



INSTITUTO TECNOLÓGICO DE COSTA  
RICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA

PROCESAMIENTO DIGITAL DE SEÑALES

---

## Tarea 1

---

*Students:*

Malcolm DAVIS  
Daniel LEON

*Professor:*

Jose BARBOZA

26 de febrero de 2017

Cuadro 1: Marcadores

Marcador	Descripcion
o	Circulo letra o
+	Signo más
*	Asterisco
.	Punto
x	Cruz equis x
d	Diamante
s	Cuadrado
p	Estrella de cinco picos
v	Triangulo abajo
^	Triangulo derecha
i	Triangulo izquierda

Cuadro 2: Estilo de Linea

Estilo de Linea	Descripcion
'_'	_____
'--'	-----
'.'	.....

## 1. Problema 1

### 1.1. `plot(x1,y1,x2,y2, 'Opciones')`

Crea un gráfico en 2D de los datos de  $x_N$  versus los de  $y_N$  de forma continua.

- X y Y deben ser vectores de la misma longitud.
- Opciones se refiere a los cambios de estilo de línea, marcador y color del gráfico.

Ver cuadros del 1 al 3.

### 1.2. `stem(x1,y1,x2,y2, 'Opciones')`

Crea un gráfico en 2D de los datos de  $x_N$  versus los de  $y_N$  de forma discreta o “muestreada”.

Cuadro 3: Color

Color	Descripcion
'y'	Amarillo
'm'	Magenta
'c'	Cyan
'r'	Rojo
'g'	Verde
'b'	Azul
'w'	Blanco
'k'	Negro

### 1.3. Hold on, hold off

Estos son comandos para que, a la hora de ejecutar dos o más funciones de gráficas, sea plot, stem u otra, se sobreponga la gráfica 2 en antecesora la gráfica 1.

### 1.4. figure('PropertyName',propertyvalue,...)

Crea una nueva ventana de tipo figura usando valores de propiedades específicas.

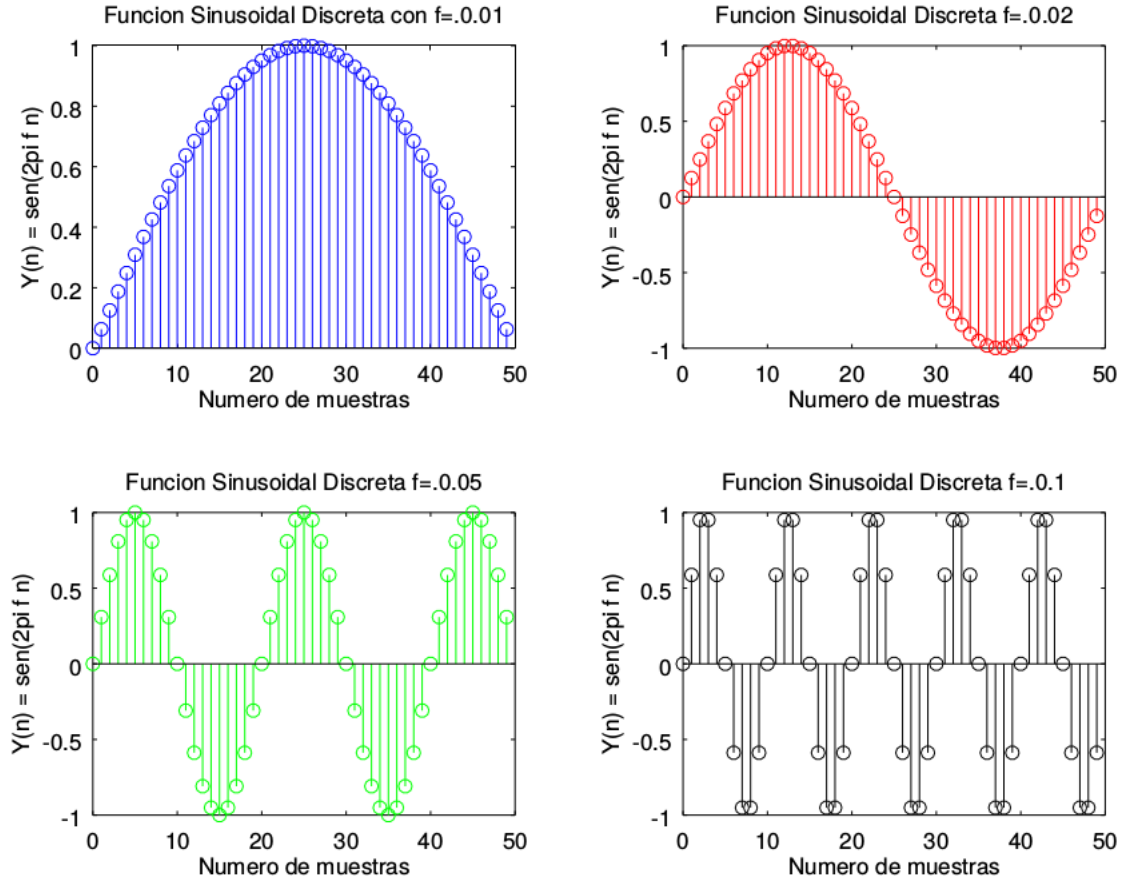
## 2. Problema 2

Para realizar este ejercicio se crea un vector de frecuencia para los 4 valores requeridos, luego se instancia una variable n para poder almacenar un vector con las primeras 50 muestras, y se almacena una funcion de seno con cada una de las frecuencias dadas con las muestras requeridas. Luego se decide graficarlas con la función **stem** que a diferencia de **plot** crea una grafica con valores puntuales y no continuos. Ver figura 1.

## 3. Problema 3

Para la solución de este ejercicio se elige declarar los vectores de frecuencia f (cuatro valores fijos,  $f = [0.01, 0.02, 0.05, 0.1]$ ) y de la variable discreta n ( $n = 0, 1, 2, 3, \dots$ ) y en este último vector es donde el usuario elige el número

Figura 1: Diferentes frecuencias la funcion seno con 50 muestras.

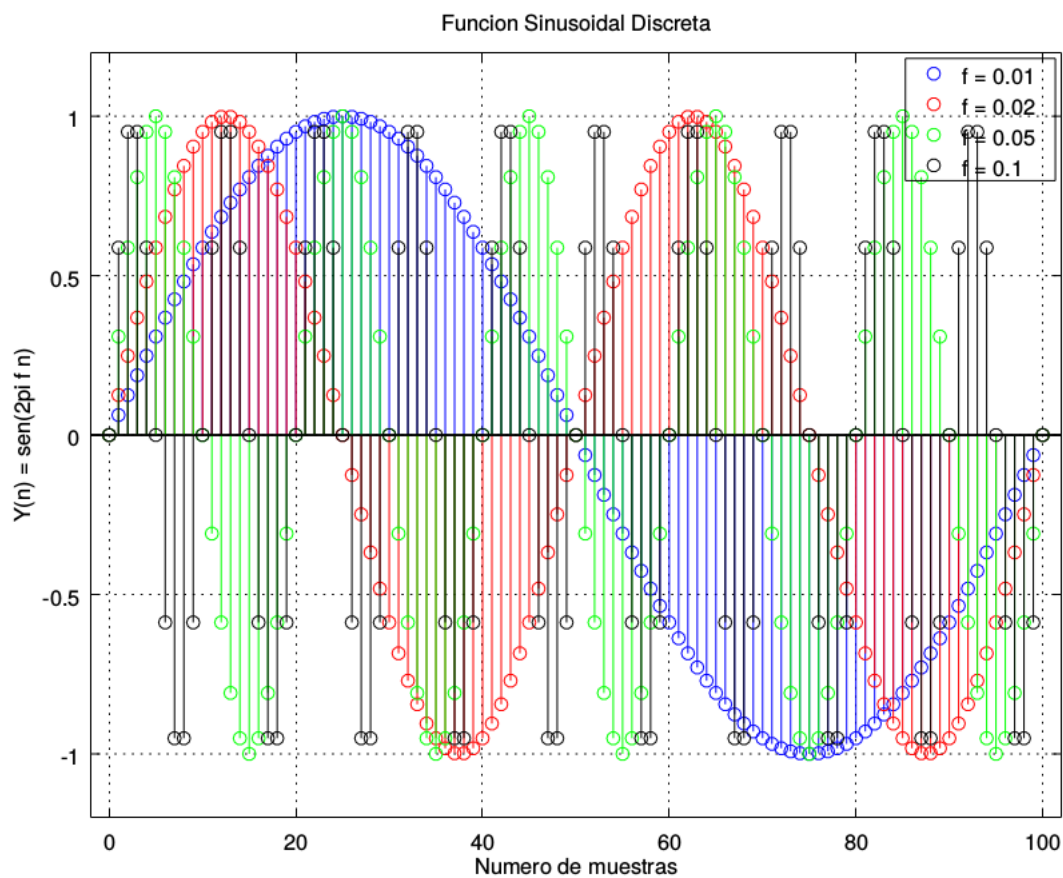


máximo de muestras  $N$ . El vector queda de la forma  $n = [0 : N]$ . El resto es solo implementar la función  $Y_p$ ; y graficar con el comando `stem(n,Yp)`. También se agregaron algunos detalles como nombre del gráfico y sus ejes, así como la leyenda. Con una función `verParcial(100)` tenemos(ver figura 2):

## 4. Problema 4

Para resolver este ejercicio, se encuentra el siguiente alias de la frecuencia 0.05 al sumar su versión angular con  $2\pi$  ya que  $\text{alias} = (w1+2\pi)n$  y con  $n = 1$ ,  $\text{alias} = w1+2\pi$ . Se puede demostrar gráficamente que esto es verdadero

Figura 2: Resultado de verParcial(100).



y que ambas funciones coinciden. Ver figura 3.

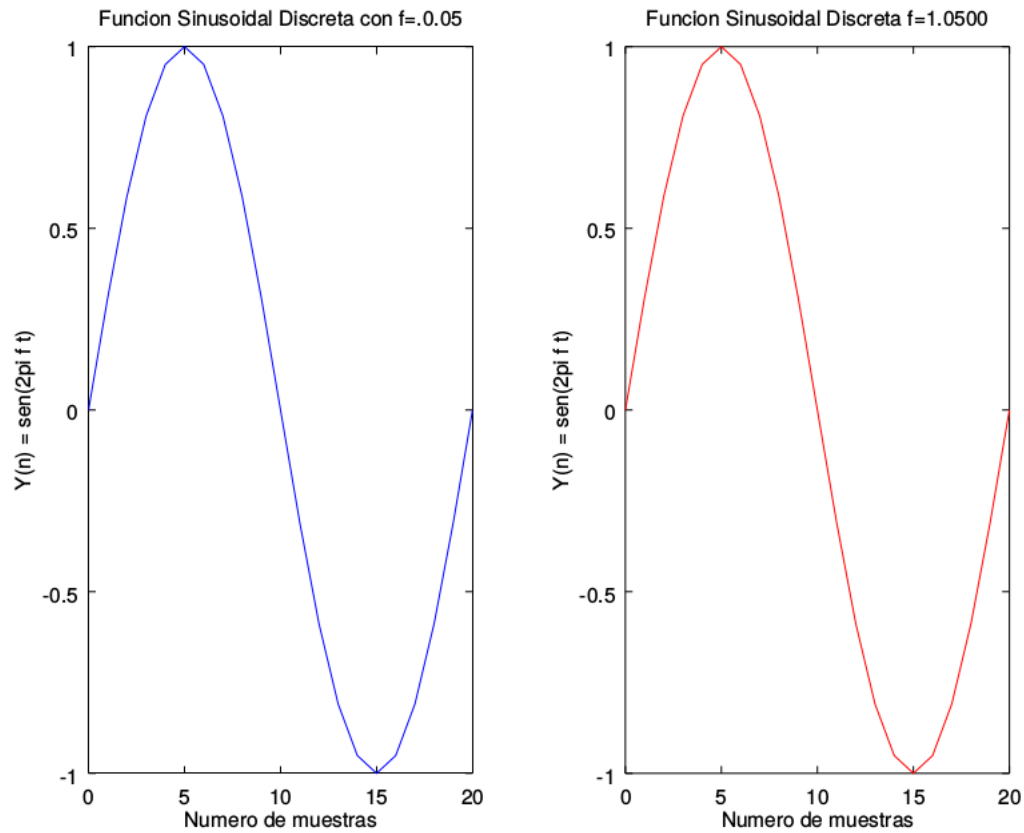
## 5. Problema 5

### 5.1. sound

sound (y)  
 sound (y, fs)  
 sound (y, fs, nbits)

Reproduce un audio “y” a una frecuencia de muestro “fs”. Si esta última no se especifica se le asigna 8000 muestras por segundo por defecto. Y “nbits”

Figura 3: Alias.



es el número de bit que se desea utilizar.  
`soundsc (y, fs, nbits [ymin, ymax])`

## 5.2. wavread

`y=wavread(filename)`  
`[y,fs,nbits] =wavread(filename)`  
Lee una señal de audio con el nombre de archivo “filename”, y la guarda como vector en y. Si se agrega fs retorna la frecuencia de muestreo de la señal, y nbits los bits utilizados.

### 5.3. wavwrite

wavwrite (y, fs, nbits, filename)

Escribe un archivo de audio con el nombre “filename” que lo lee desde el vector y, y se le puede especificar la frecuencia de muestreo fs, y la cantidad de bits a utilizar.

## 6. Problema 6

soundsc(y, fs, filename)

Con el comando **soundsc**, se reproduce un archivo y su parametro fs permite modificar la frecuencia de muestreo.

## 7. Problema 7

Solo se define las variables, de la frecuencia de 440Hz, frecuencia de muestreo (utilizando el parámetro de entrada N), el tiempo de duración del tono, y la función sinusoidal. Seguido de esto se utiliza wavwrite() y soundsc(), la primera para generar el archivo .wav con el tono generado por la función seno, y el segundo para hacer sonar el tono. Ver figura 4.

## 8. Problema 8

Primero que todo, se define el periodo de la señal a muestrear (señal continua), después el número de muestras a tomar, para esto se utiliza el parámetro de entrada tp (periodo de muestreo) y se le quitan los decimales para generar un numero entero. Luego están los vectores de t y n, variables de las funciones a crear. La primera función se define de la manera  $\sin(2\pi \cdot f \cdot t)$ , donde en ella se modela una señal en tiempo continuo, la segunda funcion se define  $\sin(2\pi \cdot f \cdot t_s \cdot n)$ , donde f es la frecuencia,  $t_s$  periodo de muestreo, n número de muestra y t variable temporal. Por último el alias se define  $\sin(2\pi \cdot f \cdot t_s \cdot n + 2k\pi)$ .

Para graficar las funciones discretas se debe escalar la variable n a  $n \cdot t_s$ .

Función sobrePos(2500,20e-6,4) esto es 2.5 kHz, 20  $\mu$ s,  $k = 4$ . Ver figura 5.

Función sobrePos(100,20e-5,10) esto es 100 Hz, 200  $\mu$ s,  $k = 10$ . Ver figura 6.

Figura 4: Wavsound.

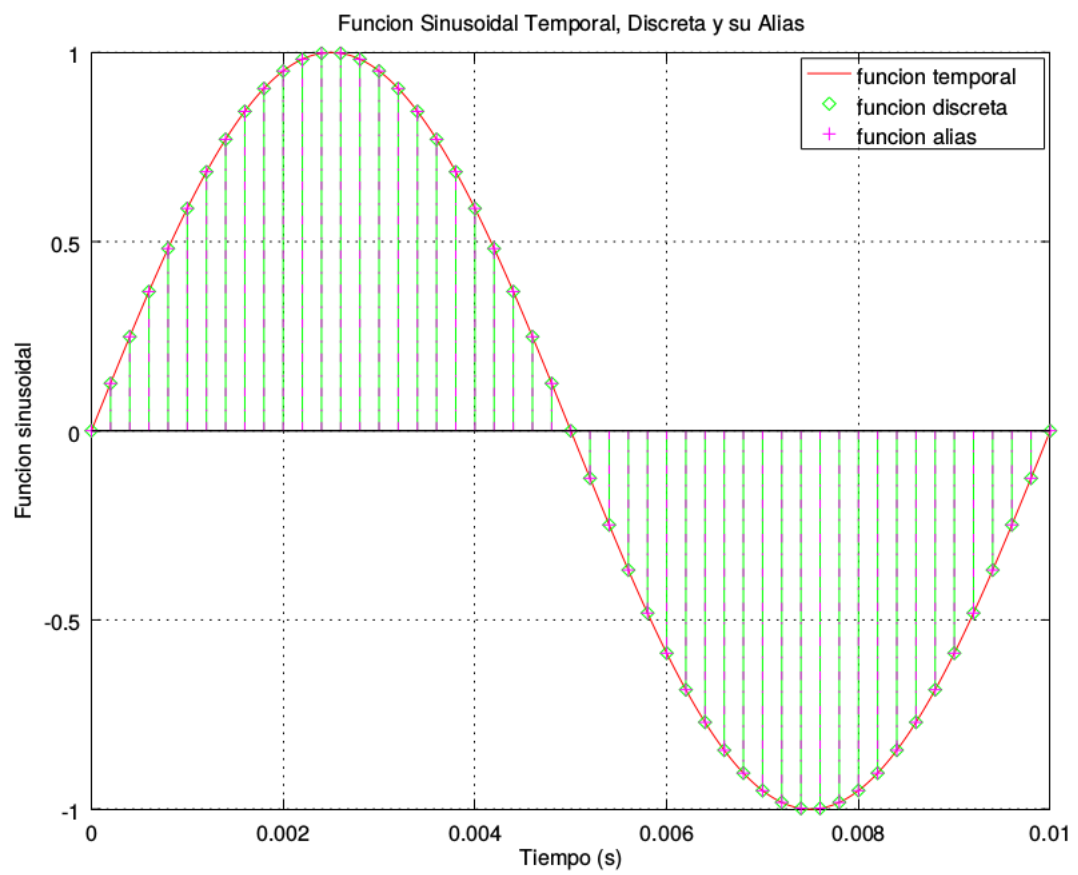




Figura 5: sobrePos(2500,20e-6,4).

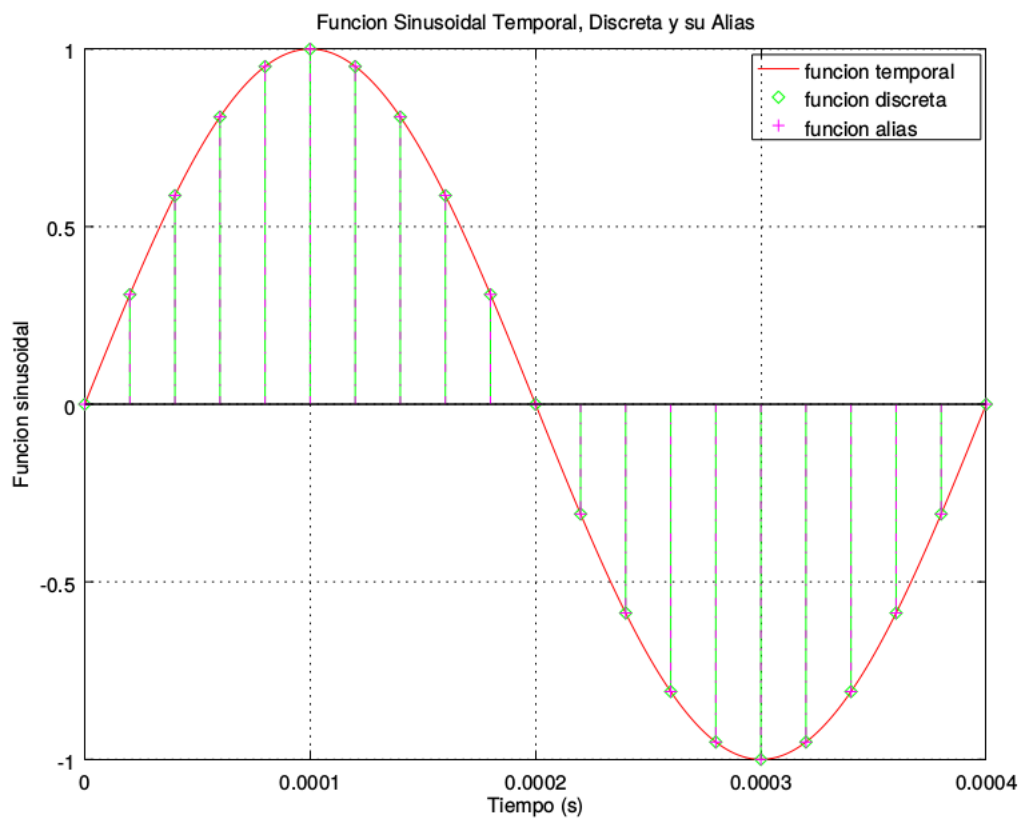


Figura 6: sobrePos(100,20e-5,10).

