

Practica 1: Muestreo

Reconocimiento de Voz

Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala Instituto
Politécnico Nacional, Tlaxcala, Tlaxcala, México

Ingeniera en Inteligencia Artificial

19 de septiembre 2024

Castro Elvira D. ²

Valeria Jahzeel Castañón H. ¹

Nancy Galicia C ⁴

Miguel Angel Sanchez Z. ¹⁸

I. INTRODUCCION

Dentro del procesamiento de señales, el muestreo es una técnica que convierte una señal continua en el tiempo en una secuencia de valores discretos. Esta conversión es esencial para el análisis y procesamiento digital de señales, ya que la mayoría de los sistemas modernos, como computadoras y dispositivos de telecomunicaciones, funcionan con señales discretas.

En esta práctica, se llevará a cabo el muestreo de una señal sinusoidal, una de las formas de onda más básicas y comunes en telecomunicaciones. Se explorarán diferentes frecuencias de muestreo y se observará cómo estas afectan la reconstrucción de la señal.

El objetivo de esta práctica es comprender y aplicar los conceptos básicos del proceso de muestreo, tales como la frecuencia de muestreo y su impacto en la fidelidad de la señal muestreada.

Objetivos

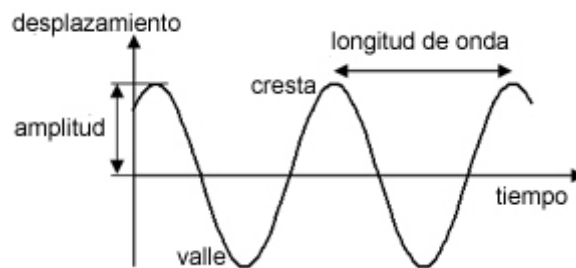
- Desarrollar un programa que permita el muestreo de una señal
- Comprender los conceptos básicos relacionados con el muestreo de señales

Marco teórico

Señal

En términos generales, una señal es una perturbación que se propaga de forma dinámica en un medio que transportan información.

Componentes de una señal



- Amplitud (A): La altura de la señal, que representa la intensidad de la información.
- Frecuencia (f): El número de ciclos que la señal completa en un segundo, que afecta la tonalidad o el timbre de la señal. Se mide en hercios (Hz).
- Fase: La posición de la señal en el tiempo, que puede influir en cómo se combinan señales diferentes.
- Longitud de onda: Es la distancia entre dos máximos o compresiones consecutivos
- Tiempo (t): Es el tiempo total que dura la señal. Se mide en cualquier tipo de tiempo (minutos, segundos, milisegundos, etc.)
- Desplazamiento (ϕ): Es una medida de como se ha movido o desfazado la señal en el tiempo respecto a su posición original. Se mide en grados o radianes.

Tipos de señales

- Señales analógicas. Señal que representa alguna magnitud física como temperatura, intensidad luminosa, energía, presión, sonido o campo eléctrico, son señales analógicas que varían constantemente y pueden tomar todos los valores posibles de un intervalo.
- Señales digitales. Señales discretas (valores finitos) en el tiempo y en amplitud; por lo tanto, sólo puede tomar uno de dos valores —0 o 1— en intervalos definidos de tiempo.
- Señales continuas. Señales que mantienen un valor constante en el tiempo, como una corriente o un voltaje constantes.
- Señales de corriente alterna (AC). Señales que varían de forma periódica como puede ser las ondas sinusoidales
- Señales de corriente directa (DC). Señales que mantiene un valor constante
- Señales de pulsos. Señales que alternan rápidamente entre valores altos y bajos

Señales sinusoidales:

Una señal sinusoidal es una función matemática que describe una onda periódica, es decir que se repite cada cierto tiempo. Este tipo de señales se definen por su amplitud, frecuencia y fase. La fórmula general para obtener una señal sinusoidal es:

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

En donde A es la amplitud, f es la frecuencia en Hz , t es el tiempo en segundos y ϕ es el desplazamiento o desfase inicial de la señal en el tiempo, se representa en radianes.

Muestreo

El muestreo es un proceso que se aplica a las ondas para pasarlas de su forma analógica (continua) a su forma digital (discreta), este proceso consiste en poner un “punto” cada cierto tiempo para obtener información de la onda original, la cantidad de muestras o puntos que se toman depende de algunas variables:

- Frecuencia máxima de la señal (F_{\max}): se refiere a cuantas veces se repite un ciclo de la señal en un segundo, esto ayuda a determinar la frecuencia de muestreo ya que, si es muy baja en relación con F_{\max} , puede no ser capaz de capturar la señal original.
- Frecuencia de muestreo (F_s): es el número de muestras que se toman por segundo y es una de las variables más importantes para que la señal muestreada conserve la mayor cantidad de información de la señal original (Figura 1). Según el teorema de Nyquist-Shannon para evitar la pérdida de información la frecuencia de muestreo debe ser al menos el doble de la F_{\max} de la señal original; es decir, entre mayor sea la frecuencia de muestreo, mayor será la información que se conserve respecto a la onda original.

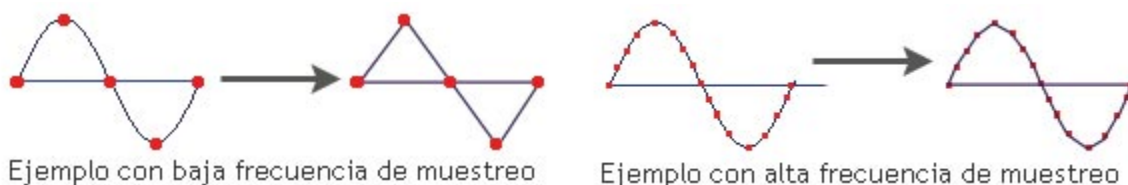


Figura 1. Comparación entre una baja frecuencia de muestreo y una alta frecuencia de muestreo

- Periodo de muestreo (T_s): es el tiempo que pasa entre una muestra y la siguiente, se usa para determinar cada cuanto tiempo se toma una muestra de la señal original; entre más pequeño sea el periodo de muestreo, más muestras se tomarán por segundo por lo que un periodo de muestreo grande menos muestras se tomarán por segundo. Se obtiene con la siguiente formula:

$$T_s = \frac{1}{F_s}$$

- Tiempo o duración de la señal (T): corresponde al tiempo que dura o existe una señal (algo así como su tiempo de vida). Esto afecta a la cantidad de muestras que se toman en total.

II. DESARROLLO

II.1. PARTE 1: MUESTREO

Limpiar variables y pantalla

```
clc;  
clear;
```

Parámetros de la señal **variables**

```
% Parámetros de muestreo  
fs = 1000; % Frecuencia de muestreo (Hz)  
tiempo = 20; % Tiempo total de grabación (milisegundos)
```

Parámetros de la señal **constantes** (no cambian, se hacen conversiones necesarias para trabajar el tiempo en *ms*)

```
% Parámetros de la señal  
A = 2; % Amplitud de la señal  
des = deg2rad(45); % Desplazamiento de fase  
f = 125; % Frecuencia de la señal (Hz)  
  
% Tiempo continuo y muestreo  
T = tiempo / 1000; % Convertir tiempo a segundos  
Ts = 1 / fs; % Periodo de muestreo  
t = 0:Ts:T-Ts; % Vector de tiempo discreto (muestreo)
```

Crear vectores para:

- Señal original (alta precisión)
- Señal muestreada

```
% Definir un tiempo continuo  
t_original = 0:1e-6:T;  
  
% Señal original  
x_original = A * sin(2 * pi * f * t_original + des);  
  
% Señal muestreada  
x_sampled = A * sin(2 * pi * f * t + des);
```

Mostrar el valor de la señal de cada muestreo

```
% Valores de las muestras
disp('Valores de las muestras:');
```

Valores de las muestras:

```
disp(x_sampled);
```

```
1.4142    2.0000    1.4142    0.0000   -1.4142   -2.0000   -1.4142   -0.0000    1.4142    2.0000
1.4142    0.0000   -1.4142   -2.0000   -1.4142   -0.0000    1.4142    2.0000    1.4142    0.0000
```

Crear gráfica donde:

- Señal original: Verde (alta precisión)
- Puntos de muestreo: Rojo
- Señal muestreada: Azul (unión de los puntos de muestreo)

```
figure;

% Señal original
plot(t_original, x_original, 'g', 'LineWidth', 1.5);
hold on; % Mantener las gráficas en una sola imagen

% Puntos de muestreo
stem(t, x_sampled, 'r', 'LineWidth', 1.5);

% Señal muestreada
plot(t, x_sampled, 'b', 'LineWidth', 1.5);

% Gráfica
title('Señal original y muestreada');
xlabel('Tiempo (s)');
ylabel('Amplitud');
legend('Señal original', 'Puntos de muestreo', 'Señal muestreada');
grid on;
hold off;
```

II.2. PARTE 2: DESARROLLO DE UNA INTERFAZ

Para la interfaz se utilizó App Designer, donde se consideraron las siguientes señales:

```
signals = {
    'seno', @sin, app.CBseno.Value, 'b';
    'coseno', @cos, app.CBcoseno.Value, 'r';
    'cuadrada', @square, app.CBcuadrada.Value, 'g';
    'triangular', @sawtooth, app.CBtriangular.Value, 'm';
};
```

De las cuales a cada una se grafica se grafica en el mismo plot para comparalas. Además se agregaron las funciones de sumar las señales, esto suma cada una de ellas y se grafica el resultado, también se agregó la opción de aplicar ruido gaussiano a cada uno de las señales.

La interfaz considera el siguiente flujo de trabajo:

1. Ingreso de datos: Se espera recibir del usuario la amplitud de la señal, la frecuencia en hercios, la desfase en grados, la frecuencia de muestro en hercios y el tiempo de grabación en milisegundos. Estos datos se les aplicara a cada señal y son los mismos para todas las señales disponibles, por defecto ya vienen ingresado unos datos de muestra.

Amplitud	<input type="text" value="2"/>		Frecuencia de muestro	<input type="text" value="1000"/>	Hz
Frecuencia	<input type="text" value="125"/>	Hz	Tiempo	<input type="text" value="10"/>	milisegundos
Desfase	<input type="text" value="45"/>	grados	<button>Actualizar Datos</button>		

2. Actualización de datos: Una vez que los datos son ingresados se le debe presionar el botón amarillo que dice “Actualizar Datos” esto lo que realiza es cargar los datos en las variables y se comienza grafica automáticamente. Cada vez que el usuario modifica algún dato debe actualizar para ver los cambios.

3. Funciones:

El sistema tiene las funciones previamente mencionadas (Seno, coseno, cuadrada, triangular), de los cuales el usuario puede marcar la casilla para su selección y graficacion. El sistema automáticamente grafica en todos los espacios (Normal, Fase, FFT, FFT centrada) y al desmarcar la casilla esta se quitará del espacio de trabajo y ya no es considerada. Estas acciones de desmarcar y marcar la señal se pueden hacer tantas veces se desea, por defecto viene seleccionada la función seno

Funciones

- ☒ Seno
- ☐ Coseno
- ☐ Cuadrada
- ☐ Triangular

4. Acciones:

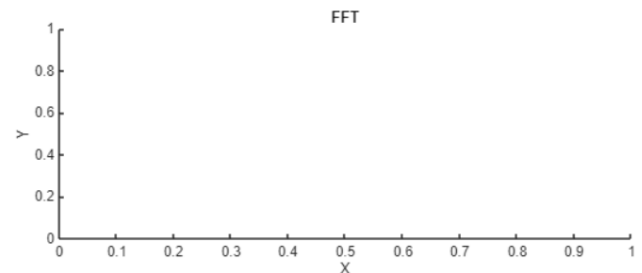
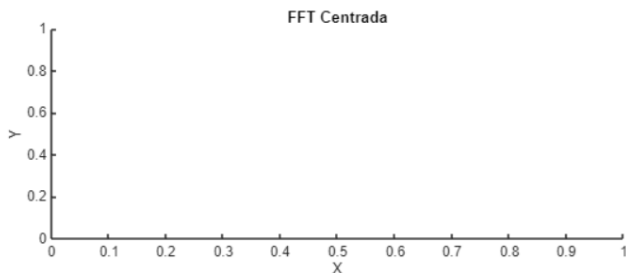
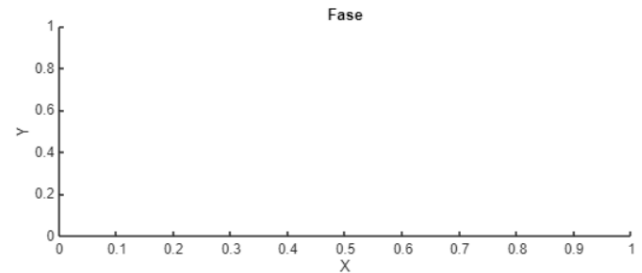
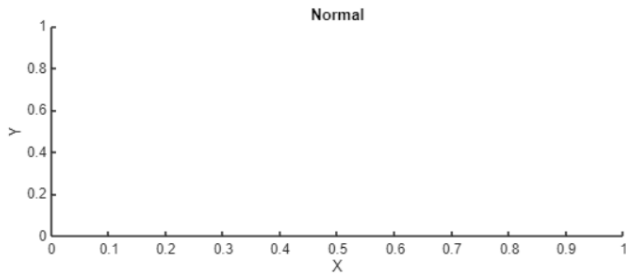
Las acciones disponibles en el sistema es la suma de las señales, donde se suman cada una de las señales marcadas previamente por el usuario. Además, se agrega el ruido gaussiano, donde afecta a todas las señales seleccionadas, esta opción conlleva un proceso de aleatoriedad, por lo que cada vez que se selecciona genera diferentes comportamientos en las señales

Acciones

☐ Suma de señales☐ Rudio

5. Espacio de trabajo:

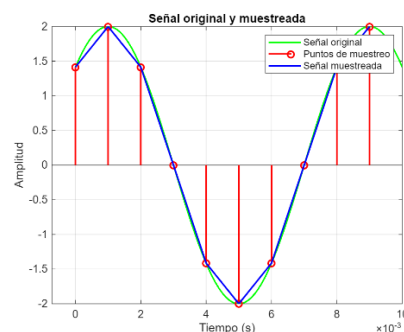
En el espacio de trabajo esta enfocado en la visualización de los resultados, una vez que se ingresaron los datos deseados y actualizados, ya se puede ver los resultados. Este espacio de trabajo cambia dinámicamente al seleccionar las diferentes funciones y acciones del sistema.



III. PRUEBAS

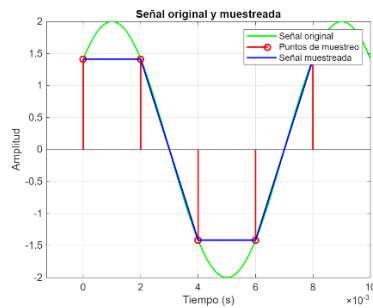
III.1. PARTE 1: MUESTREO

```
% Prueba 1
% Parámetros de muestreo
fs = 1000; % Frecuencia de muestreo (Hz)
tiempo = 10; % Tiempo total de grabación (milisegundos)
```



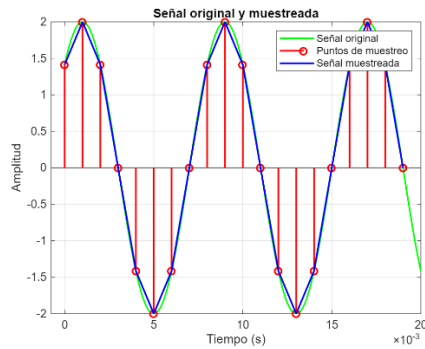
Prueba 1

```
% Prueba 2
% Parámetros de muestreo
fs = 500; % Frecuencia de muestreo (Hz)
tiempo = 10; % Tiempo total de grabación (milisegundos)
```



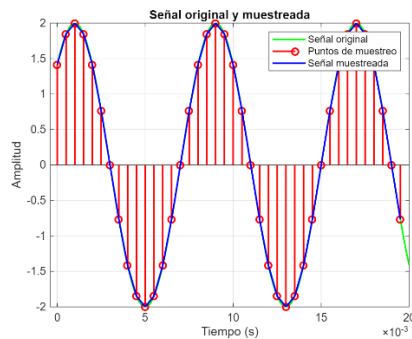
Prueba 2

```
% Prueba 3
% Parámetros de muestreo
fs = 1000; % Frecuencia de muestreo (Hz)
tiempo = 20; % Tiempo total de grabación (milisegundos)
```



Prueba 3

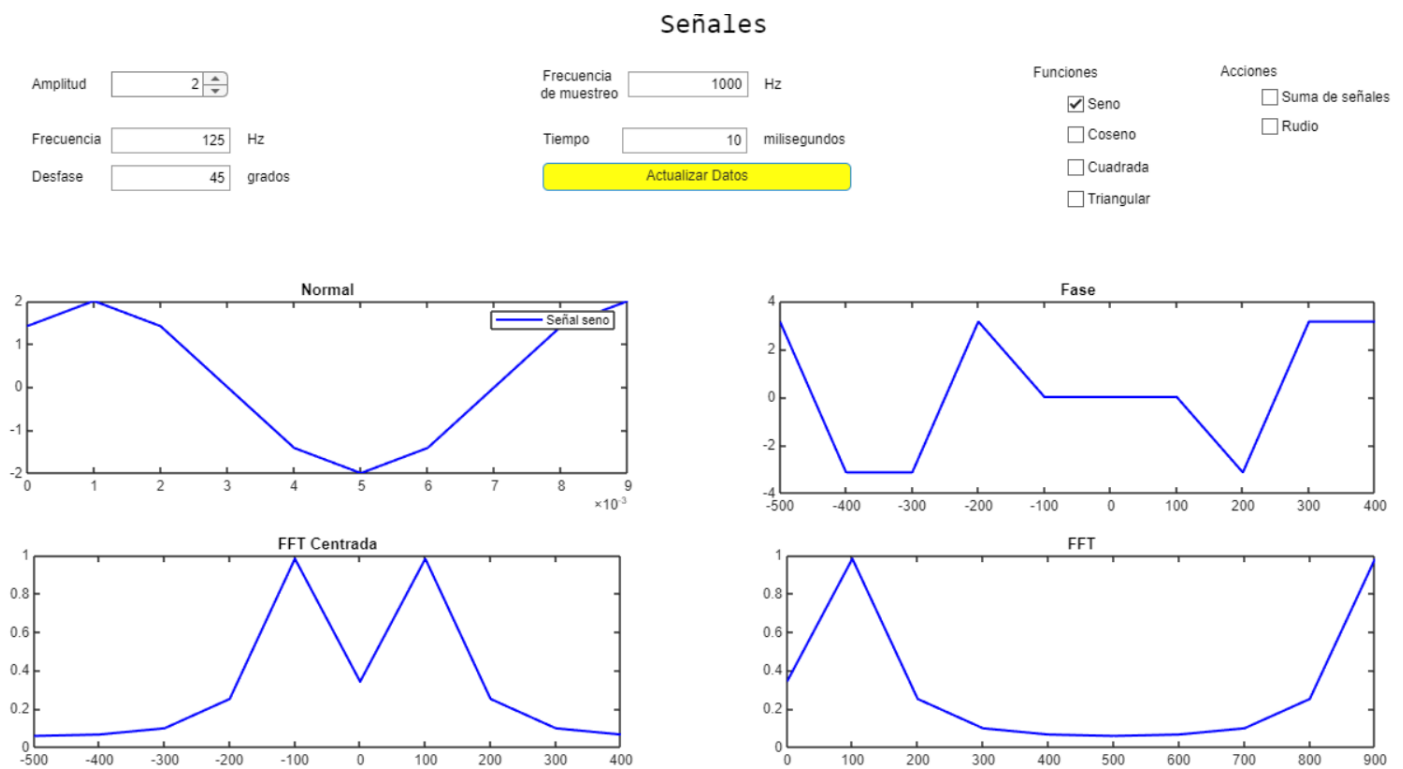
```
% Prueba 4
% Parámetros de muestreo
fs = 2000; % Frecuencia de muestreo (Hz)
tiempo = 20; % Tiempo total de grabación (milisegundos)
```



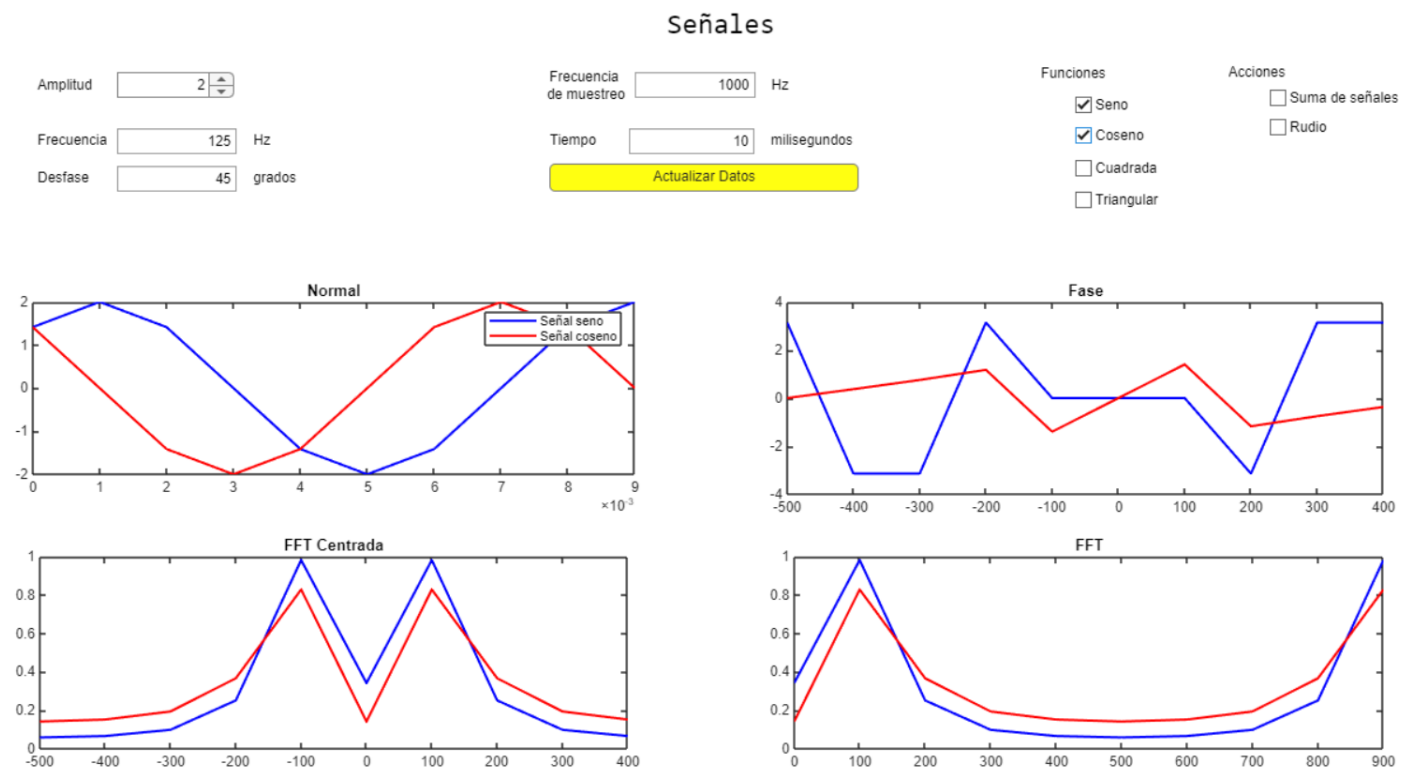
Prueba 4

III.2. PARTE 2: PRUEBAS DE LA INTERFAZ

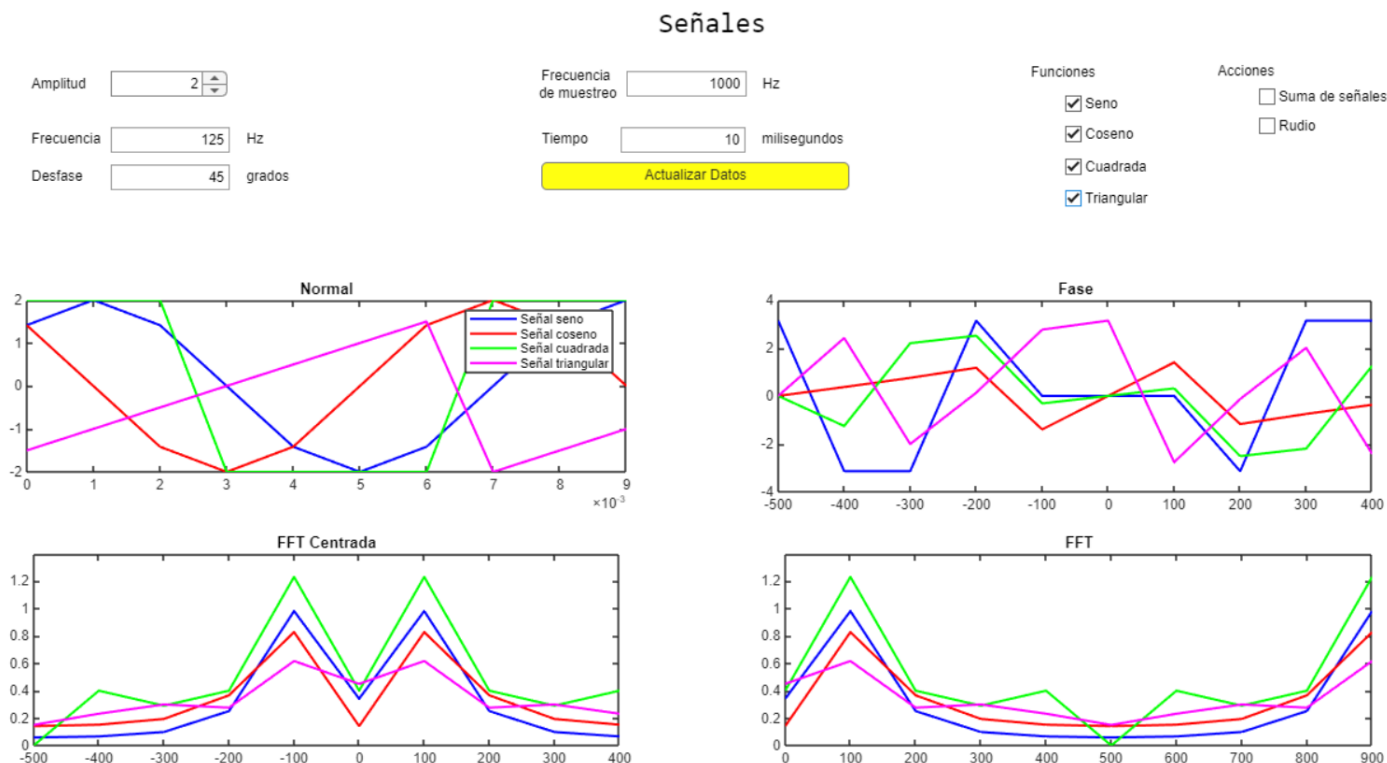
Utilizando los datos por defecto seleccionado en el sistema se pueden visualizar la función seno:



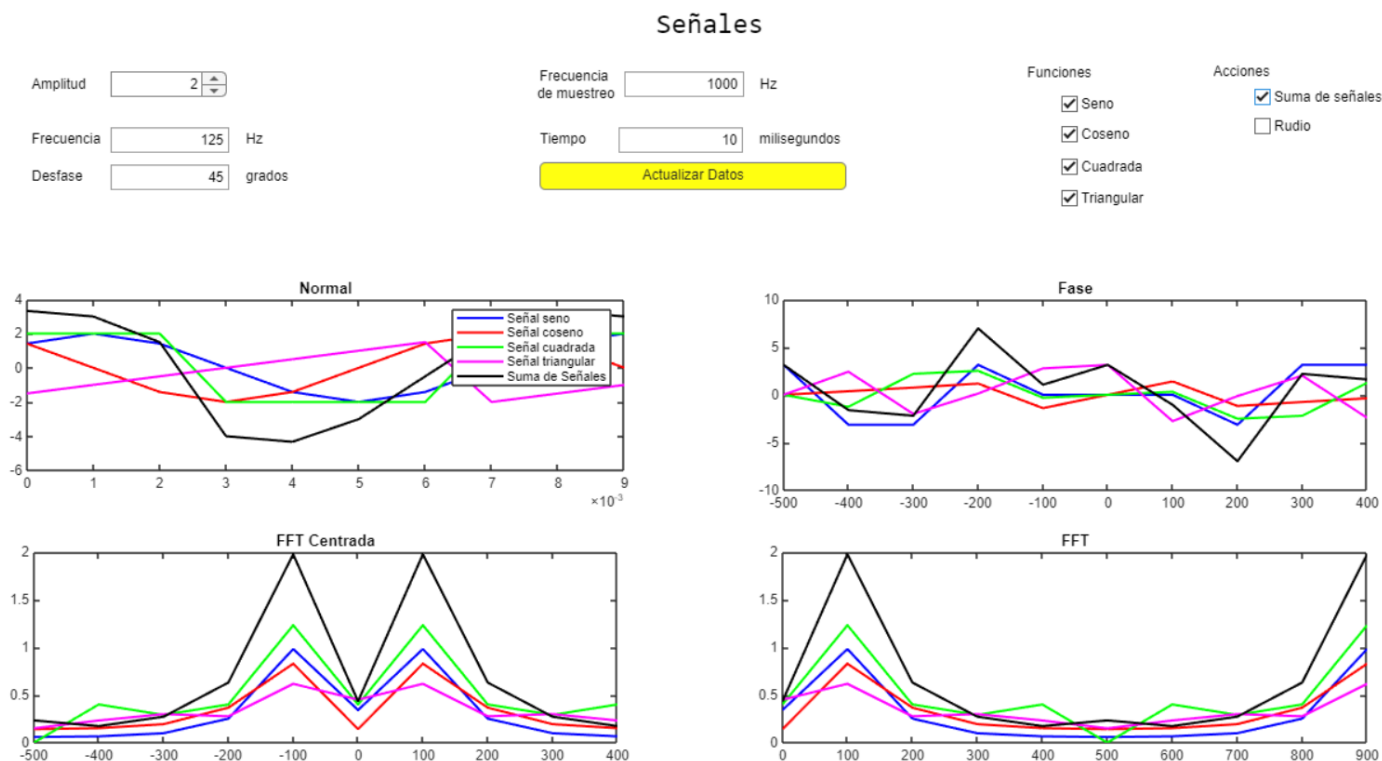
Dinámicamente podemos agregar más funciones como el coseno:



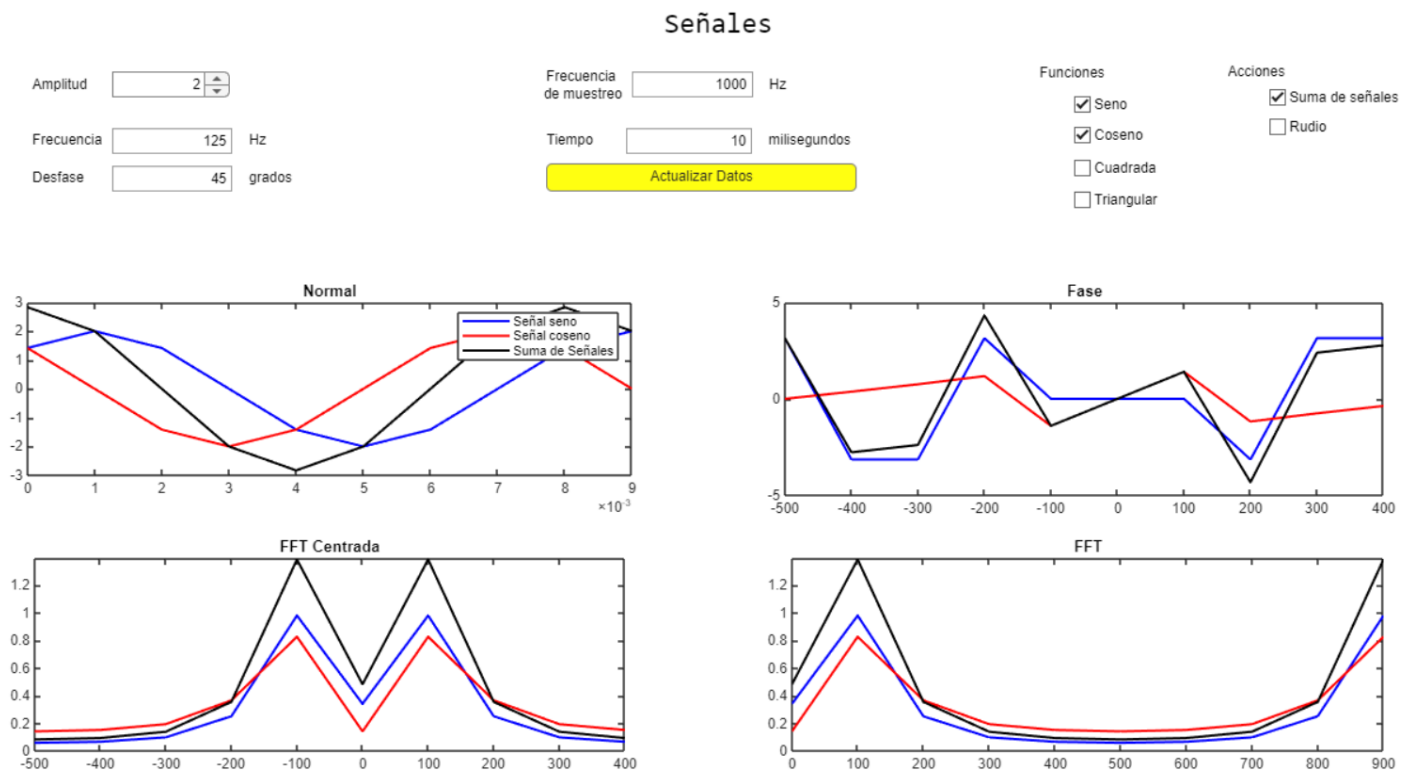
Se puede observar cómo se colocan en las mismas gráficas, esto ayuda a comparar directamente el comportamiento de las señales y ver las diferencias en cada espacio de trabajo. Además, observamos la leyenda que nos indica de qué color le pertenece a cada una. También podemos agregar todas las funciones:



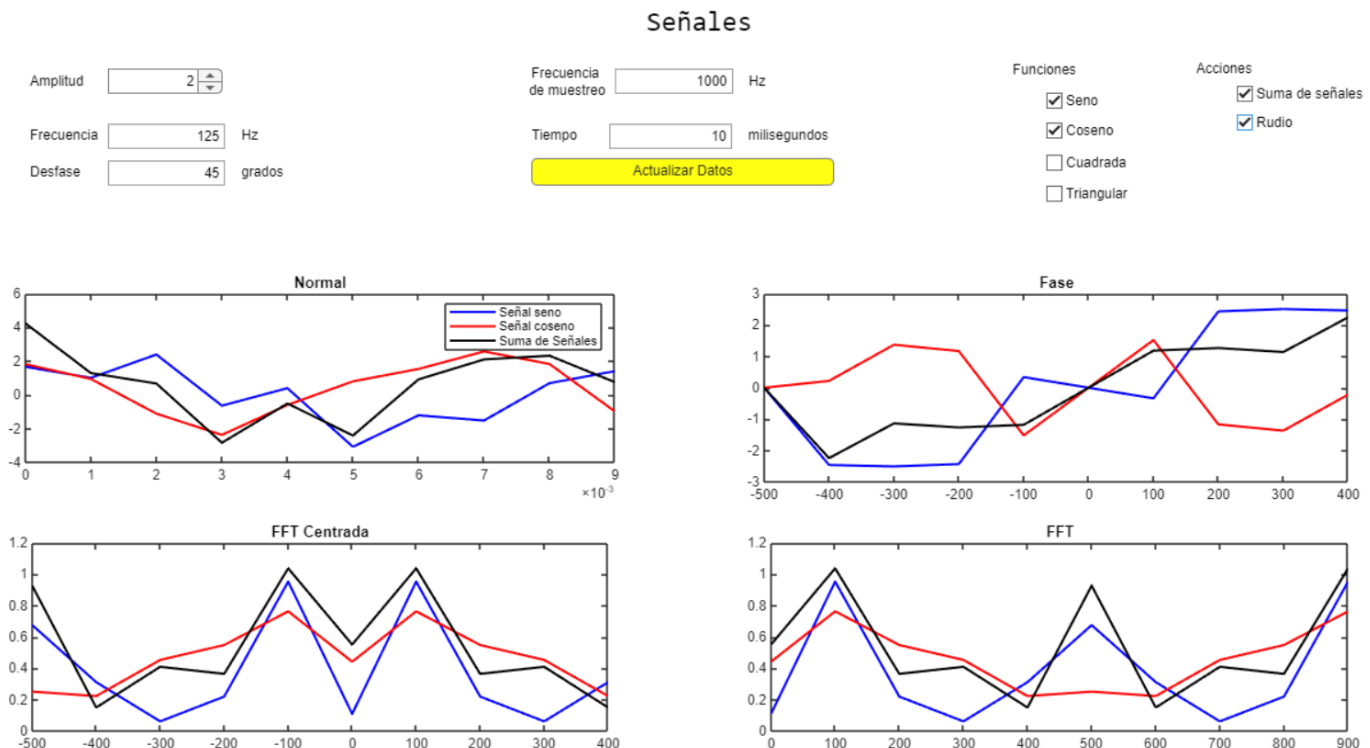
Podemos agregar la parte de la “suma de señales”



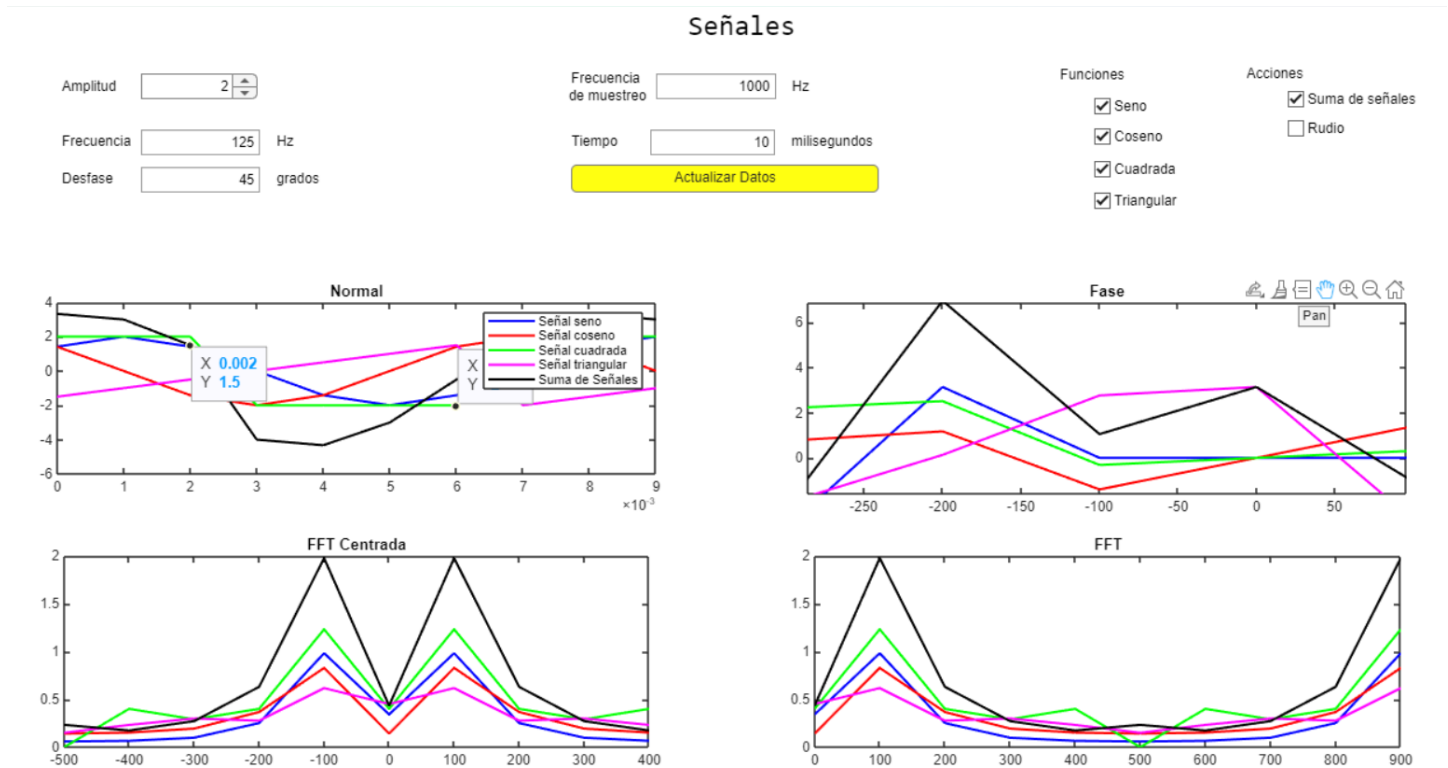
Donde se muestra con la línea negra, esta suma cambia si desmarcamos o marcamos las funciones, por ejemplo:



Aquí observamos el comportamiento con únicamente la señales seno y coseno. Además, a estas se le puede aplicar el ruido gaussiano:



Como se mencionó previamente, este proceso de ruido es aleatorio entonces no veremos el mismo comportamiento de nuevo. Además con las graficas se puede interactuar, es decir, podemos ver los puntos y hacer las acciones normales de un plot



IV. CONCLUSIONES

En esta práctica, se implementó el proceso de muestreo de una señal, con el objetivo de comprender cómo una señal continua puede ser representada de manera discreta. Al utilizar una frecuencia de muestreo alta, se capturaron los detalles de la señal original sin perder información significativa. El muestreo permite digitalizar señales analógicas y prepararlas para su procesamiento posterior. Además, se exploró cómo los parámetros de desplazamiento de fase y amplitud afectan la forma de la señal.

La interfaz desarrollada complementa este proceso al proporcionar un flujo de trabajo sencillo y estructurado. Los usuarios ingresan parámetros clave como la amplitud, frecuencia, desfase, y tiempo de grabación, y pueden ver de forma inmediata cómo estos afectan a las señales seleccionadas (seno, coseno, cuadrada, triangular). La opción de actualizar los datos dinámicamente permite un análisis interactivo, visualizando en tiempo real las modificaciones en la forma de la señal, su fase, su representación en el dominio de la frecuencia (FFT), y la versión centrada de la FFT.

Entre las acciones disponibles, el sistema permite sumar múltiples señales seleccionadas y añadir ruido gaussiano, lo cual agrega un factor de aleatoriedad al comportamiento de las señales. El espacio de trabajo interactivo se ajusta dinámicamente para reflejar los resultados de las señales procesadas, facilitando así la visualización y análisis de las transformaciones aplicadas.

V. BIBLIOGRAFIA

- [1] Libretexts. (2022, 1 noviembre). 1.2: Formas de onda sinusoidales. LibreTexts Español. [https://espanol.libretexts.org/Ingenieria/Libro%3A_An%C3%A1lisis_de_circuitos_el%C3%A9ctricos_de_CA%3A_un_enfoque_pr%C3%A1ctico_\(Fiore\)/01%3A_Fundamentos/1.2%3A_Formas_de_onda_sinusoidales](https://espanol.libretexts.org/Ingenieria/Libro%3A_An%C3%A1lisis_de_circuitos_el%C3%A9ctricos_de_CA%3A_un_enfoque_pr%C3%A1ctico_(Fiore)/01%3A_Fundamentos/1.2%3A_Formas_de_onda_sinusoidales)
- [2] Adquirir una señal analógica: ancho de banda, teorema de muestreo de Nyquist y aliasing. (2006, 31 agosto). NI. <https://www.ni.com/es/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>
- [3] SÁEZ, D. (2021, 19 abril). ¿Qué es frecuencia de muestreo y profundidad de bits? Hoy Grabo. <https://hoygrabo.com/que-es-frecuencia-de-muestreo-y-profundidad-de-bits/>
- [4] Cervantes, M. H. (n.d.). *Señales*. https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2743/mod_resource/content/1/UAPA-Senales/index.html
- [5] *Tipos de señales eléctricas y electrónicas*. (2023, March 21). MICROCHIPOTLE. <https://microchipotle.com/tipos-de-senales-electricas-y-electronicas/>