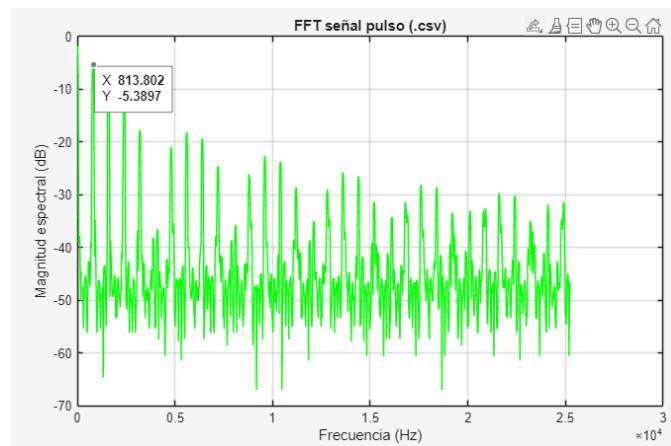
Gráfica 3. FFT pulso en MatLab

- Pulso (Vp=1, DC=0, Ciclo útil= 80%)

Se implementa el siguiente código para poder asignar las variables de la tabla a variables más simples y lograr visualizar la gráfica que se genera de la FFT de la señal pulso con un voltaje pico de 1V, un ciclo útil del 80% y una frecuencia de 800Hz, con sus respectivos ejes y título.

```
f=L.x0;  
  
d=L.x_4_1897;  
  
plot(f,d,Color='g')  
  
xlabel("Frecuencia (Hz)")  
  
ylabel("Magnitud espectral (dB)")  
  
title("FFT señal pulso (.csv)")
```

A continuación, en la Gráfica 3, se logran visualizar los datos obtenidos del osciloscopio.



Gráfica 3. FFT pulso en MatLab

Espectro en frecuencia de las señales mediante el cálculo de la FFT de los csv de tiempo

En esta sección, se requiere obtener el espectro frecuencial a partir de la transformada rápida de Fourier (FFT) aplicada a los datos registrados en el archivo CSV, los cuales fueron capturados experimentalmente en el dominio del tiempo. Con este fin, se empleó el software Matlab para desarrollar un programa que automatiza y facilita el proceso de análisis. A continuación, se presenta el código que se implementó:

```
x1 = x1senoidaltiempo.tiempo;  
  
y1 = x1senoidaltiempo.voltaje;  
  
x2 = x2senoidaltiempo.tiempo;  
  
y2 = x2senoidaltiempo.voltaje;  
  
x3 = x1triangulartiempo.tiempo;  
  
y3 = x1triangulartiempo.voltaje;  
  
x4 = x2triangulartiempo.tiempo;  
  
y4 = x2triangulartiempo.voltaje;  
  
x5 = x1cuadradoltiempo.tiempo;  
  
y5 = x1cuadradoltiempo.voltaje;  
  
x6 = x2cuadrado2tiempo.tiempo;  
  
y6 = x2cuadrado2tiempo.voltaje;  
  
x7 = x1pulsoltiempo.tiempo;  
  
y7 = x1pulsoltiempo.voltaje;
```



```
x8 = x2pulso2tiempo.tiempo;
y8 = x2pulso2tiempo.voltaje;
x9 = x3pulso3tiempo.tiempo;
y9 = x3pulso3tiempo.voltaje;
x10 = x4pulso4tiempo.tiempo;
y10 = x4pulso4tiempo.voltaje;

senales_t = {x1, x2, x3, x4, x5, x6,
x7, x8, x9, x10};

senales_v = {y1, y2, y3, y4, y5, y6,
y7, y8, y9, y10};

nombres = {'Senoidal 1', 'Senoidal
2', 'Triangular 1', 'Triangular 2',
...
'Cuadrada 1', 'Cuadrada
2', 'Pulso 1', 'Pulso 2', 'Pulso 3',
'Pulso 4'};

Fs = 500000;

figure('Position', [100, 100, 1200,
900]);

for k = 1:10

    t = senales_t{k};
    v = senales_v{k};

    validIdx = isnan(t) &
isnan(v);

    t = t(validIdx);
    v = v(validIdx);

    N = length(t);

    if N < 2

        fprintf('Advertencia: señal
%d tiene menos de 2 puntos
válidos.\n', k);

        continue;

    end

    f = (0:N/2) * (Fs / N);

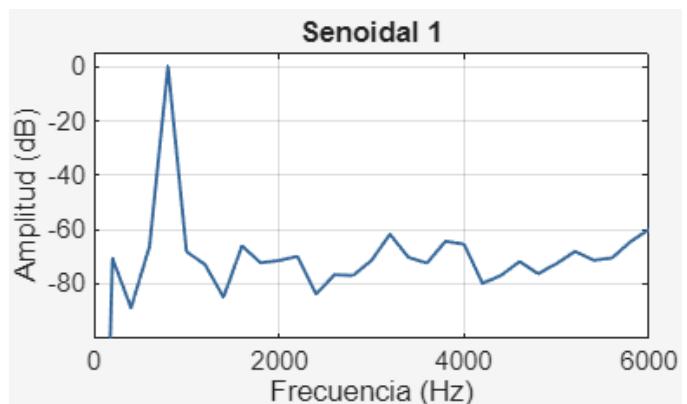
    v_ac = v - mean(v);

    Y = fft(v_ac);
    Y = Y(1:N/2+1);
    P1 = abs(Y) / N;
    P1(2:end-1) = 2 * P1(2:end-1);
    P1_dB = 20 * log10(P1 + eps);
    subplot(5,2,k);
    plot(f, P1_dB, 'Color', [0.2 0.4
0.6], 'LineWidth', 1.1);
    hold on;
    title(nombres{k});
    xlabel('Frecuencia (Hz)');
    ylabel('Amplitud (dB)');
    xlim([0, 6000]);
    ylim([max(P1_dB) - 100,
max(P1_dB) + 5]);
    grid on;
    hold off;
end

sgtitle('Espectros de FFT en dB - 10
señales (fo = 800 Hz)');
```

Una vez ejecutado el código se generan los siguientes espectros de cada señal:

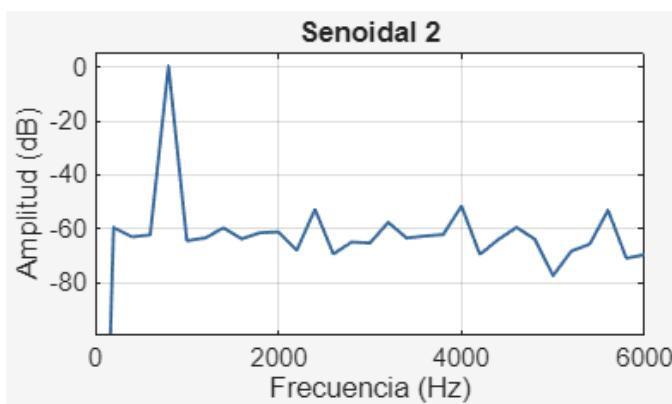
- Senoidal ($V_p=1$, $DC=0$)



Gráfica 3. FFT senoidal en MatLab

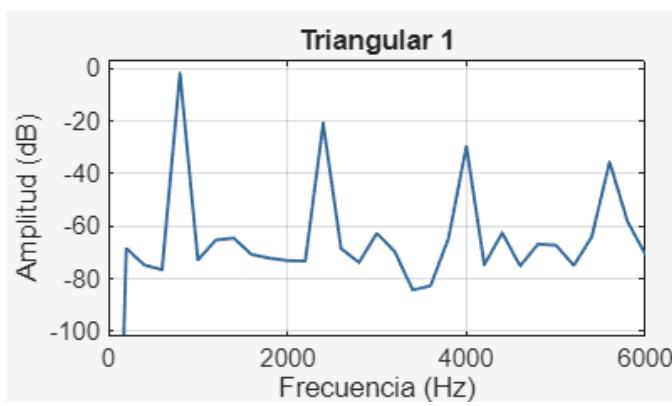


- Senoidal ($V_p=1$, $DC=1$)



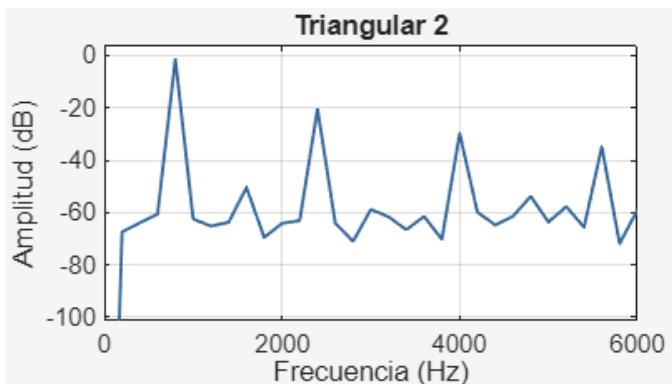
Gráfica 3. FFT senoidal en MatLab

- Triangular ($V_p=1$, $DC=0$)



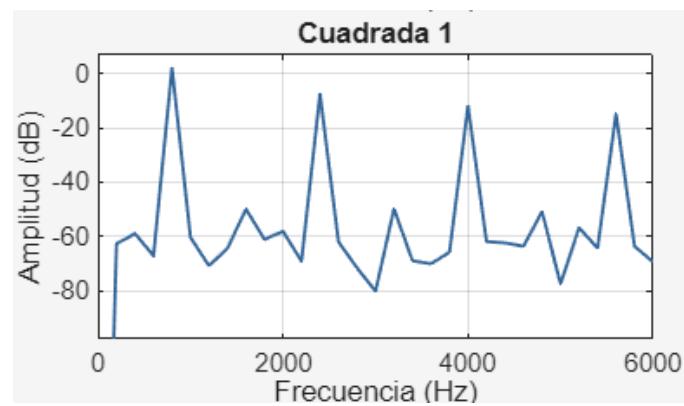
Gráfica 3. FFT triangular en MatLab

- Triangular ($V_p=1$, $DC=1.5$)



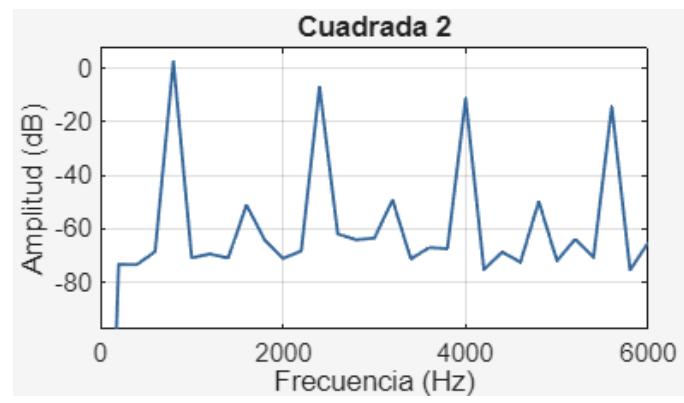
Gráfica 3. FFT triangular en MatLab

- Cuadrada ($V_p=1$, $DC=0$)



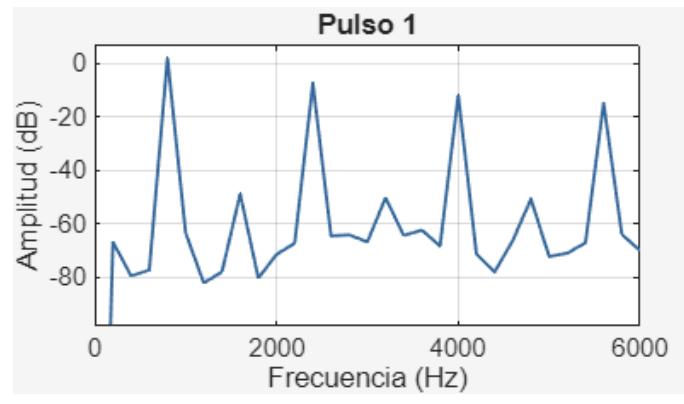
Gráfica 3. FFT cuadrada en MatLab

- Cuadrada ($V_p=1$, $DC=1.25$)



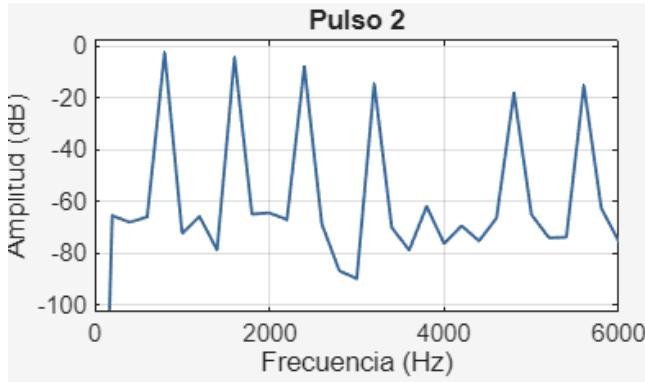
Gráfica 3. FFT cuadrada en MatLab

- Pulso 1 ($V_p=1$, $DC=0$, Ciclo útil= 50%)



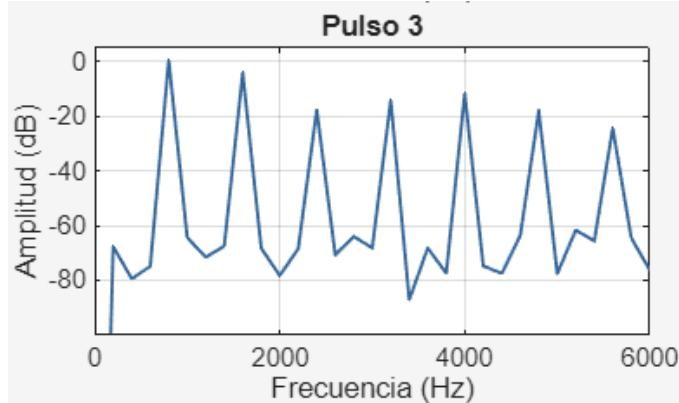
Gráfica 3. FFT pulso en MatLab

- Pulso 2 ($V_p=1$, $DC=0$, Ciclo útil= 20%)



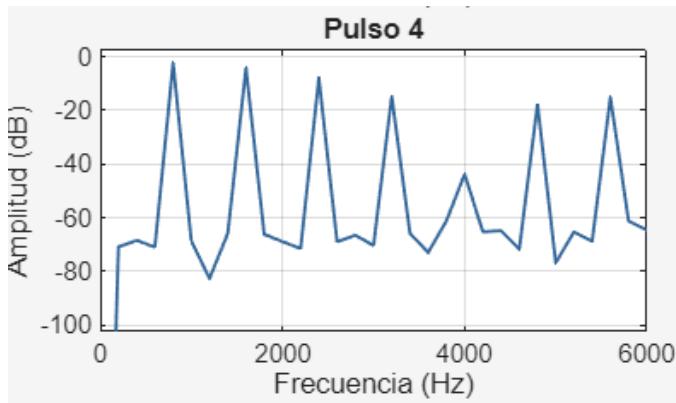
Gráfica 3. FFT pulso en MatLab

- Pulso 3 (Vp=1, DC=0, Ciclo útil= 30%)



Gráfica 3. FFT pulso en MatLab

- Pulso 4 (Vp=1, DC=0, Ciclo útil= 80%)



Gráfica 3. FFT pulso en MatLab

Comparación espectro obtenido de FFT de csv en dominio tiempo con el espectro capturado en el laboratorio

Se puede observar que los picos en cada una de las señales son similares, lo que indica una alta fidelidad en los valores fundamentales y relevantes. Sin embargo, es evidente que en algunas gráficas se presenta una menor resolución, lo cual se debe principalmente a los datos utilizados para calcular la FFT. Dado que estos datos provienen del archivo CSV correspondiente al dominio del tiempo, es posible que carezcan de la densidad o precisión necesaria para generar una representación espectral detallada. Esto subraya la importancia de garantizar que los datos temporales sean suficientemente densos y de alta calidad antes de realizar una transformación de Fourier, ya que la calidad inicial de los datos tiene un impacto directo en la exactitud del análisis espectral final.

Reconstrucción de la señal en el tiempo con el espectro en csv

En esta última parte del informe se intentó realizar la reconstrucción de las señales en el dominio del tiempo a partir del archivo CSV que contiene el espectro de frecuencia. Sin embargo, no fue posible obtener una reconstrucción precisa debido a varios factores clave.

En primer lugar, el archivo solo registraba la magnitud del espectro, sin incluir la información de fase. Esta última es fundamental para recuperar la forma original de la señal, ya que define cómo se alinean temporalmente los diferentes componentes frecuenciales. Su ausencia impide que la señal reconstruida reproduzca adecuadamente eventos como transitorios o saltos bruscos, incluso si las magnitudes son correctas.

Otro problema fue la baja resolución frecuencial del espectro, originada por una ventana de tiempo demasiado corta durante la adquisición o el cálculo de la transformada de Fourier. Esta limitación dificulta la representación de detalles finos de la señal, como armónicos cercanos entre sí o variaciones rápidas en el tiempo.

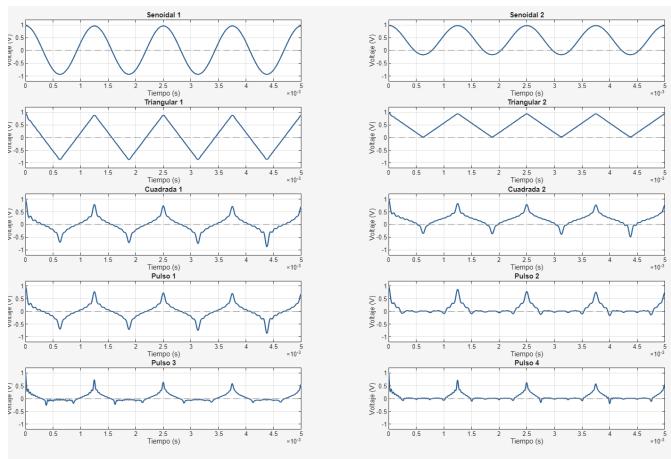
Además, es probable que el espectro proporcionado fuera unilateral (solo frecuencias positivas), mientras que la reconstrucción precisa de señales reales requiere un espectro bilateral que incluya también las frecuencias negativas. El formato del archivo CSV no contemplaba esta información, lo que imposibilitaba una inversión adecuada.

Por último, es probable que los datos almacenados en el CSV hubieran sufrido errores de cuantización o truncamiento durante su exportación, especialmente si se utilizó una precisión numérica limitada. Estas pequeñas inexactitudes se amplificaron durante el proceso de reconstrucción, introduciendo ruido y distorsión adicional.

En conjunto, la falta de fase, la baja resolución, la ausencia de frecuencias negativas y las imprecisiones numéricas explican por qué las señales reconstruidas presentaron una forma



claramente distorsionada en comparación con la original. A continuación se observan las gráficas obtenidas:



Gráfica 3. Señales en el tiempo con sus espectros en MatLab

Reconstrucción de la señal en el tiempo con IFFT

De esta manera, se encontró una solución adecuada la cual fue realizar el siguiente procedimiento:

1. Tomar los datos csv del tiempo
2. Transformarlos a espectro con FFT
3. Tomar datos transformados y aplicarle IFFT
4. Visualizar resultados

Por consiguiente, se generó el siguiente código:

```
x1 = x1senoidaltiempo.tiempo;  
  
y1 = x1senoidaltiempo.voltaje;  
  
x2 = x2senoidaltiempo.tiempo;  
  
y2 = x2senoidaltiempo.voltaje;  
  
x3 = x1triangulart tiempo.tiempo;  
  
y3 = x1triangulart tiempo.voltaje;  
  
x4 = x2triangulart tiempo.tiempo;  
  
y4 = x2triangulart tiempo.voltaje;  
  
x5 = x1cuadradoltiempo.tiempo;  
  
y5 = x1cuadradoltiempo.voltaje;  
  
x6 = x2cuadrado2tiempo.tiempo;  
  
y6 = x2cuadrado2tiempo.voltaje;  
  
x7 = x1pulsoltiempo.tiempo;
```

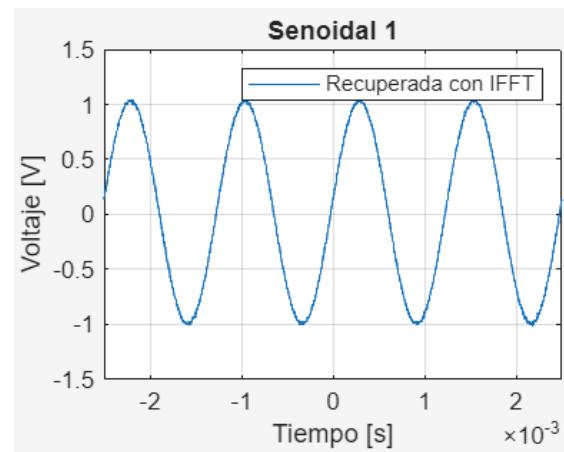
```
y7 = x1pulsoltiempo.voltaje;  
  
x8 = x2pulso2tiempo.tiempo;  
  
y8 = x2pulso2tiempo.voltaje;  
  
x9 = x3pulso3tiempo.tiempo;  
  
y9 = x3pulso3tiempo.voltaje;  
  
x10 = x4pulso4tiempo.tiempo;  
  
y10 = x4pulso4tiempo.voltaje;  
  
senales_t = {x1, x2, x3, x4, x5, x6,  
x7, x8, x9, x10};  
  
senales_v = {y1, y2, y3, y4, y5, y6,  
y7, y8, y9, y10};  
  
nombres = {'Senoidal 1', 'Senoidal  
2', 'Triangular 1', 'Triangular 2',  
...  
'Cuadrada 1', 'Cuadrada  
2', 'Pulso 1', 'Pulso 2', 'Pulso 3',  
'Pulso 4'};  
  
Fs = 500000;  
  
figure('Position', [100, 100, 1200,  
900]);  
  
for k = 1:10  
  
    t = senales_t{k};  
  
    v = senales_v{k};  
  
    validIdx = isnan(t) &  
    isnan(v);  
  
    t = t(validIdx);  
  
    v = v(validIdx);  
  
    N = length(t);  
  
    if N < 2  
  
        fprintf('Advertencia: señal  
%d tiene menos de 2 puntos  
válidos.\n', k);  
  
        continue;  
  
    end  
  
    f = (0:N/2) * (Fs / N);
```



```
v_dc = mean(v);  
  
v_ac = v - v_dc;  
  
Y_full = fft(v_ac);  
  
Y_pos = Y_full(1:N/2+1);  
  
P1 = abs(Y_pos) / N;  
  
P1(2:end-1) = 2 * P1(2:end-1);  
  
P1_dB = 20 * log10(P1 + eps);  
  
v_recuperada_ac = ifft(Y_full);  
  
v_recuperada_ac =  
real(v_recuperada_ac);  
  
v_recuperada = v_recuperada_ac +  
v_dc;  
  
subplot(3, 3, k);  
  
plot(t, v_recuperada,  
'DisplayName', 'Recuperada con  
IFFT');  
  
hold off;  
  
title(nombres{k});  
  
xlabel('Tiempo [s]');  
  
ylabel('Voltaje [V]');  
  
legend('Location', 'northeast');  
  
grid on;  
  
end  
  
linkaxes(findobj(gcf, 'Type',  
'axes'), 'x');
```

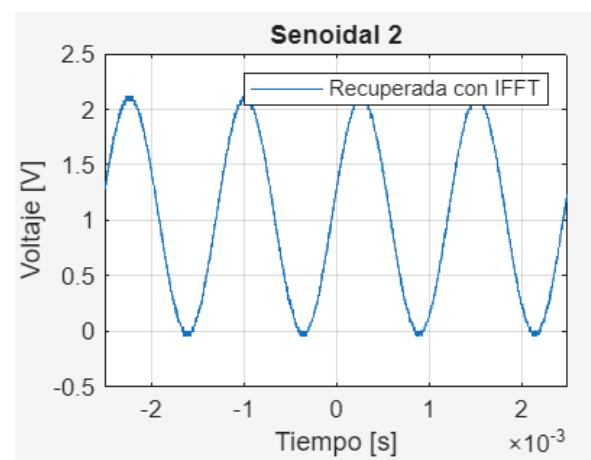
Una vez finalizado el código se procede a ejecutarlo y analizar las gráficas a continuación:

- Senoidal (Vp=1, DC=0)



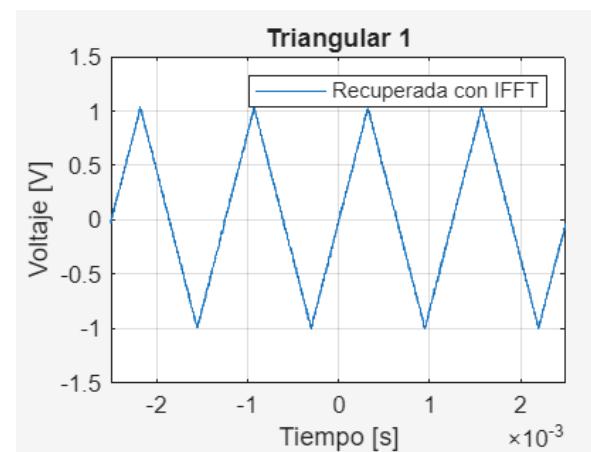
Gráfica 3. IFFT senoidal en MatLab

- Senoidal (Vp=1, DC=1)



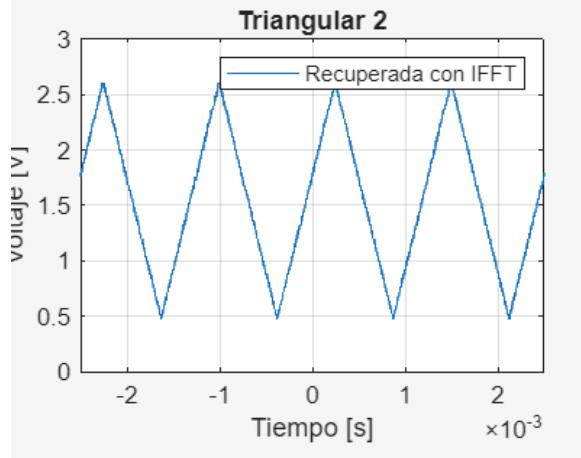
Gráfica 3. IFFT senoidal en MatLab

- Triangular (Vp=1, DC=0)



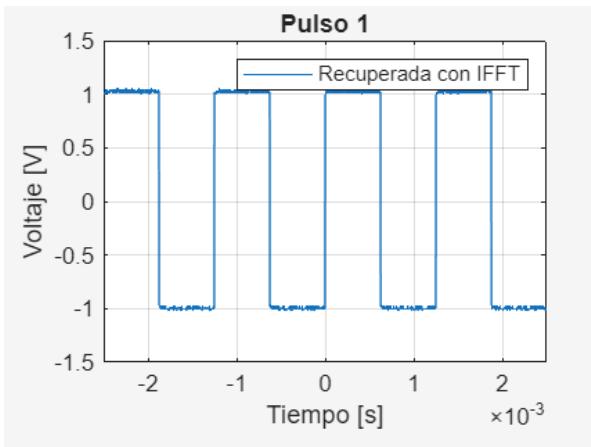
Gráfica 3. IFFT triangular en MatLab

- Triangular (Vp=1, DC=0)



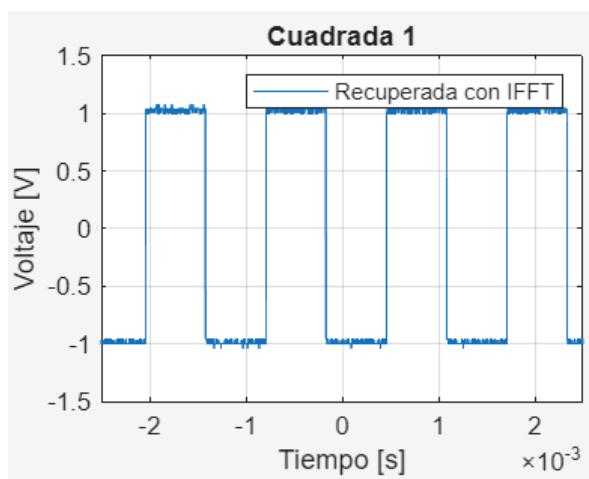
Gráfica 3. IFFT triangular en MatLab

- Pulso 1 ($V_p=1$, $DC=0$, Ciclo útil= 50%)



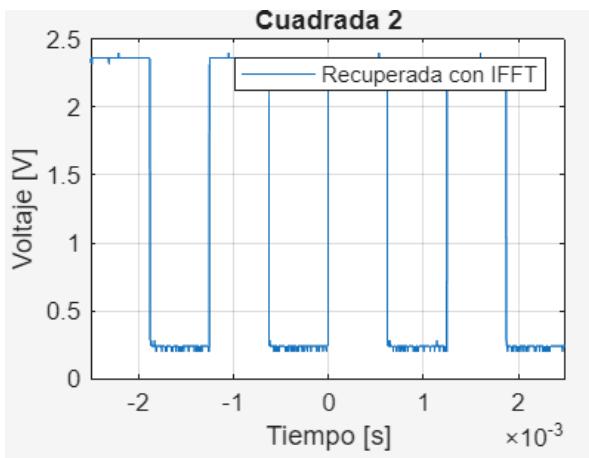
Gráfica 3. IFFT cuadrada en MatLab

- Cuadrada ($V_p=1$, $DC=0$)



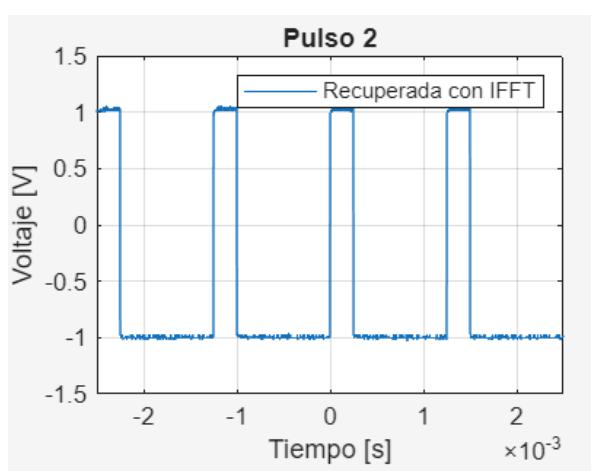
Gráfica 3. IFFT cuadrada en MatLab

- Cuadrada ($V_p=1$, $DC=1.25$)



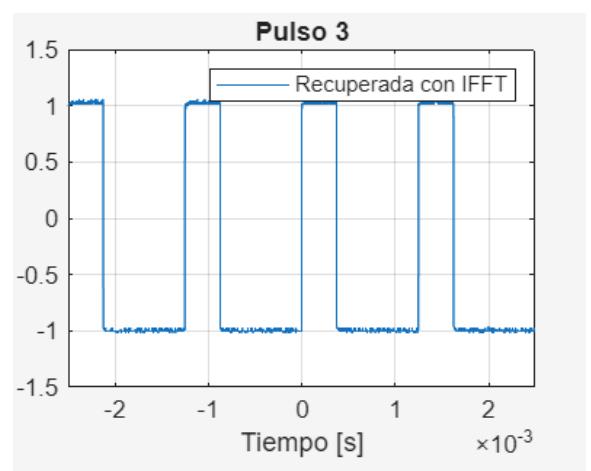
Gráfica 3. IFFT cuadrada en MatLab

- Pulso 2 ($V_p=1$, $DC=0$, Ciclo útil= 20%)



Gráfica 3. IFFT pulso en MatLab

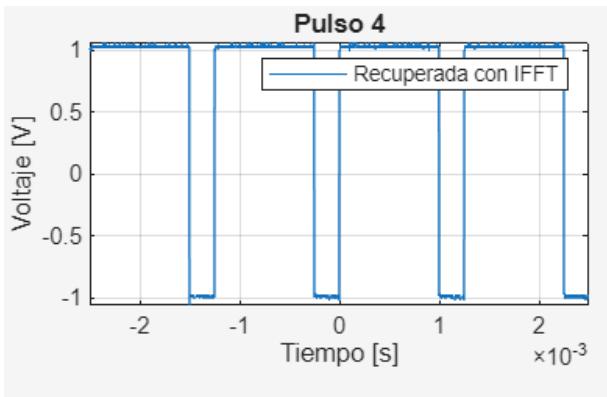
- Pulso 3 ($V_p=1$, $DC=0$, Ciclo útil= 30%)





Gráfica 3. IFFT pulso en MatLab

- Pulso 4 (Vp=1, DC=0, Ciclo útil= 80%)



Gráfica 3. IFFT pulso en MatLab

IV. CONCLUSIONES

Este laboratorio demostró con éxito la relación entre el dominio temporal y frecuencial de diversas señales periódicas, validando teóricamente los principios de las series de Fourier mediante mediciones experimentales. Los resultados obtenidos revelan una excelente concordancia entre los valores teóricos y experimentales de los armónicos, con porcentajes de error que generalmente disminuyen conforme aumenta el orden del armónico, lo que confirma que los componentes de alta frecuencia se miden con mayor precisión en este entorno experimental.

La adición de componentes DC afecta principalmente al primer armónico, incrementando ligeramente el error en su medición, mientras que los armónicos superiores mantienen una alta precisión. Este hallazgo subraya la sensibilidad del armónico fundamental a perturbaciones en la medición, probablemente debido a factores como ruido o interferencias en la banda de frecuencias más bajas.

Un aspecto crítico identificado fue la imposibilidad de reconstruir con precisión las señales en el dominio temporal a partir de los espectros frecuenciales proporcionados, debido principalmente a la falta de información de fase, la resolución frecuencial insuficiente, la naturaleza unilateral de los espectros y errores de cuantización en los datos. Este resultado es fundamental, ya que evidencia que la magnitud espectral por sí sola no es suficiente para recuperar la forma exacta de una señal; la información de fase es igualmente crucial para una reconstrucción precisa.

Finalmente, el proceso de reconstrucción utilizando IFFT a partir de los datos temporales originales demostró ser efectivo, validando la reversibilidad de la transformada de Fourier cuando se dispone de información completa (magnitud y fase). Este trabajo refuerza la comprensión de los fundamentos del procesamiento de señales y la importancia de considerar tanto

la magnitud como la fase en el análisis espectral, con implicaciones prácticas en diversas aplicaciones de ingeniería de telecomunicaciones y procesamiento digital de señales.

V. REFERENCIAS

1. Sistemas de comunicaciones electrónicas. Cuarta Edición Tomasi, Wayne. Capítulo 1.
2. Analog and Digital Communications. Second Edition. Schaum's outlines. Chapter 1.
3. Sistemas electrónicos de Comunicaciones. Louis E. Frenzel. Alfaomega

VI. REPOSITORIO DE GITHUB

<https://github.com/Vallentincita/Comunicaci-n-digital-TEL-A->