



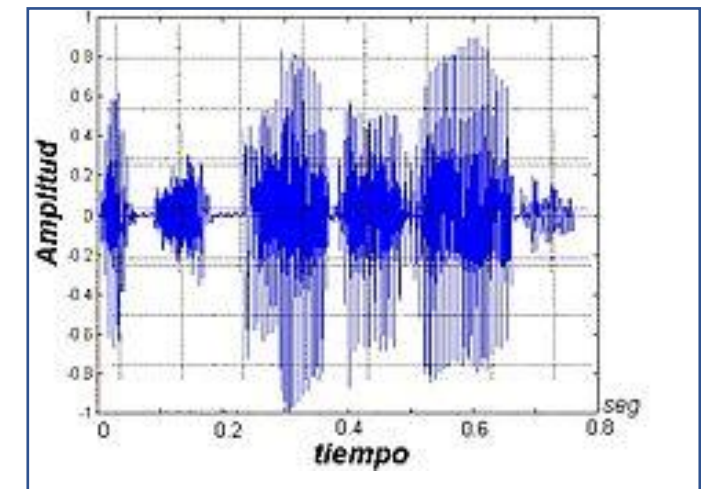
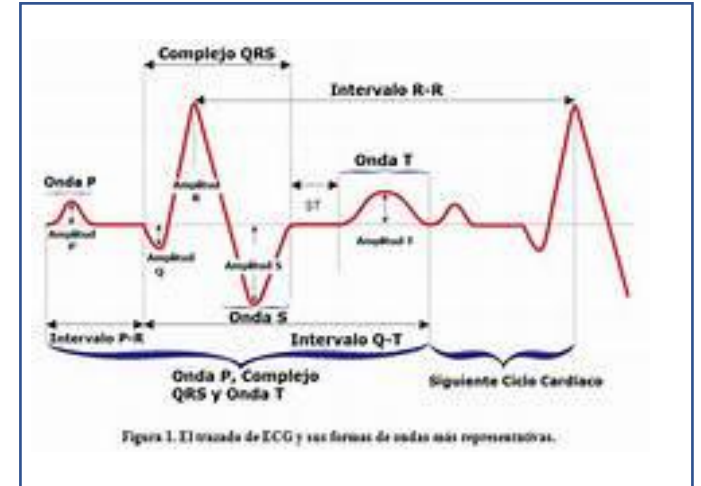
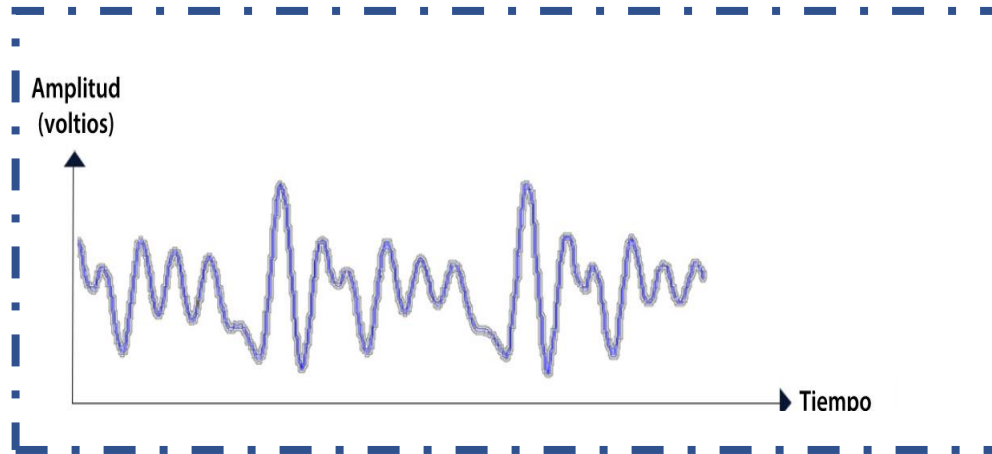
CLASE 1

PROCESAMIENTO DIGITAL DE SEÑALES

BY: Jorge Miranda

SEÑAL

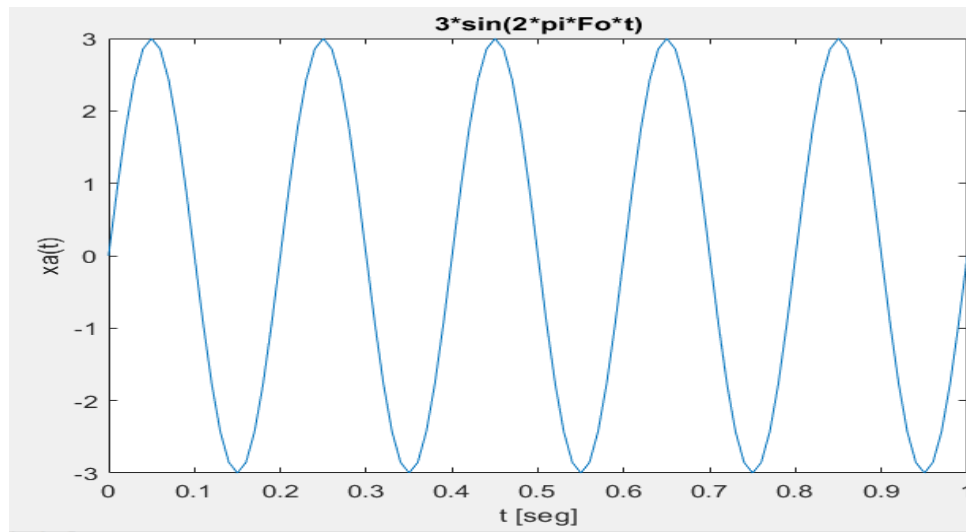
- Función de una o mas variables independientes que transportan algún tipo de información



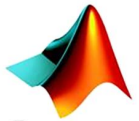
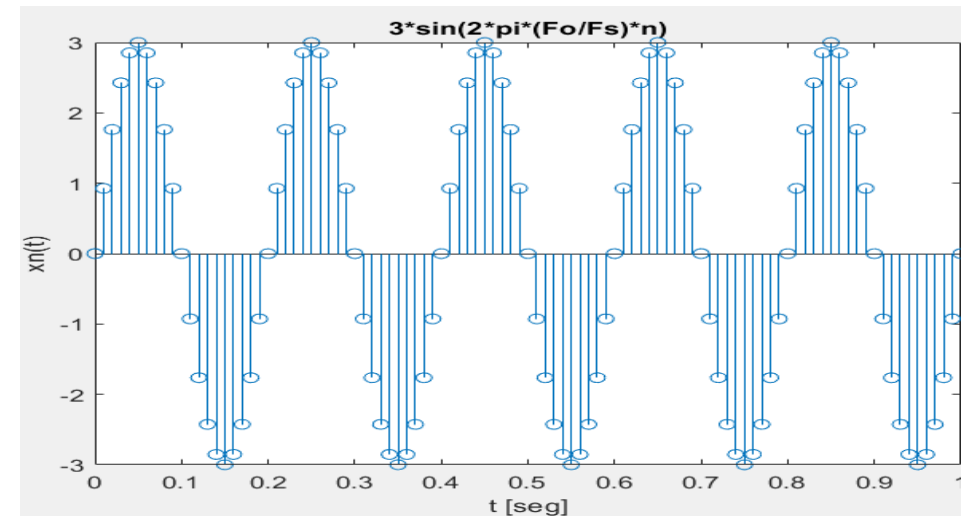
CLASIFICACIÓN DE LAS SEÑALES

Existen diversas clasificaciones de una señal , en el contexto en como se define el tiempo como variable independiente de la señal encontramos señal en tiempo continuo y señal en tiempo discreto.

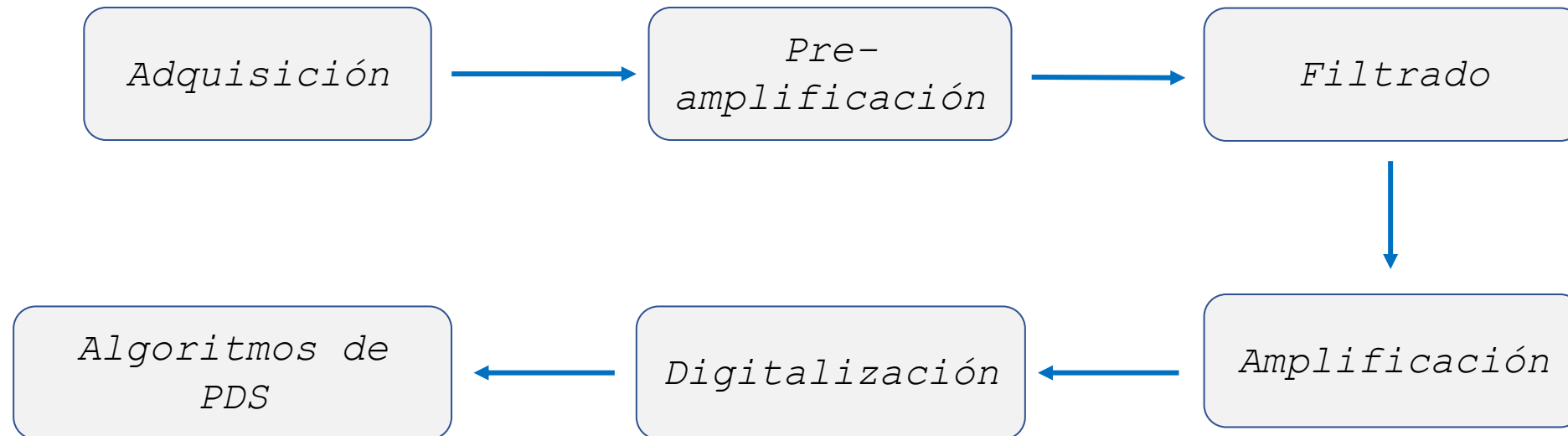
SEÑAL EN TIEMPO CONTINUA



SEÑAL EN TIEMPO DISCRETO



ADQUISICIÓN SEÑALES ANALÓGICAS



ADQUISICIÓN

Requiere del uso de transductores de señales como etapa previa del acondicionamiento de la señal



SEÑAL ECG, EOG: Electrodos

SEÑAL DE VOZ: Micrófono

Temperatura : Termopar

La amplitud de estas señales adquiridas se encuentren en el orden de unos pocos milivoltios (Necesitan ser acondicionados para que digitalmente puedan ser procesados).

ADQUISICIÓN SEÑAL EOG

Los **electrodos superficiales** obtienen o detectan la diferencia de potencial existente entre la **retina y la cornea**.

Detección del movimiento horizontal:

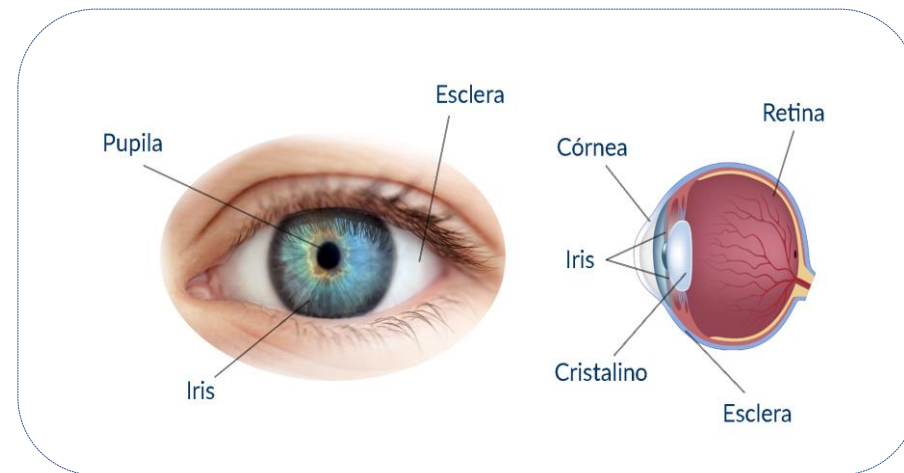
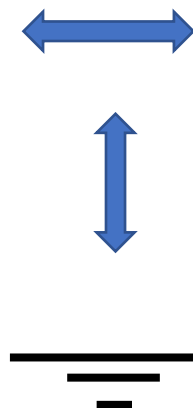
2 electrodos

Detección del movimiento vertical:

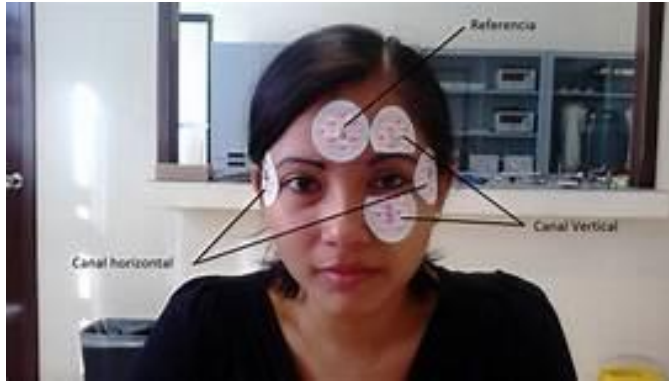
2 electrodos

Nivel de referencia :

1 electrodo

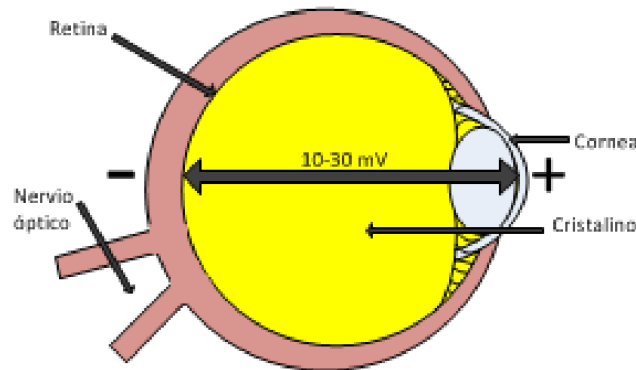


ADQUISICIÓN SEÑAL EOG

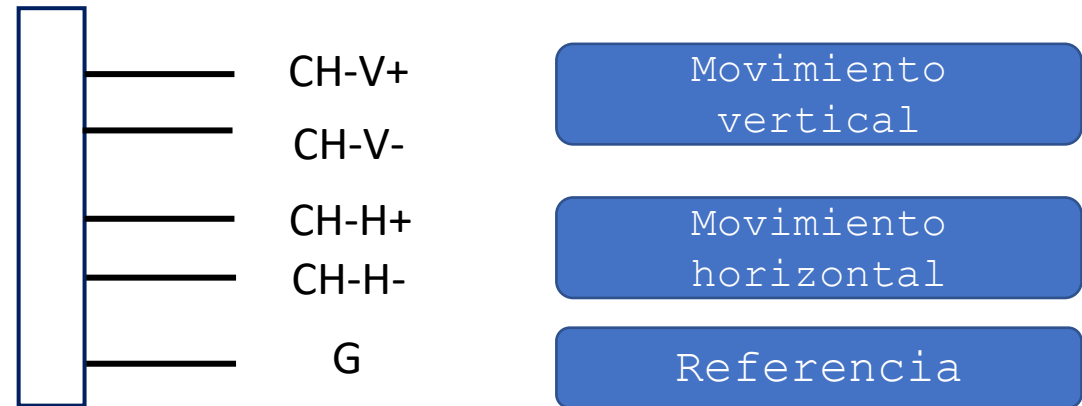
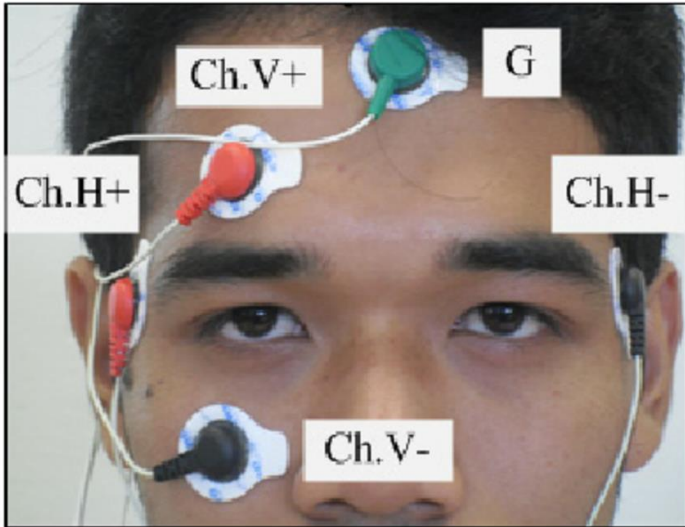


Electrooculograma:

Registro de los movimientos oculares basado en la medición de la diferencia de potencial existente entre la córnea y la retina.



ADQUISICIÓN SEÑAL EOG



Las señales EOG se encuentran comúnmente en el rango de amplitud de unos 10 μV a 300 μV

PRE-AMPLIFICACIÓN

La información del movimiento ocular en cualquier sentido se encuentra en la **diferencia de potencial** existente en cada par de electrodos

PRE-AMPLIFICACIÓN

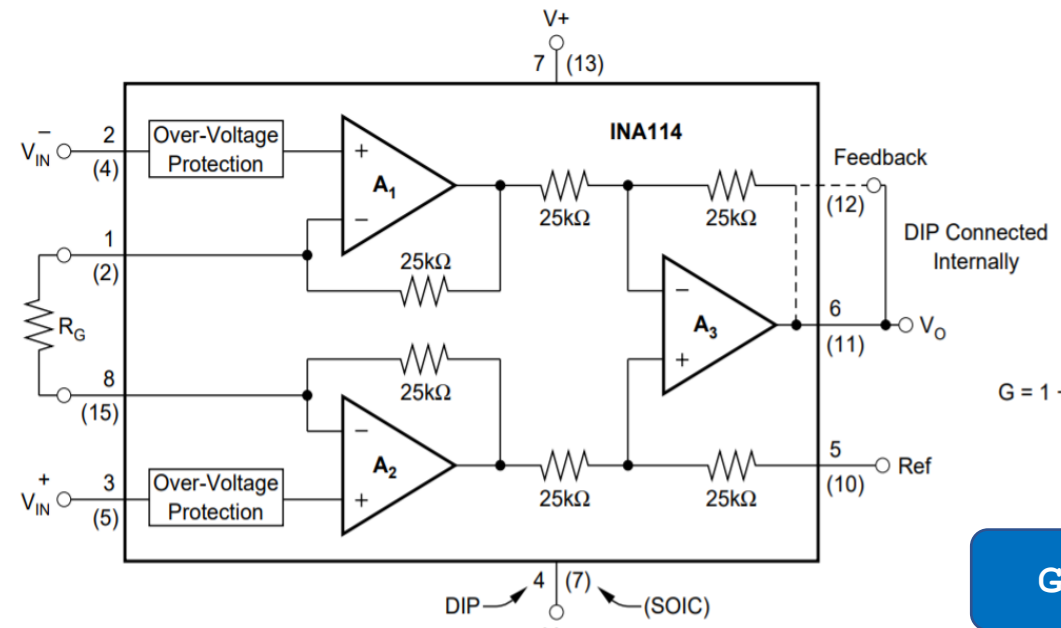
Consiste en la amplificación de la señal de información con un alto rechazo al **ruido común**

instrumentación ,debido al **alto CMRR** ,
alta impedancia de entrada, **baja impedancia de salida**.

PRE-AMPLIFICACIÓN

El amplificador de instrumentación es por excelencia lo más recomendable a la hora de adquirir señales diferenciales con baja amplitud .

INA114



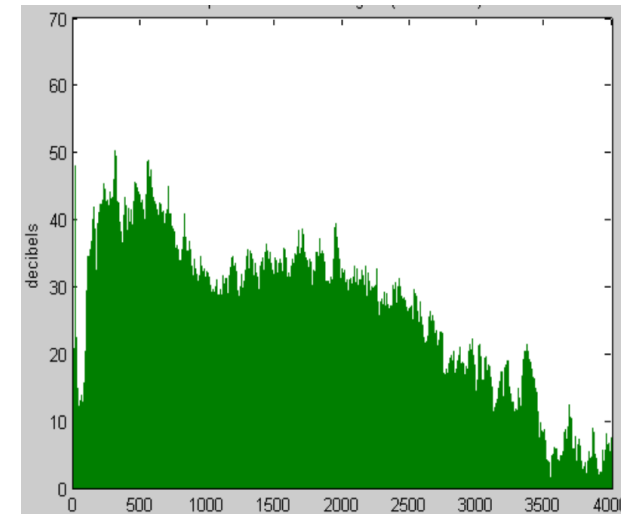
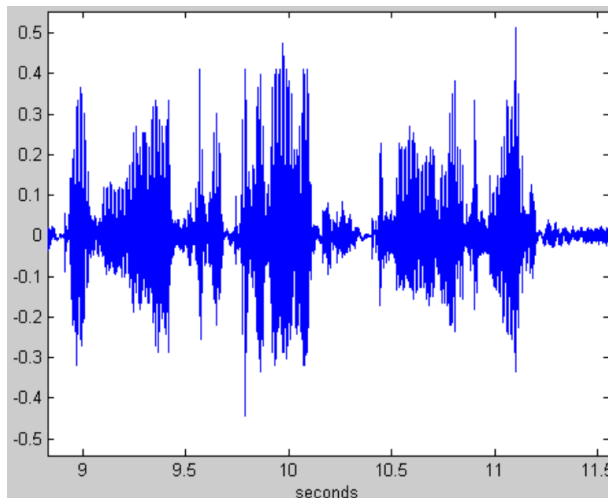
GANANCIA

SOIC: Encapsulado SMT
DIP: Encapsulado THT

FILTRADO

Las señales poseen una identificación única y que se pueda apreciar de la mejor manera en el dominio de la frecuencia.

De acuerdo al espectro de frecuencia de la señal de interés, se diseña un FILTRO que solo deje pasar la información de interés y rechazar otro.



FILTRADO

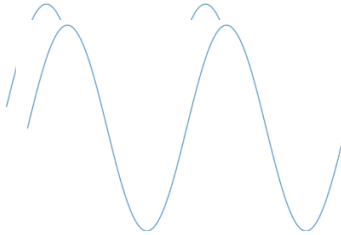
El filtro estará constituido de elementos pasivos (Resistencias, condensadores, bobinas) y activos (amplificadores operacionales).

El filtro es el encargado de dejar pasar solo las componentes de frecuencia asociada a la señal de interés y rechazar lo que no se corresponda con la señal.

Señal	Rango de Frecuencia
EOG	0-30 Hz
ECG	0-100 Hz
EMG	10-200 Hz
VOZ	100-4000 Hz

FILTRADO

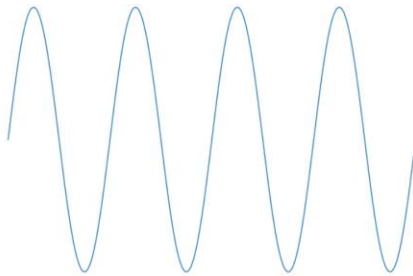
$F_1 = 10$



$$F_1 < F_2 < F_3$$

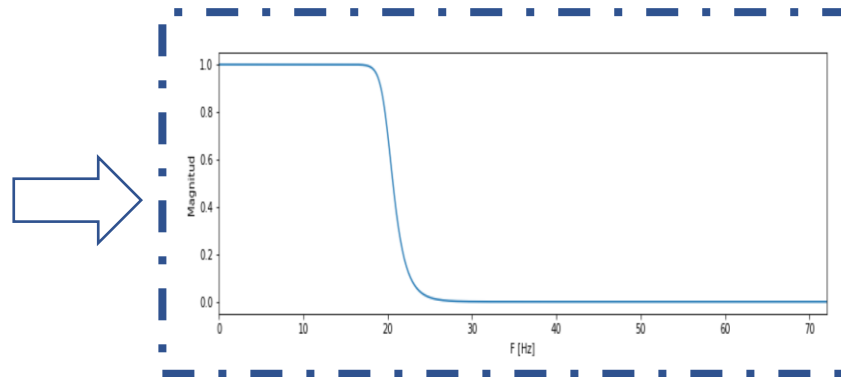
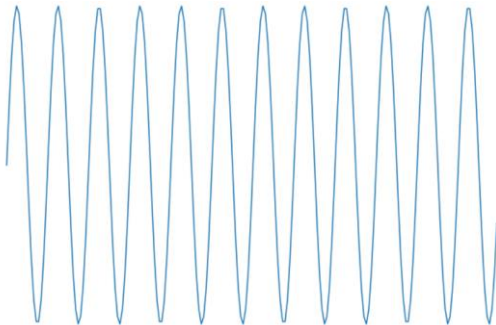
$F_2 = 25$

+

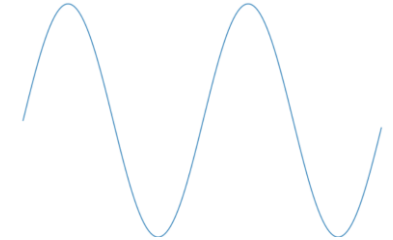


$F_3 = 60$

+



$F = 10$



AMPLIFICACIÓN

La señal pre-amplificada y filtrada ahora deberá ser amplificada nuevamente.

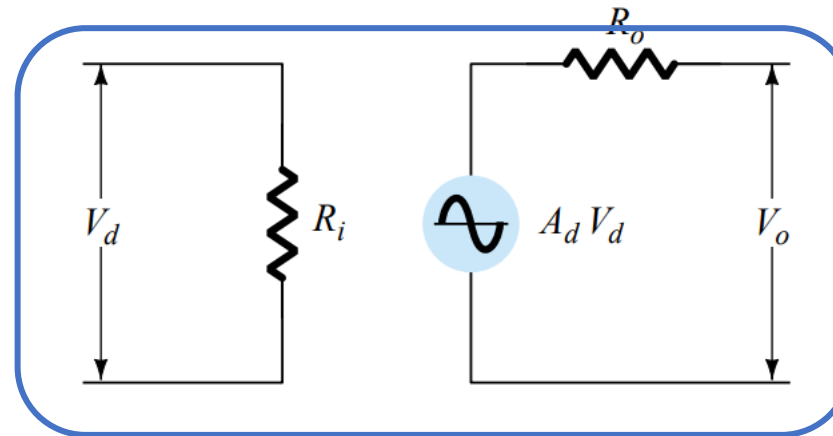
*Esta etapa se puede realizar por medio de amplificadores operacionales de **bajo costo** .*

Un ejemplo de opamp de bajo costo es el TL-082.

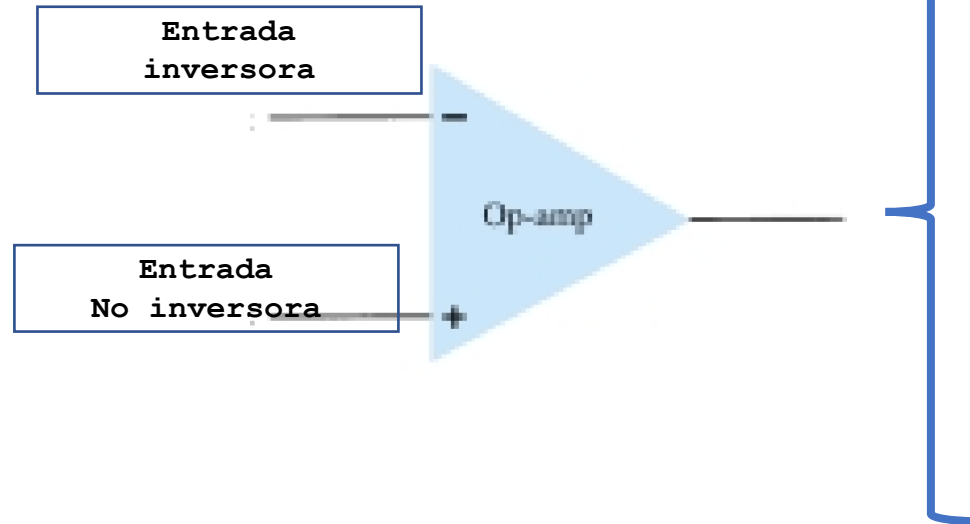
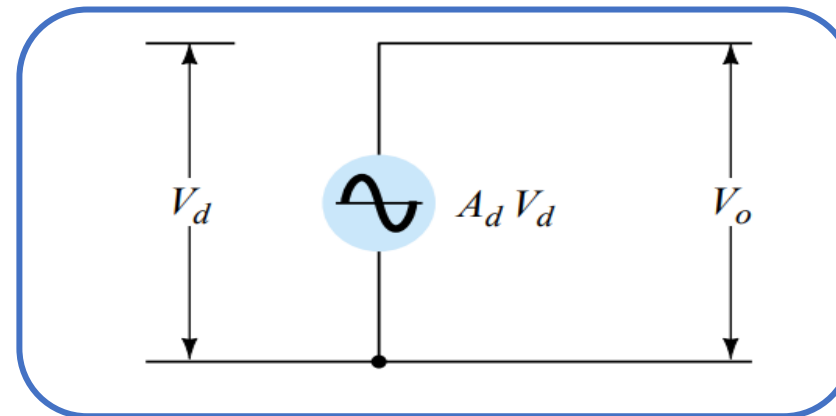
*La amplificación debe garantizar que la señal pueda ser digitalizada por un **ADC de manera eficiente***

AMPLIFICADOR OPERACIONAL

CIRCUITO EQUIVALENTE I

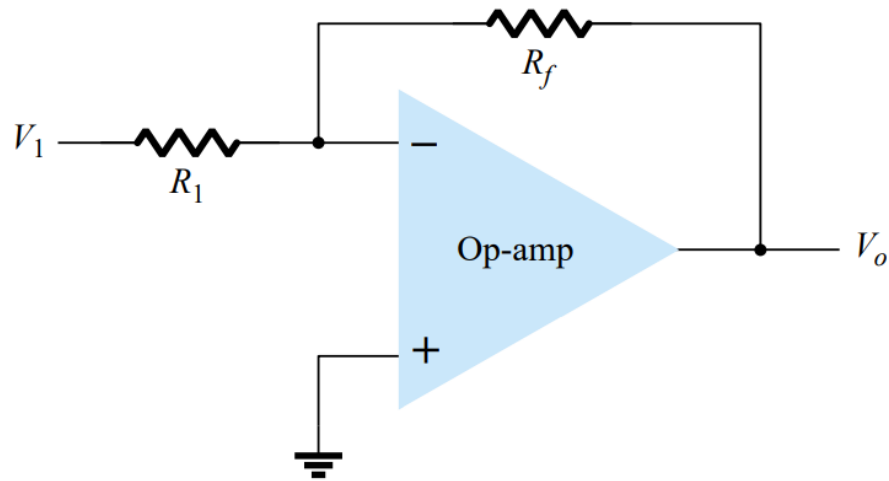


CIRCUITO EQUIVALENTE II

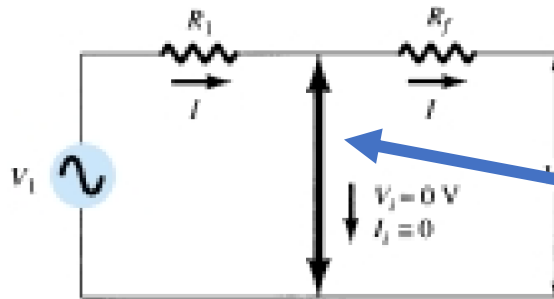


Negative Feedback

CIRCUITO AMPLIFICADOR INVERSOR



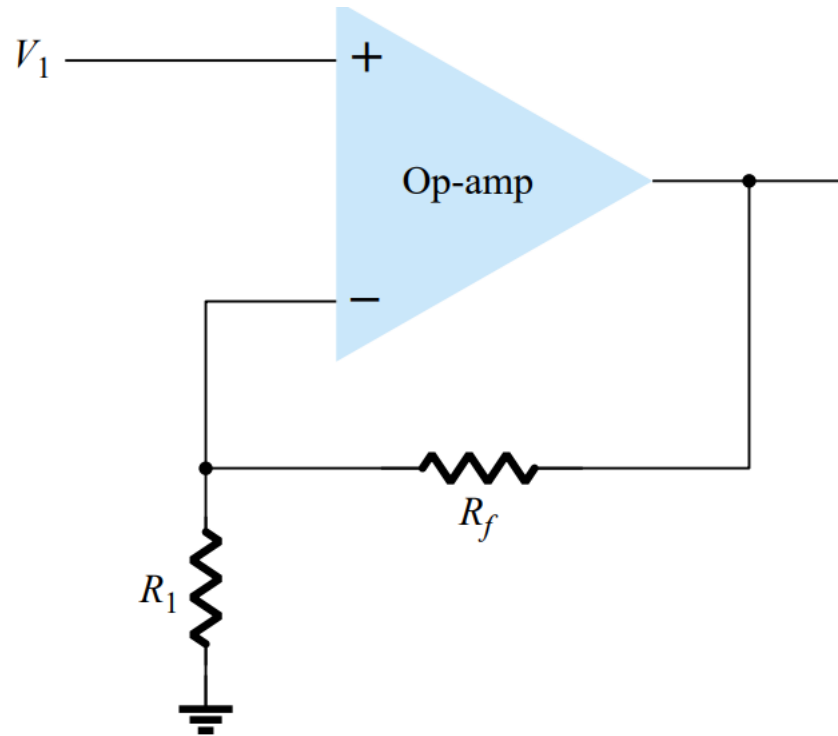
$$V_o = -\frac{R_f}{R_i} V_i$$



TIERRA VIRTUAL

Negative Feedback

CIRCUITO AMPLIFICADOR NO INVERSOR



$$V_o = \left(1 + \frac{R_f}{R_1}\right) V_i$$

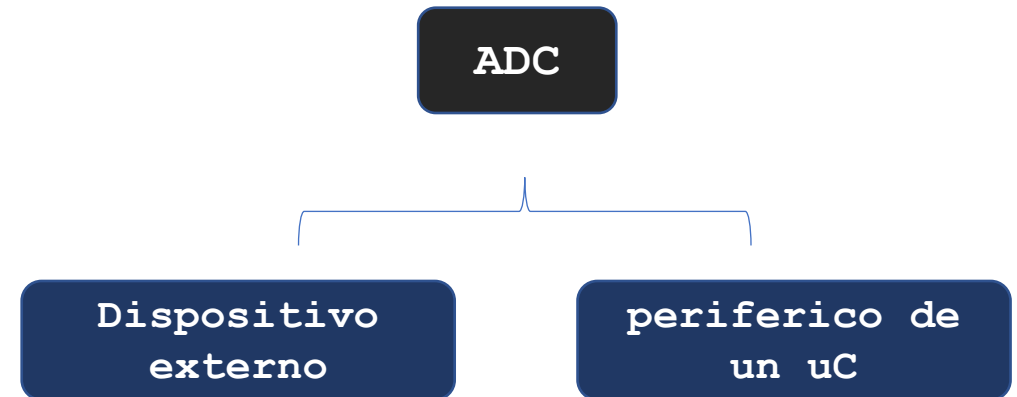
OPAMP

CARACTERISTICAS

A_d : Ganancia en lazo abierto
 Z_i : Impedancia de entrada
 Z_o : Impedancia de salida
CMRR; índice de rechazo de modo común
 P_{tot} : potencia disipada

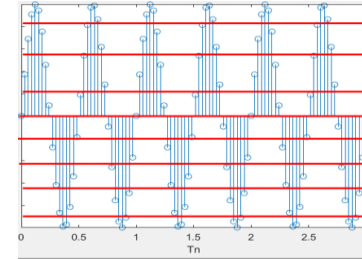
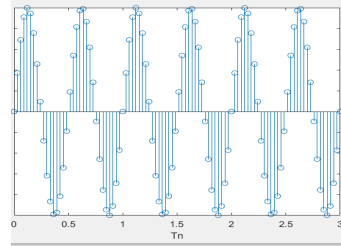
Los dispositivos ADC se encargan de muestrear, cuantificar y codificar la señal analógica.

*Se los encuentra como periférico de un uC o como dispositivo externo (bus de comunicación **SPI** o **I2C**)*



DIGITALIZACIÓN

ADC



BUFFER

0b011
0b010
0b001
0b000
0b111
0b110
0b001
0b100

MUESTREADOR

CUANTIFICADOR

CODIFICADOR

*Mantiene o congela el
valor de señal
durante un intervalo
de tiempo.*

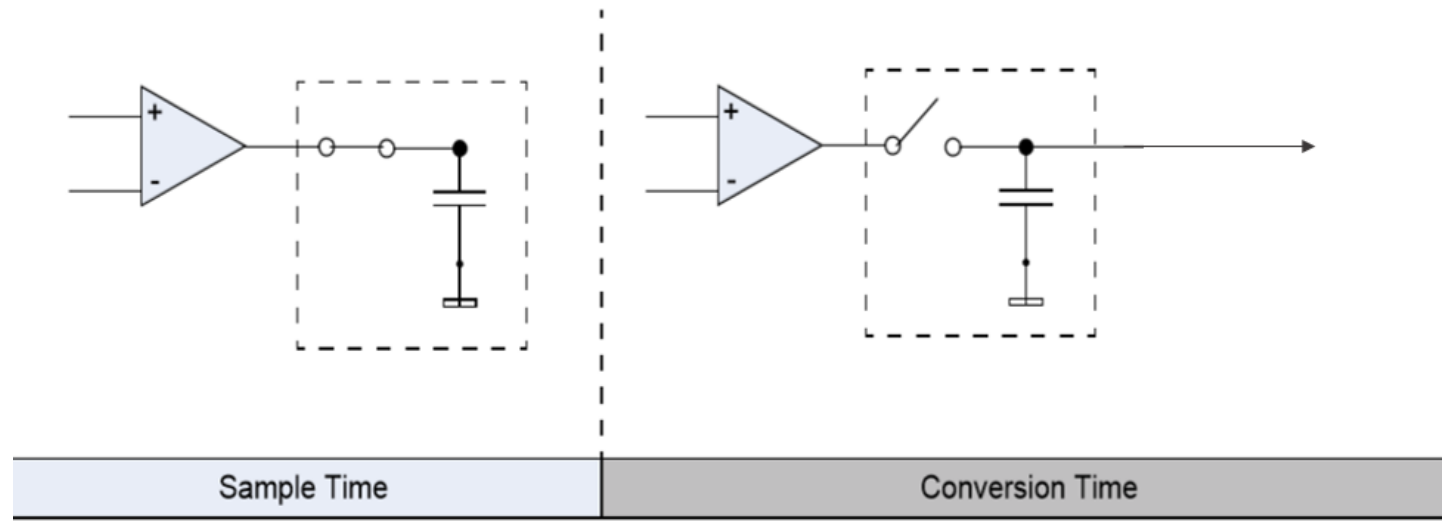
Establece una
cantidad de valores
finitos a la señal
muestreada

Asignar una
palabra digital a
cada valor
cuantificado

DIGITALIZACIÓN

SAMPLE & HOLD

Circuito analógico que se encarga de muestrear continuamente la señal de entrada y luego retener dicho valor hasta que el conversor A/D lo pueda digitalizar



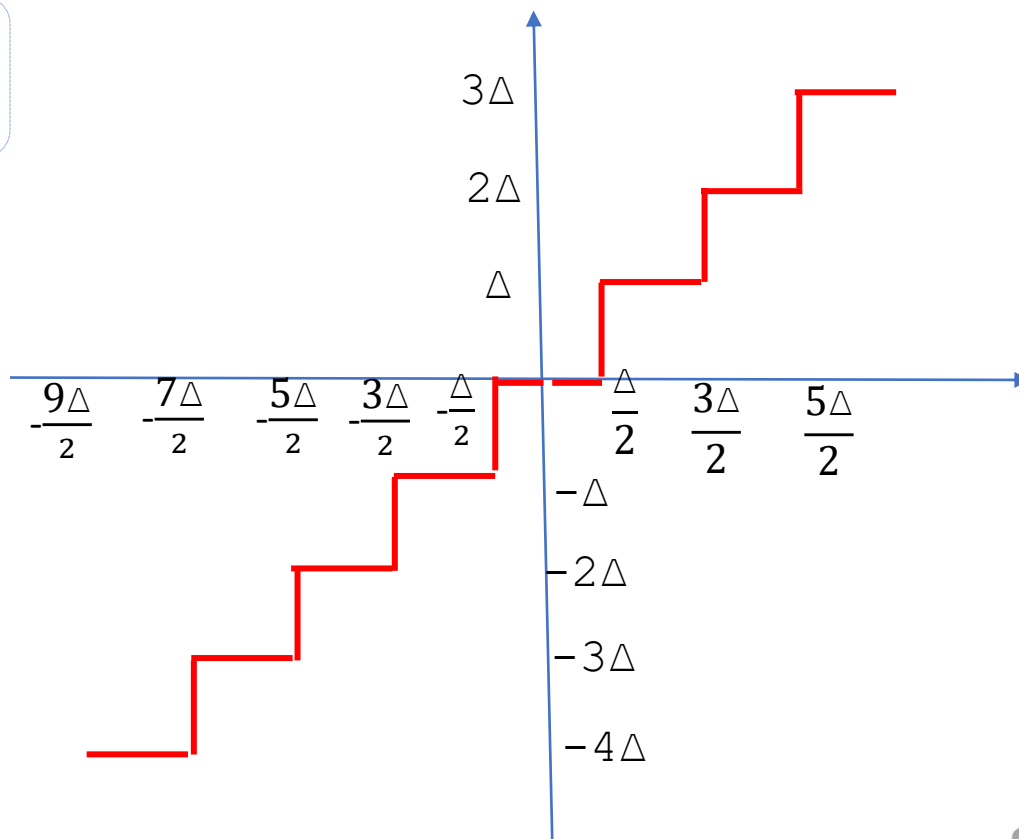
Δ

DIGITALIZACIÓN

CUANTIFICACIÓN

Señal con valores reales

$x \rightarrow Q(x) \rightarrow x_q$



CUANTIFICACIÓN UNIFORME

DIGITALIZACIÓN

CODIFICACIÓN

La codificación consiste en asignar una palabra digital a cada valor previamente cuantificado. La representación binaria puede variar de acuerdo al fabricante del ADC

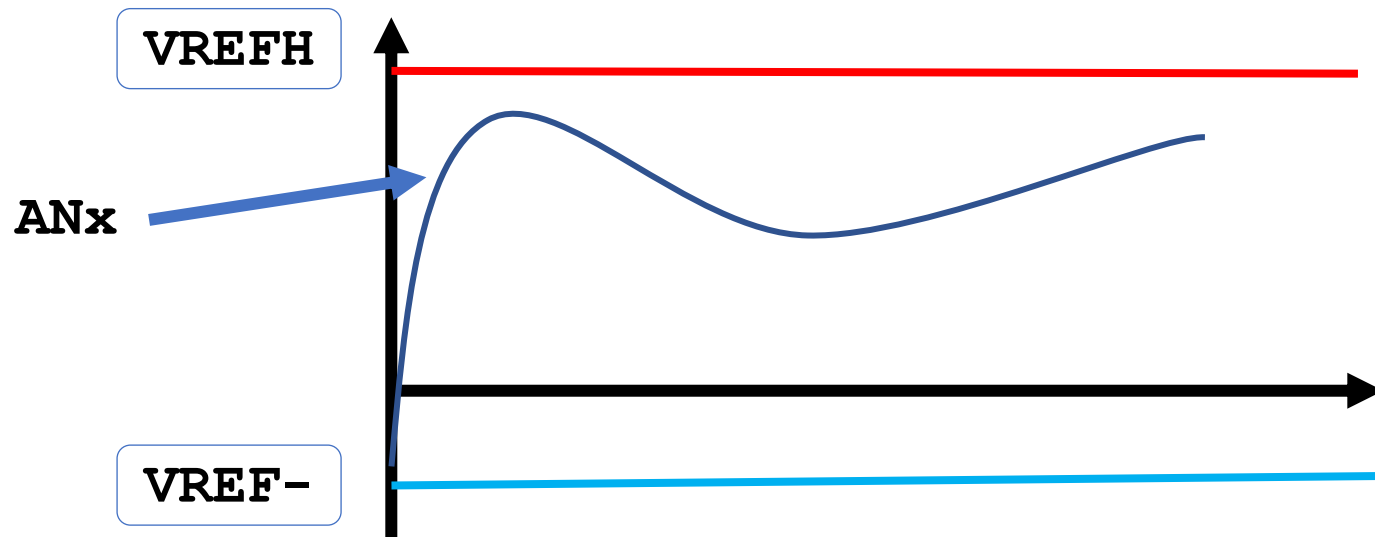
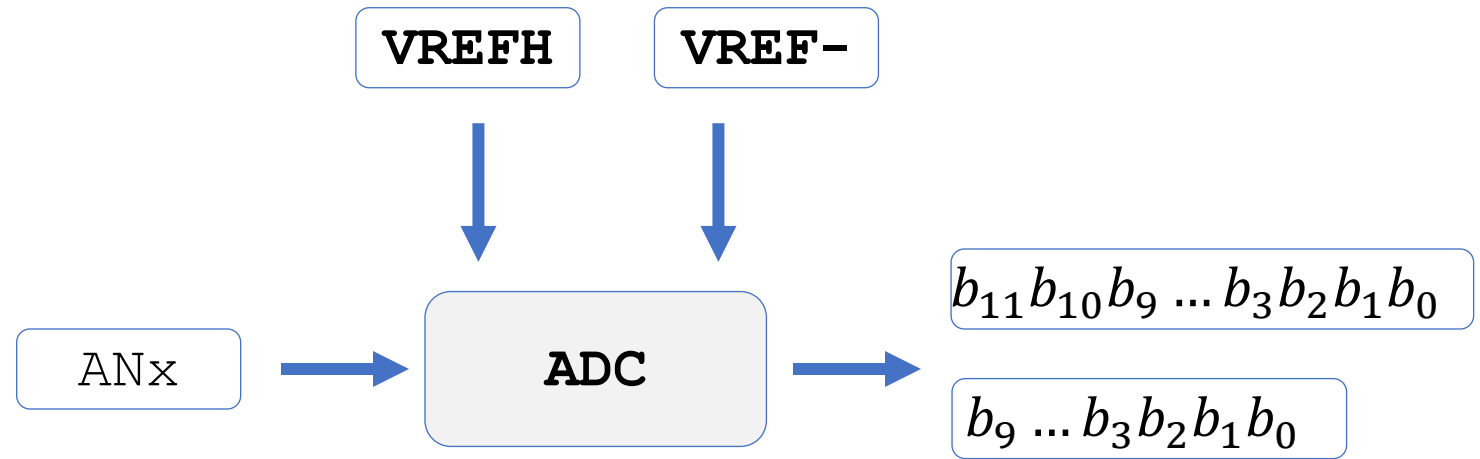
REPRESENTACIÓN BINARIA

Complemento a 2

Complemento a 1

signo-magnitud

DIGITALIZACIÓN



$$\Delta = \frac{V_{REFH} - V_{REFL}}{2^n - 1}$$

PROCESAMIENTO DIGITAL DE LA SEÑAL

En esta etapa se emplea algoritmos que deben ser diseñados con el fin de cumplir alguna aplicación

Estos algoritmos se pueden realizar en distintos lenguajes de programación e implementarlos en hardware (DSP , uC) .

TRANSFORMADA DE FOURIER

*Ecuación de
análisis*

$$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi Ft} dt$$

*Ecuación de
síntesis*

$$x(t) = \int_{-\infty}^{\infty} X(F) e^{j2\pi Ft} dF$$

herramienta matemática para realizar la descomposición de ciertos tipos de señales, definidos en tiempo continuo ,en el dominio de la frecuencia.

Esto se logra proyecta la función señal $x(t)$ sobre funciones ortonormales $e^{-j2\pi Ft}$.

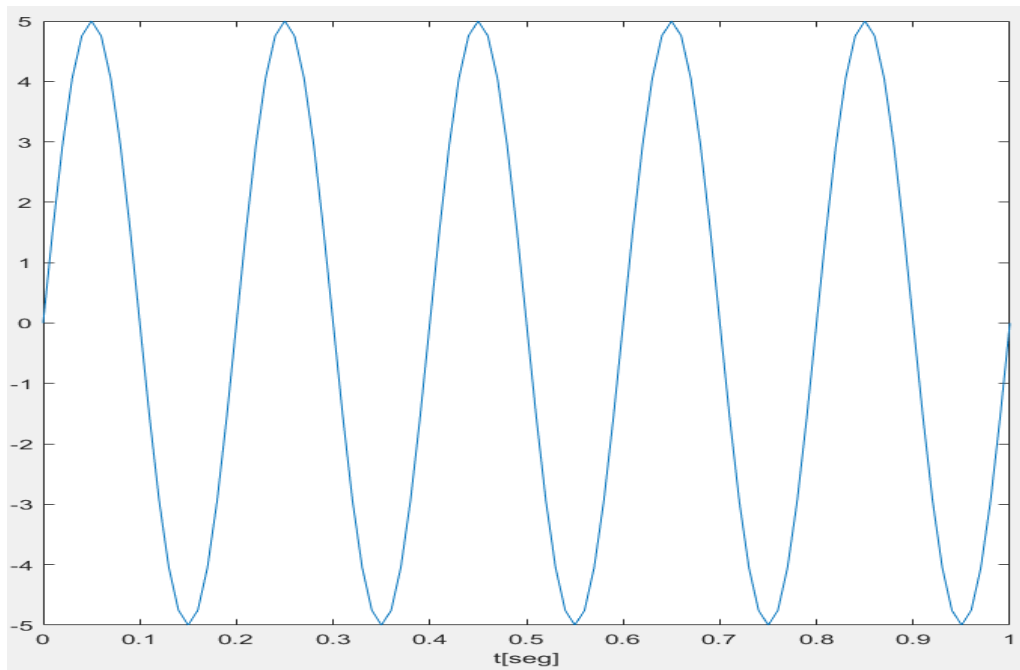
Aplicable a señales aperiodicas y periodicas en tiempo continuo.

TRANSFORMADA DE FOURIER DE SEÑAL EN TIEMPO
CONTINUO

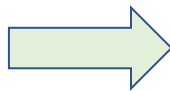
TRANSFORMADA DE FOURIER

$$y1 = 5 * \sin(2 * \pi * F1 * t)$$

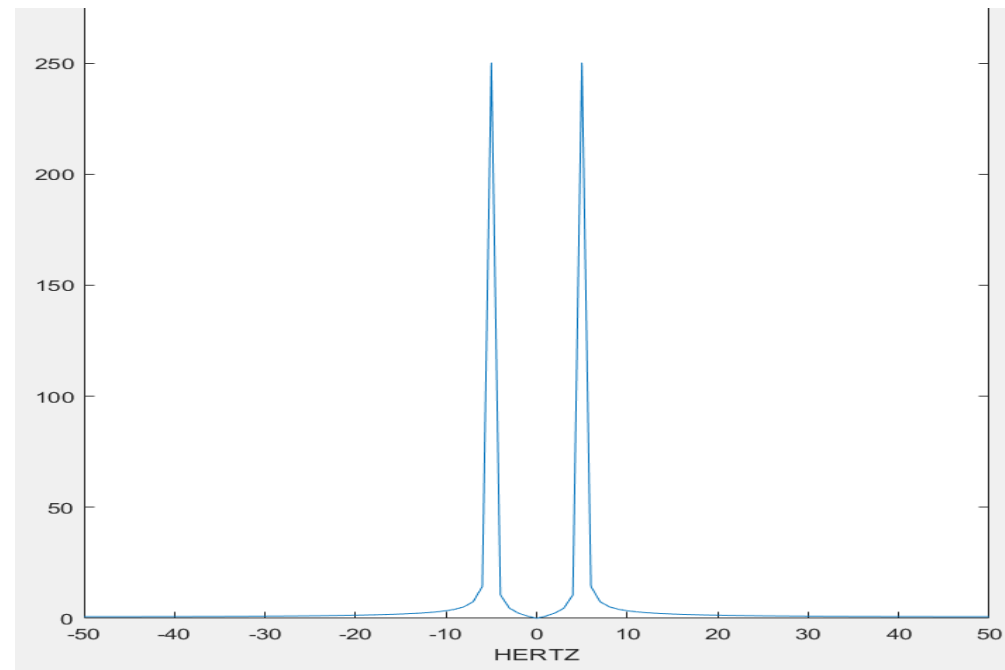
$F1 = 5 \text{ Hertz}$



$TF\{.\}$



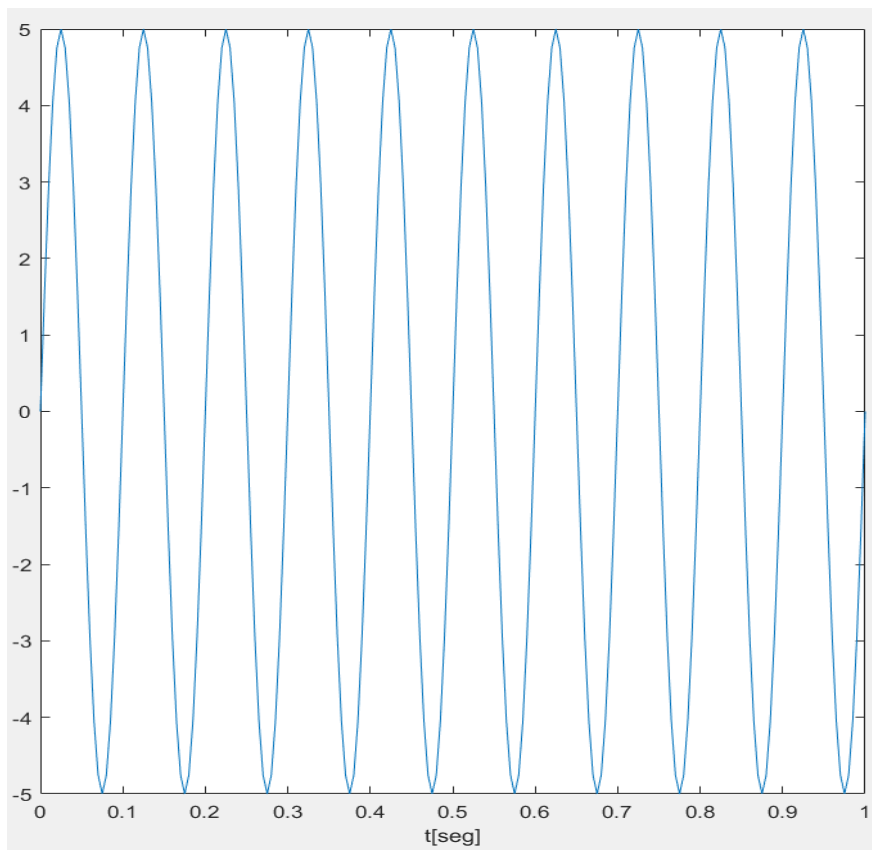
*Magnitud de la
transformada de Fourier*



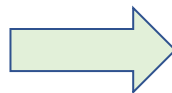
TRANSFORMADA DE FOURIER

$$y2 = 5 * \sin(2 * \pi * F2 * t)$$

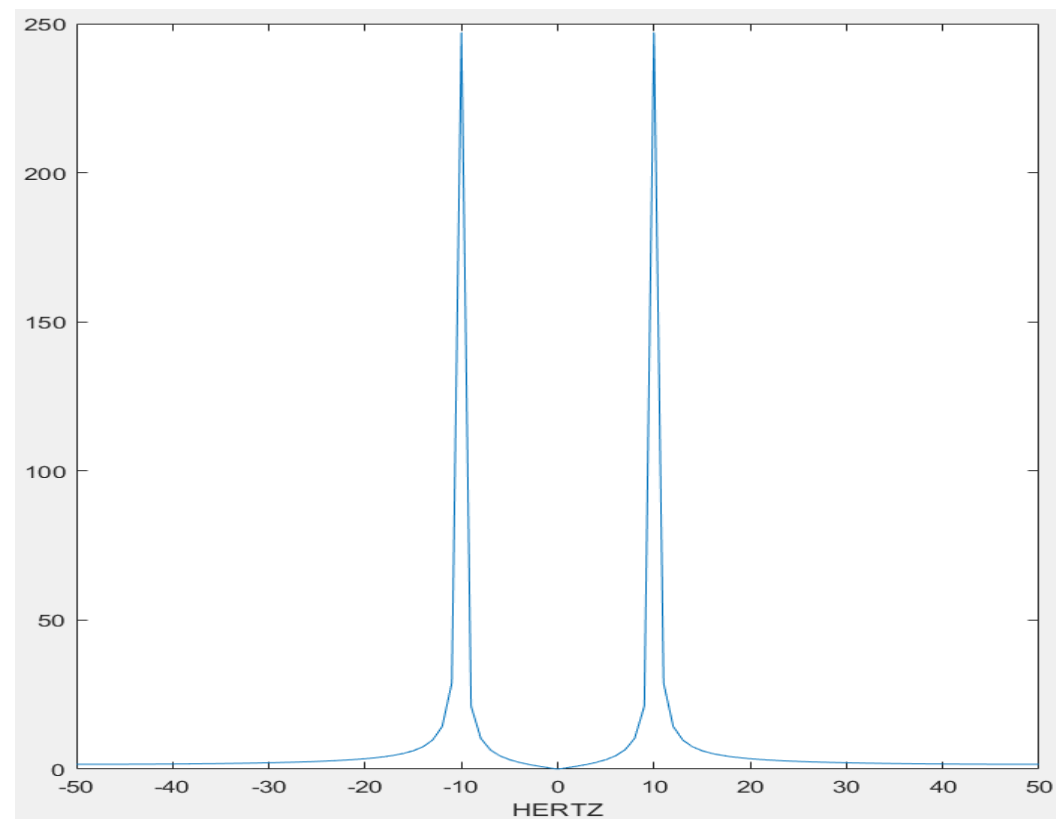
$F2 = 10 \text{ Hertz}$



$TF\{.\}$



*Magnitud de la
transformada de Fourier*

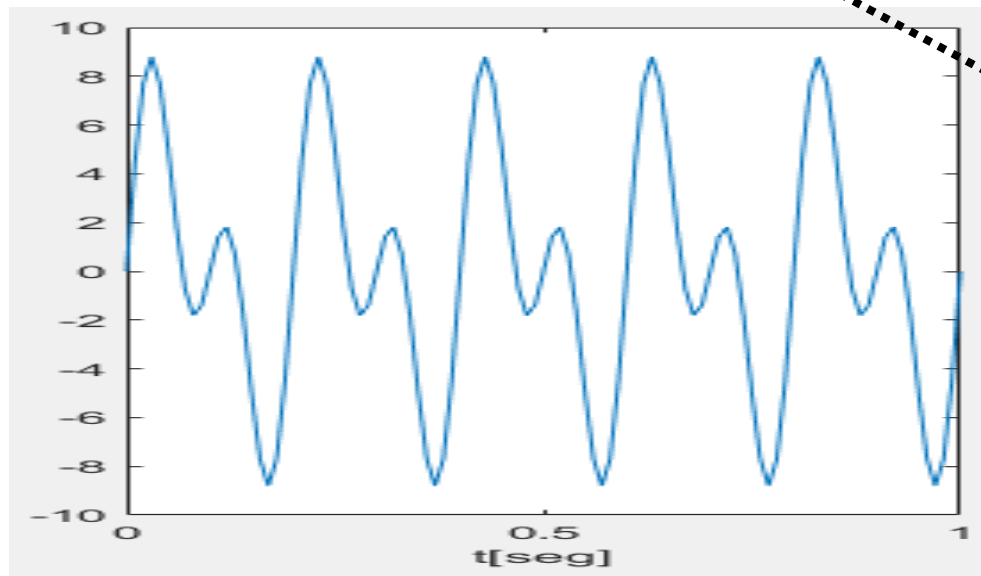


TRANSFORMADA DE FOURIER

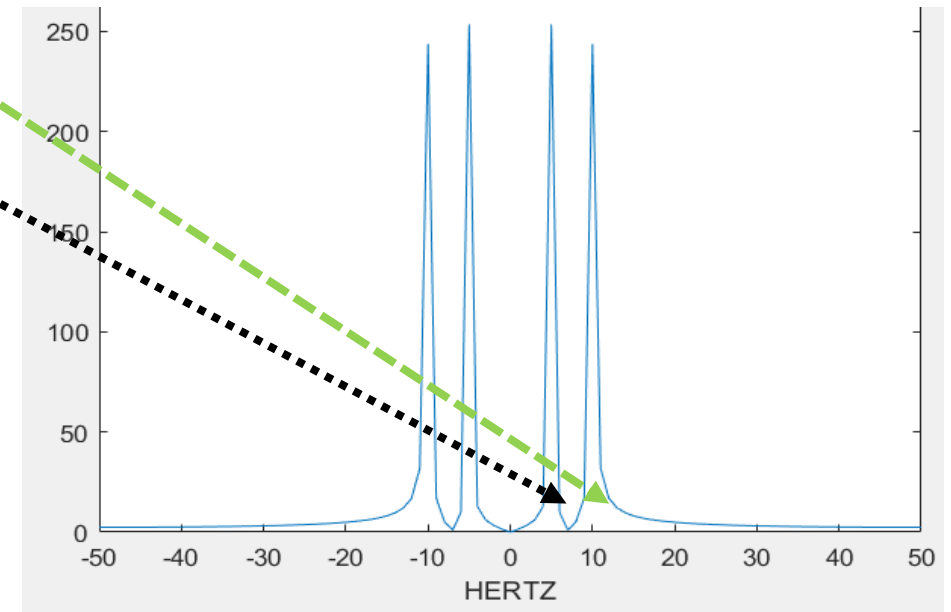
$$y = 5 * \sin(2 * \pi * F1 * t) + 5 * \sin(2 * \pi * F2 * t)$$

$$F1 = 5 \text{ Hertz} , \quad F2 = 10 \text{ Hertz}$$

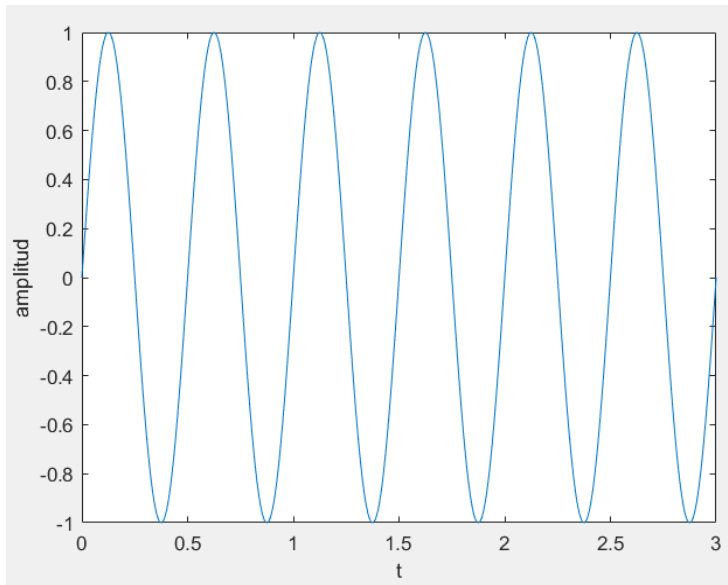
*Magnitud de la
transformada de Fourier*



$TF\{.\}$



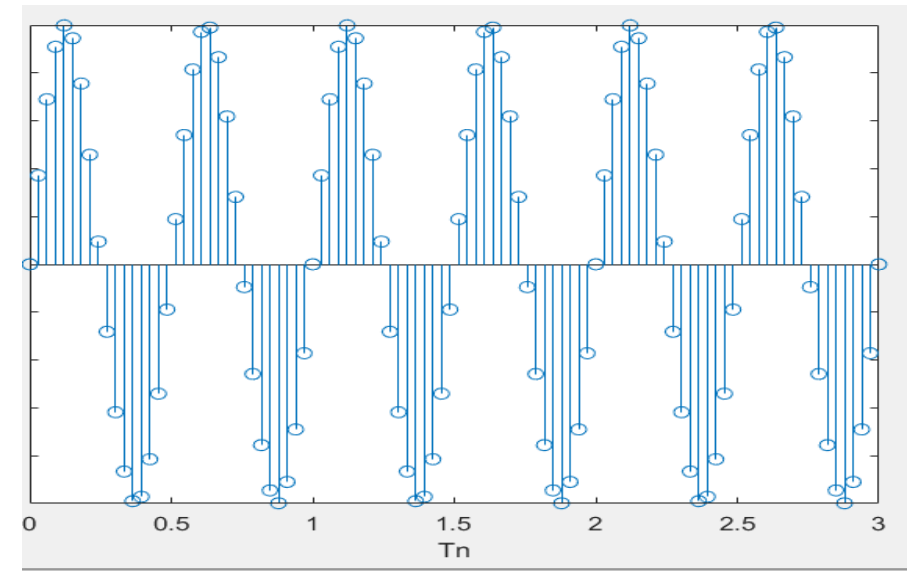
TEOREMA DE MUESTREO



MUESTREADOR



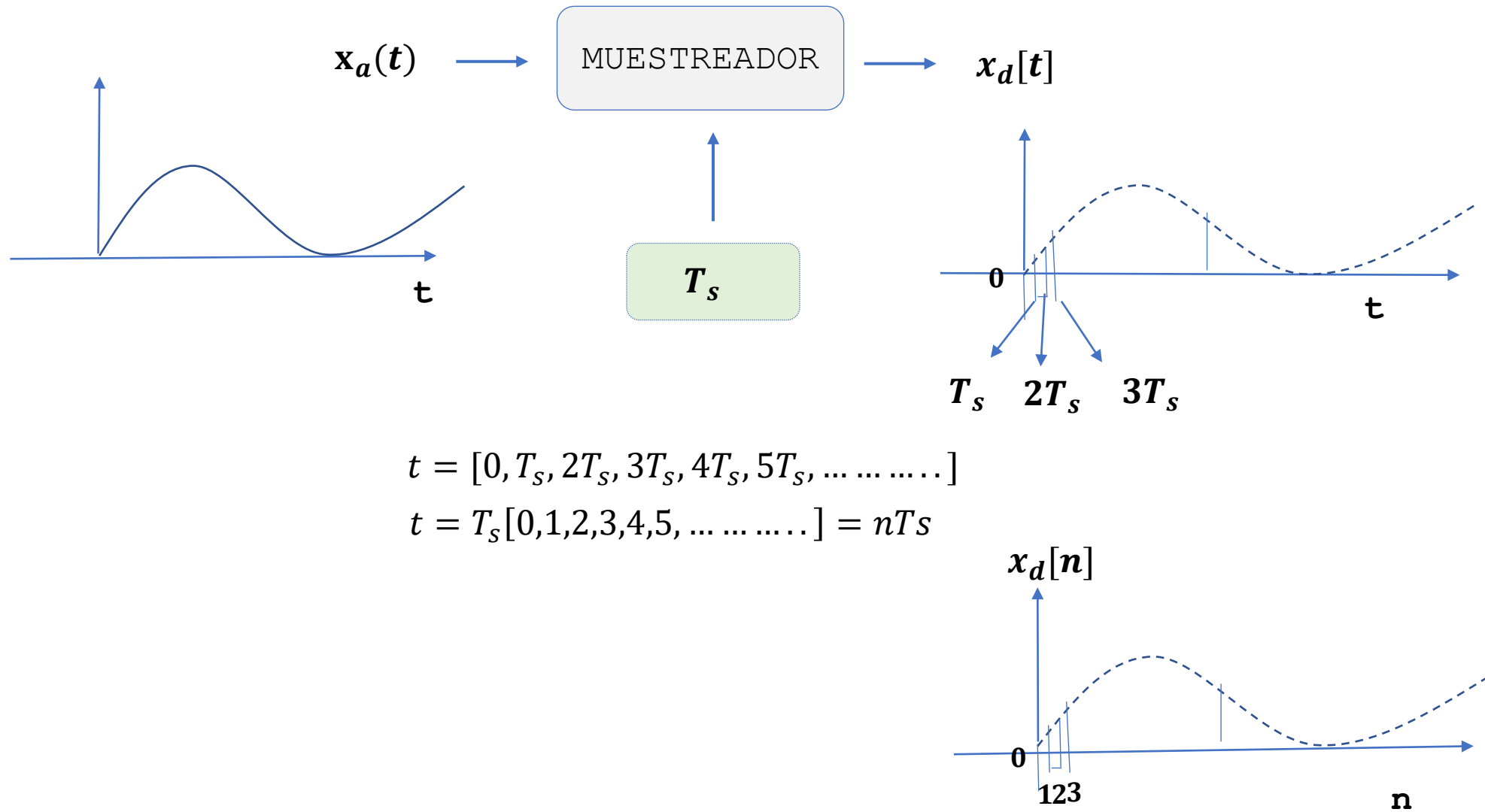
T_s



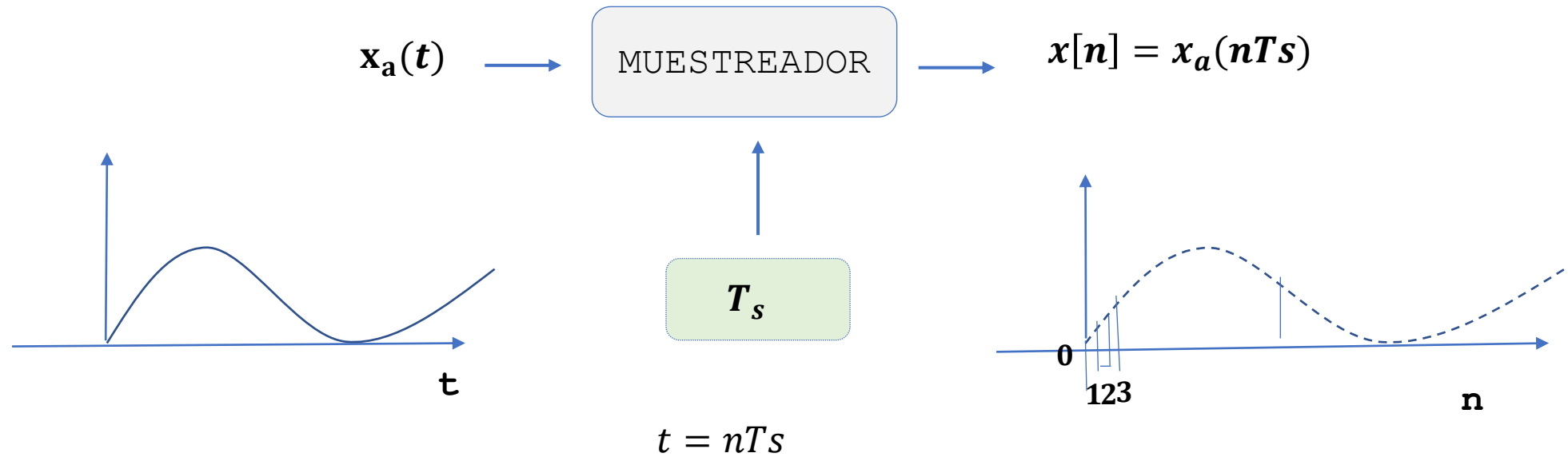
T_s : Periodo de muestreo (segundos)

F_s : Frecuencia de muestreo (Hz)

MUESTREO



MUESTREO



$$n = [\dots, -3, -2, -1, 0, 1, 2, 3, 4, 5, \dots]$$

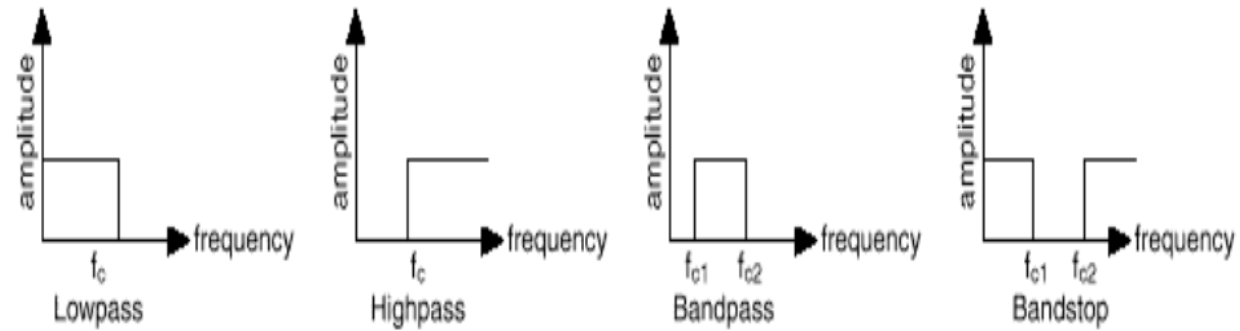
TEOREMA DE NYQUIST

El teorema de Nyquist establece una condición en la elección de la frecuencia de muestreo para evitar el efecto del **aliasing** en el dominio de la frecuencia y la reconstrucción de la señal a partir de sus muestras.

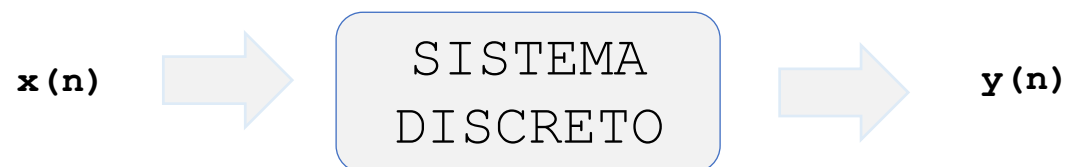
$$F_s \geq 2B$$

Para el caso de una señal de banda limitada con una máxima frecuencia de B Hz.

TIPOS DE FILTROS SEGÚN SU RESPUESTA EN FRECUENCIA



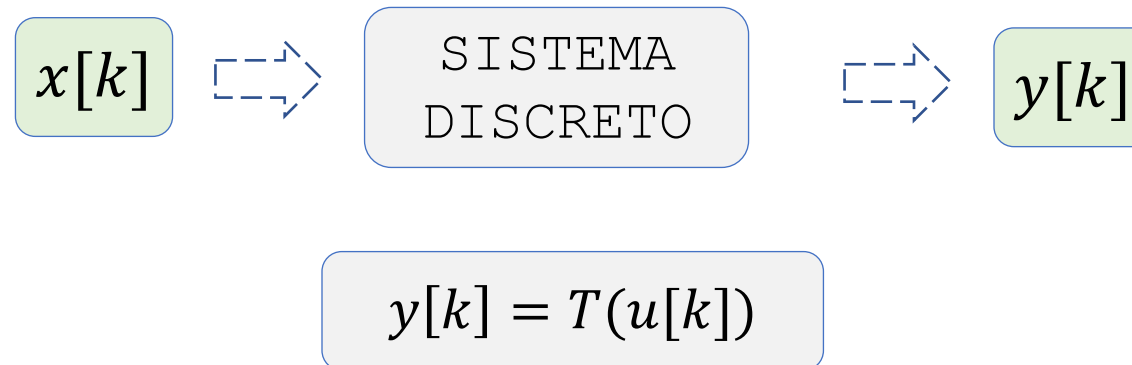
ECUACIÓN EN DIFERENCIAS



Ecuación que describe el comportamiento entrada-salida de un sistema discreto lineal o no lineal. En el contexto lineal e invariante en el tiempo , se pueden aplicar herramientas matemáticas como la DFT, transformada Z para simplificar su análisis .

SISTEMA DISCRETO

Dispositivo o algoritmo que opera sobre una o mas señales definidas en tiempo discreto mediante una regla definida produciendo una respuesta del sistema



La señal de salida se obtiene mediante la convolución de la entrada y la respuesta al impulso del sistema

CARACTERISTICAS DE LOS SISTEMAS DISCRETOS



causalidad

Invarianza en el
tiempo

estabilidad

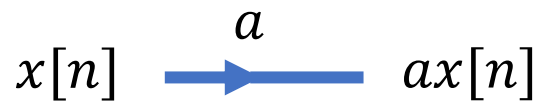
linealidad

controlabilidad

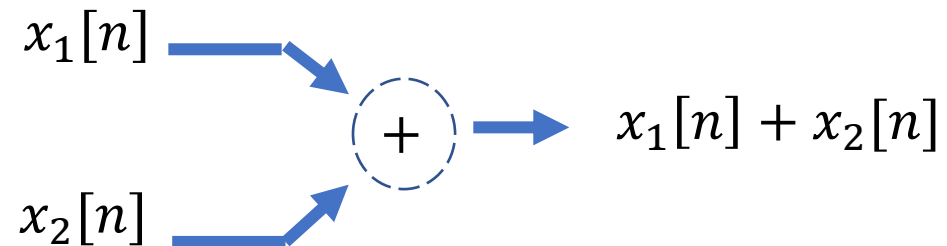
observabilidad

OPERACIONES ELEMENTALES

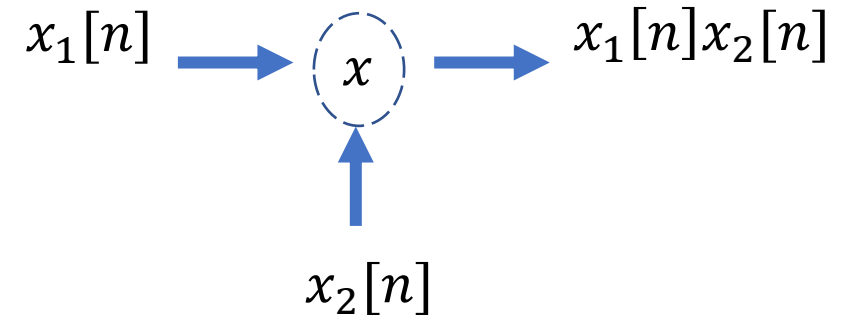
*Multiplicación
por una constante*



*Suma de señales
discretas*

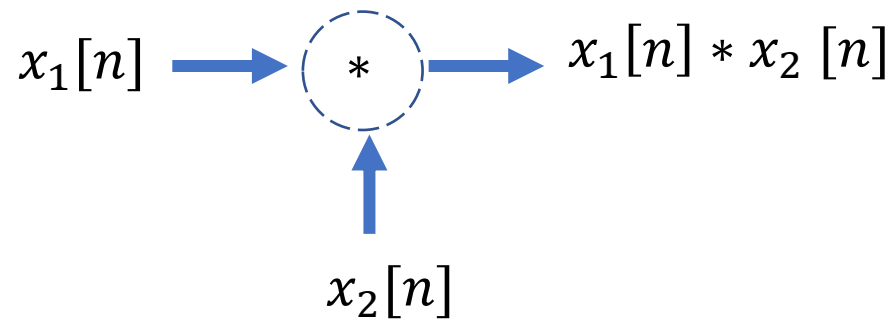


*Producto de
señales discretas*

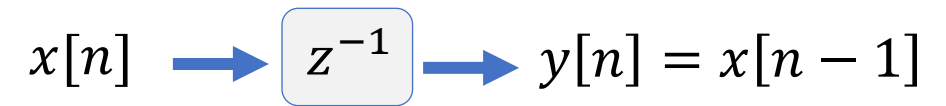


OPERACIONES ELEMENTALES

Convolución de
señales

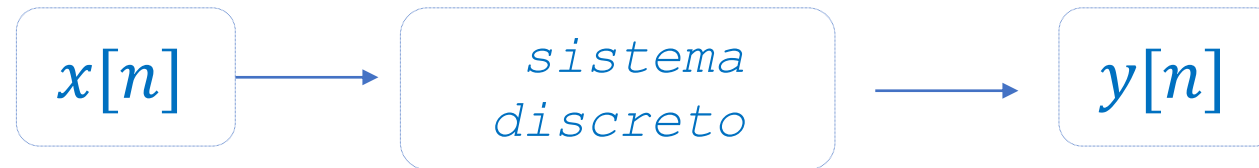


Retardo unitario



ECUACIÓN EN DIFERENCIAS

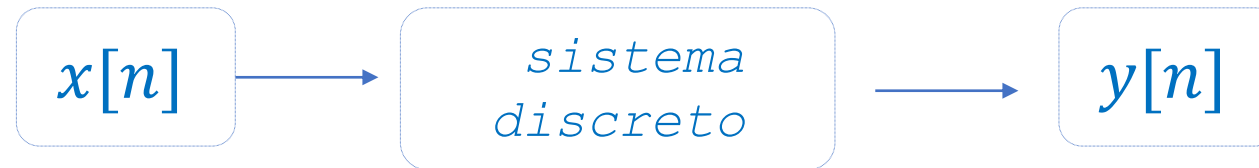
EJEMPLO 1



$$y[n] = 0.1x[n] + 0.2x[n - 1] + 0.1x[n - 2]$$

ECUACIÓN EN DIFERENCIAS

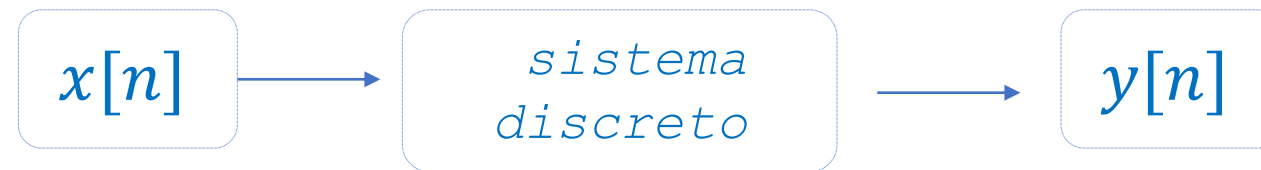
EJEMPLO 2



$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}$$

ECUACIÓN EN DIFERENCIAS

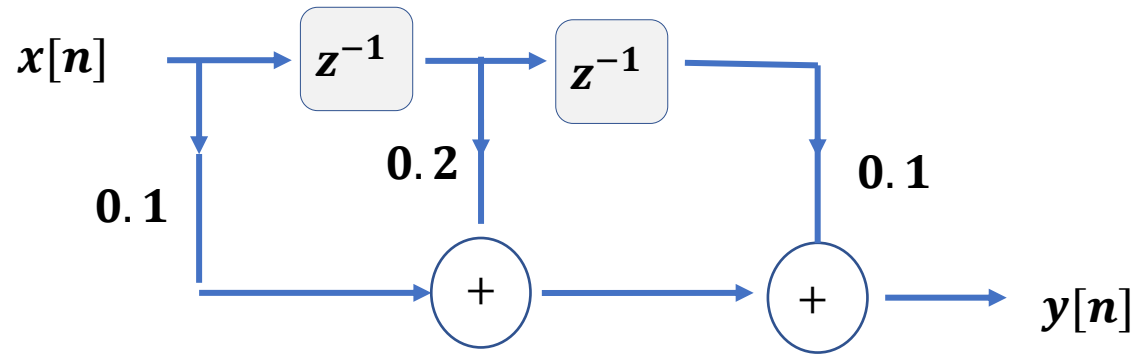
EJEMPLO 3



$$y[n] = 0.1x[n] + 0.1x[n - 2] - 0.5y[n - 1]$$

ESTRUCTURA DE IMPLEMENTACIÓN

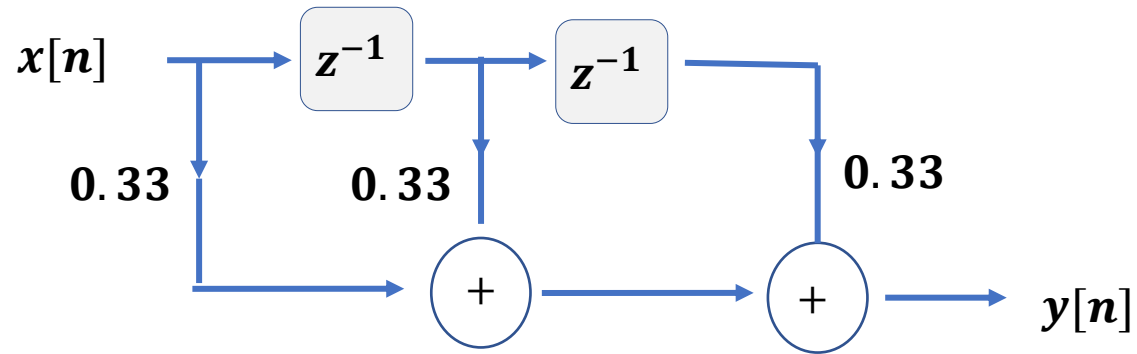
EJEMPLO 1



$$y[n] = 0.1x[n] + 0.2x[n - 1] + 0.1x[n - 2]$$

ESTRUCTURA DE IMPLEMENTACIÓN

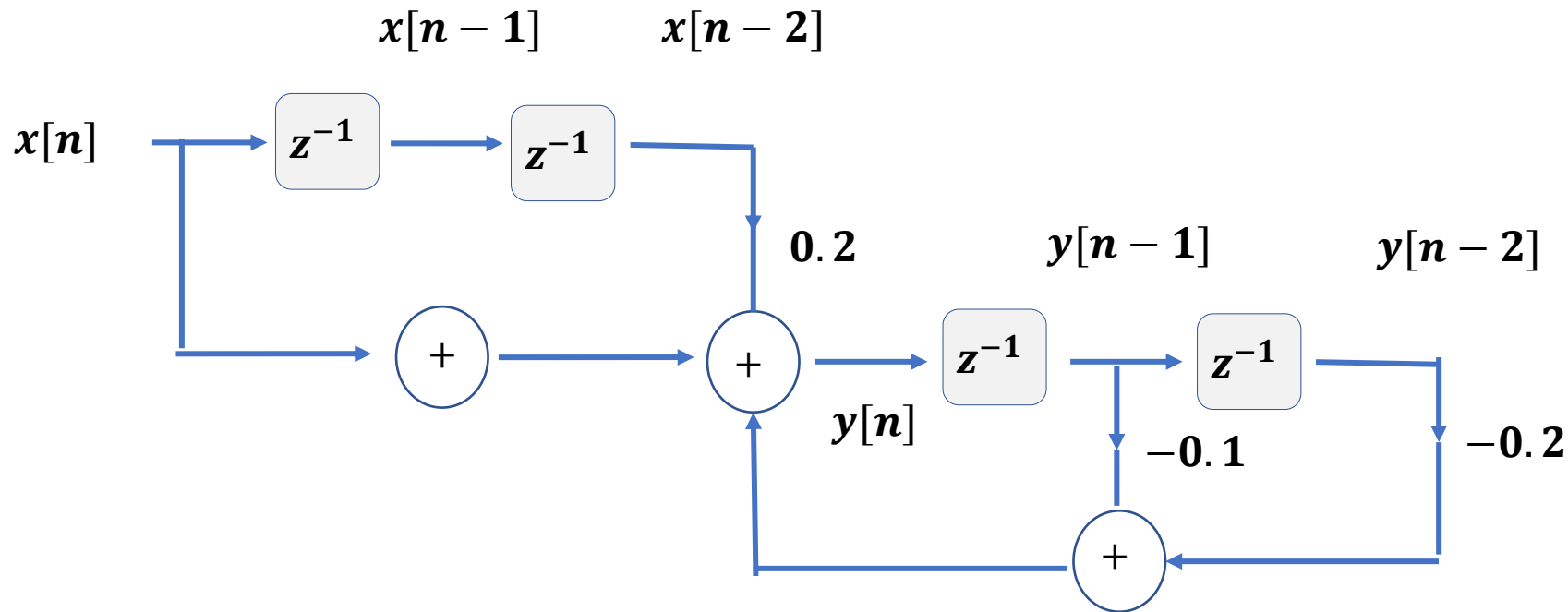
EJEMPLO 2



$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3}$$

ESTRUCTURA DE IMPLEMENTACIÓN

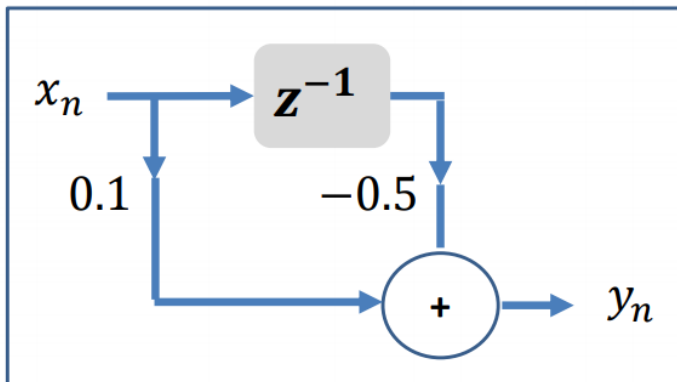
EJEMPLO 3



$$y[n] = x[n] + 0.1x[n-2] - 0.1y[n-1] - 0.2y[n-2]$$

$$y_n = 0.1x_n - 0.5x_{n-1}$$

Estructura de implementación



```
1  #include <stdio.h>
2  float x_n1=0; //condición inicial
3  float x[]={1,2,2.5,3}; //arreglo flotante (input)
4  float y[4]; //arreglo flotante (output)
5  int n;
6  int main(void)
7  {
8      for(n=0;n<=3;n++)
9      {
10         //algoritmo
11         y[n]=0.1*x[n]-0.5*x_n1;
12         //Actualizar condición inicial
13         x_n1=x[n];
14         //mostrar resultado en cada iteración
15         printf("%.2f \n",y[n]);
16     }
17
18     return 0;
19 }
```

DATOS IMPORTANTES

Los archivos (.mat) representan datos de señal EOG normalizado entre $[-1 \ 1]$. La frecuencia de muestreo F_s fue de 500Hz.

Archivos
(.mat)


dato_H1.mat

dato_V1.mat

.Determinar la duración de cada señal en segundos.

.Graficar la magnitud de su espectro de frecuencia.

```
load('dato_H1.mat');
```

Name ▲	Value
 h	1x2502 double

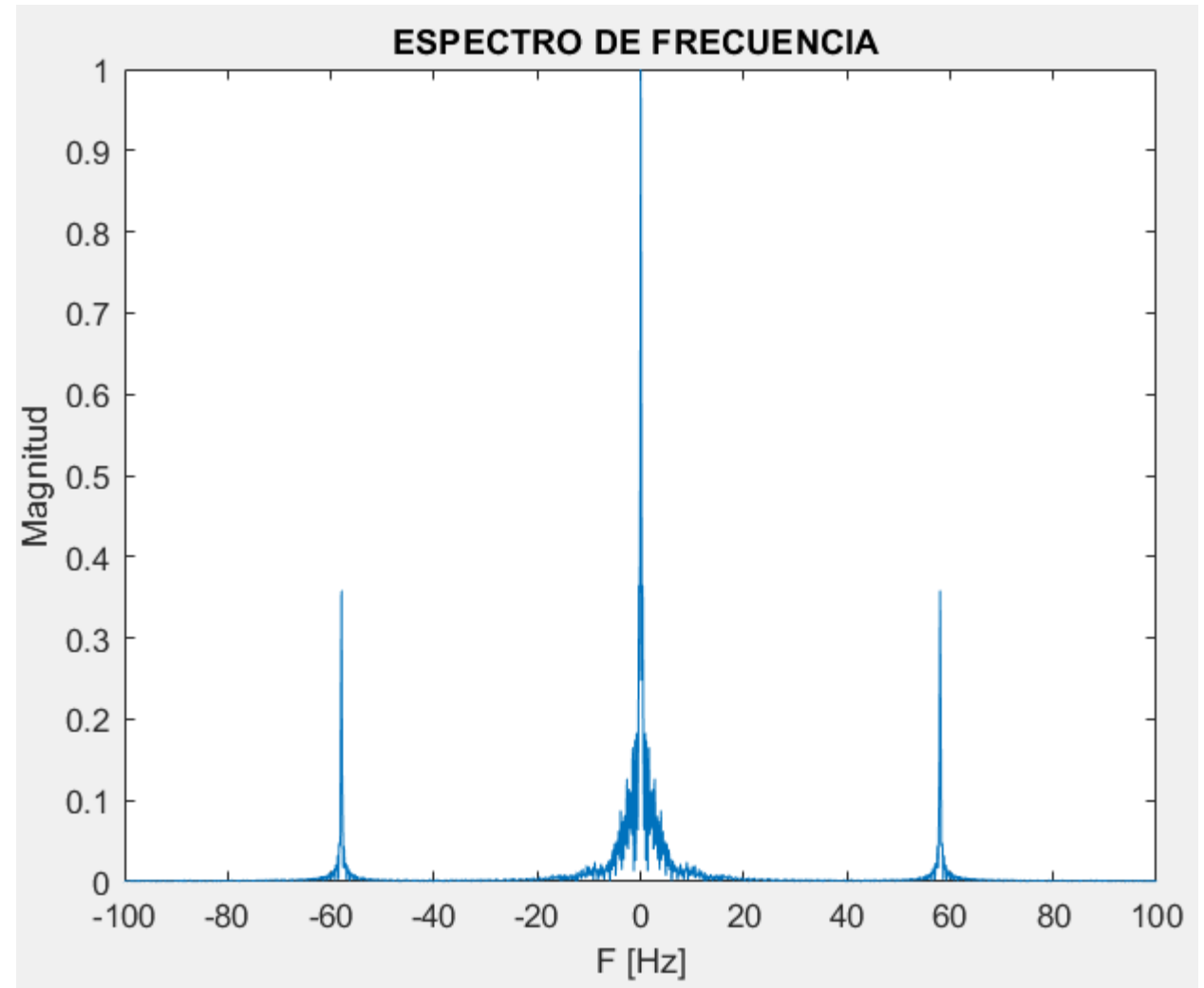
```
plot(F,X_m);  
title('ESPECTRO DE FRECUENCIA ')  
xlabel('F [Hz]')  
ylabel('Magnitud')  
xlim([-100,100])
```

```
plot(x,y)
```

Los vectores x e y deben tener la misma longitud

```
xlim([min,max])
```

Limita el campo de visibilidad en el eje x entre [min,max]



COMANDOS DE MATLAB

`linspace(x,y,N)`

Crea vector de N elementos igualmente espaciados en el intervalo [x y]

`length`

Determina la cantidad de elementos de un arreglo

`size`

Determina la forma del arreglo

`max`

Encuentre el máximo o máximos valores de un arreglo

COMANDOS DE MATLAB

`nextpow2(L)`

Retorna el exponente de la siguiente potencia superior de 2. $2^p \geq L$

`fft(x,N)`

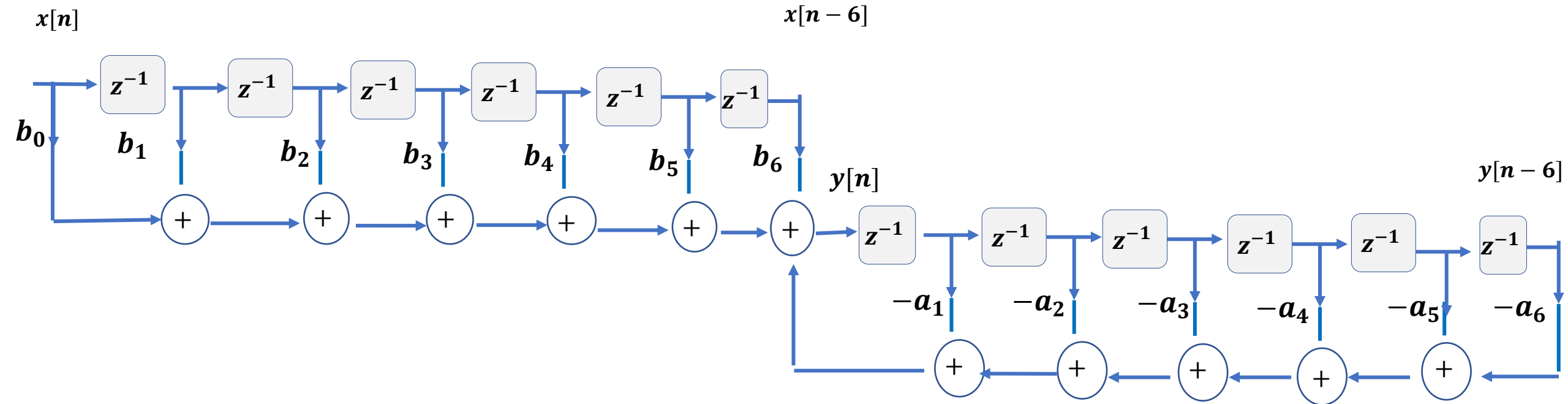
Determina la DFT (Discrete Fourier Transform) de N puntos de una señal x

`abs`

En una señal real determina su valor absoluto , en una señal compleja retorna su magnitud.

`Ifft(X,N)`

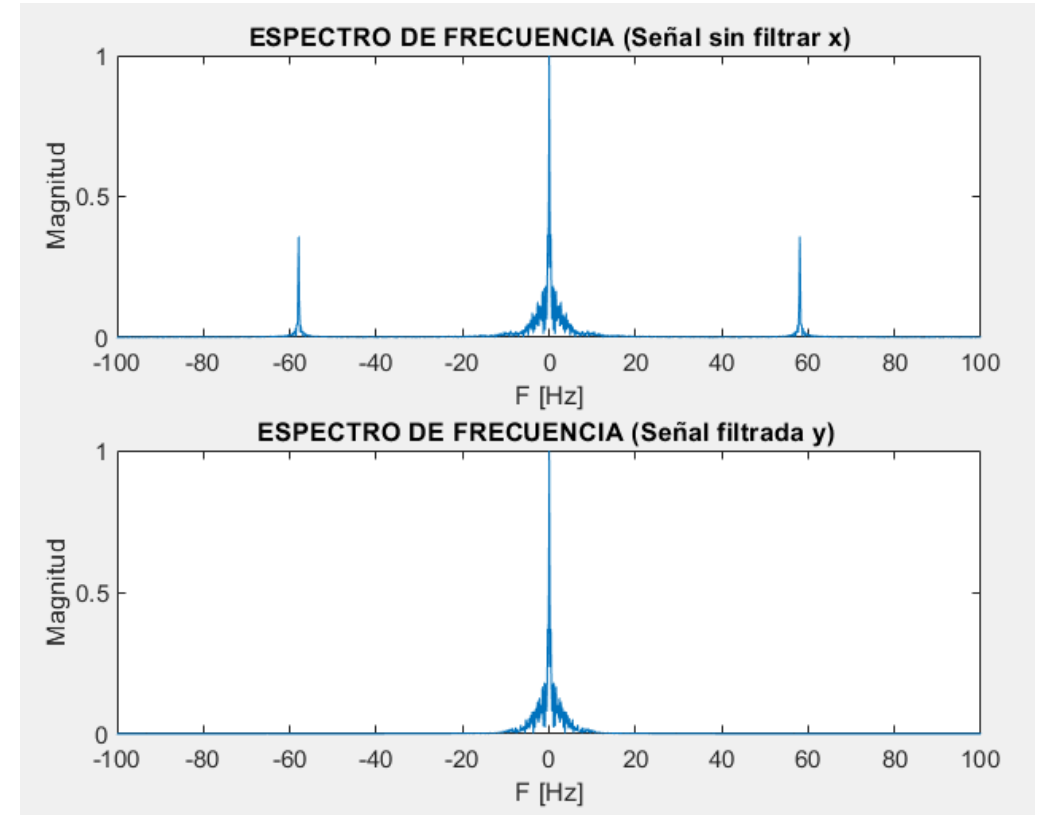
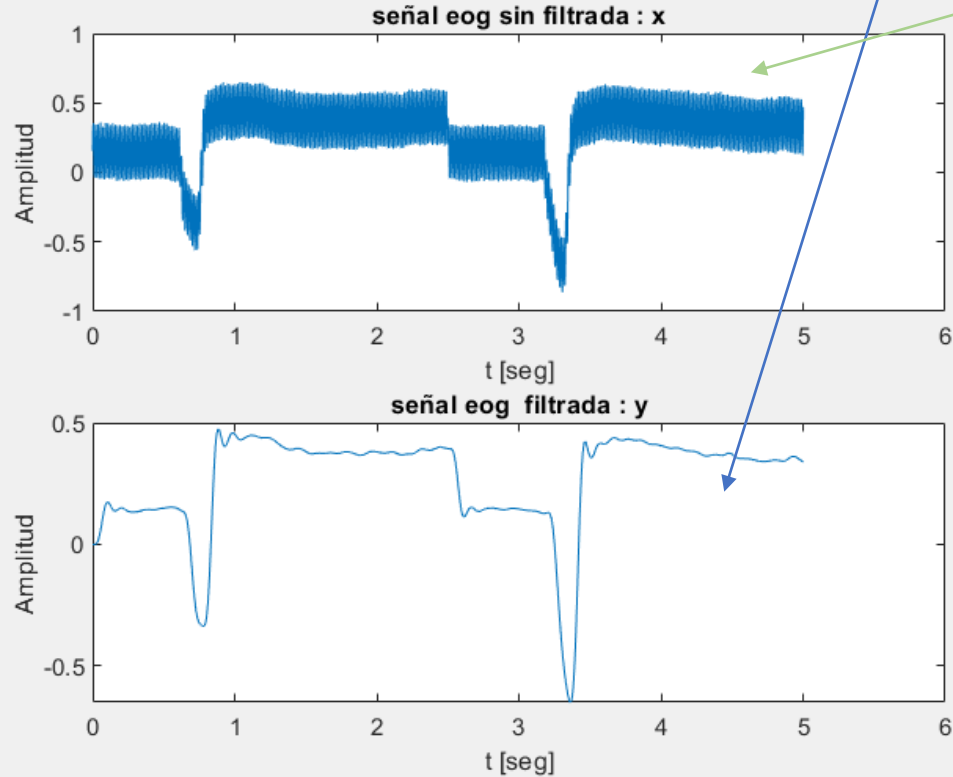
Determina la IDFT (Inverse Discrete Fourier Transform) de N puntos de una señal X.



$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + b_3x[n-3] + b_4x[n-4] + b_5x[n-5] + b_6x[n-6] - a_1y[n-1] - a_2y[n-2] - a_3y[n-3] - a_4y[n-4] - a_5y[n-5] - a_6y[n-6]$$

COMANDOS DE MATLAB

```
y=filter(b,a,x)
```



APLICACIÓN DEL FILTRO MEDIANTE CODIGO

```
%almacena en x[];
```

```
for k=1:L
```

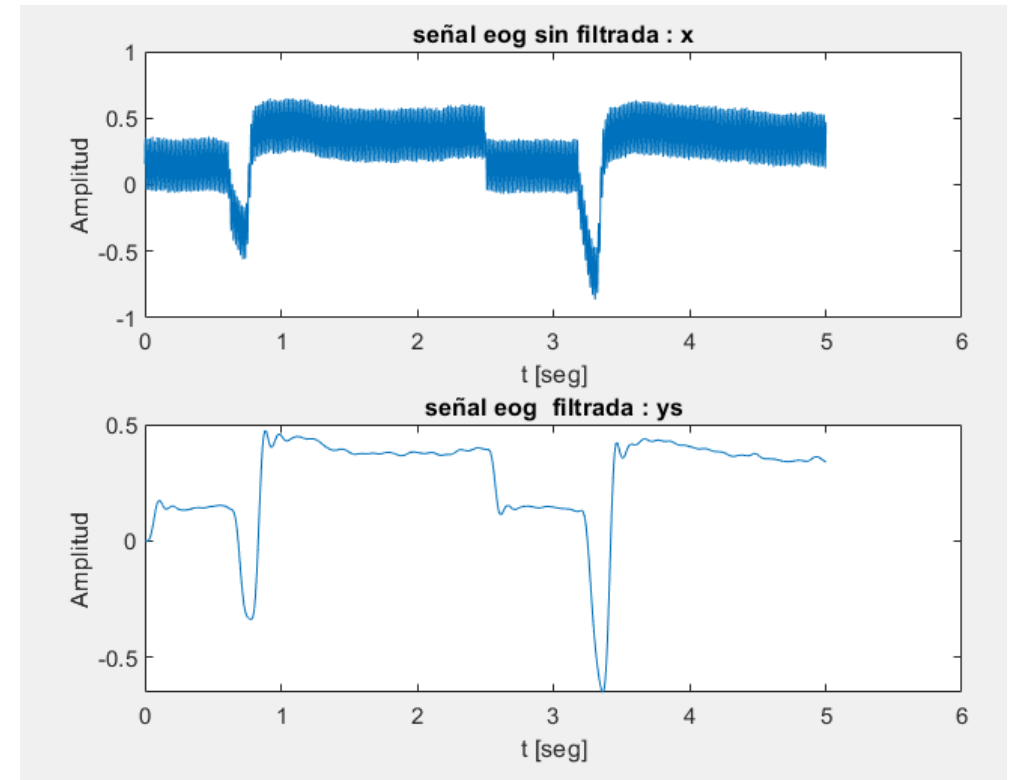
```
yk=b(1)*x(k)+b(2)*xk1+b(3)*xk2+b(4)*xk3+b(5)*xk4+b(6)*xk5+b(7)*xk6...  
-a(2)*yk1-a(3)*yk2-a(4)*yk3-a(5)*yk4-a(6)*yk5-a(7)*yk6;
```

```
%actualizaciones
```

```
xk6=xk5;  
xk5=xk4;  
xk4=xk3;  
xk3=xk2;  
xk2=xk1;  
xk1=x(k);
```

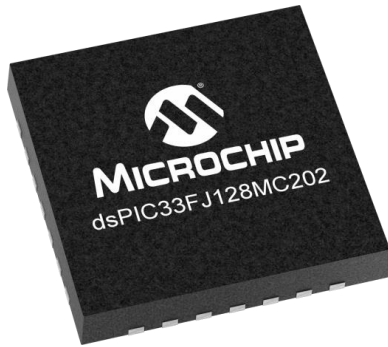
```
yk6=yk5;  
yk5=yk4;  
yk4=yk3;  
yk3=yk2;  
yk2=yk1;  
yk1=yk;  
ys(k)=yk;
```

```
end
```



IMPLEMENTACIÓN EN HARDWARE

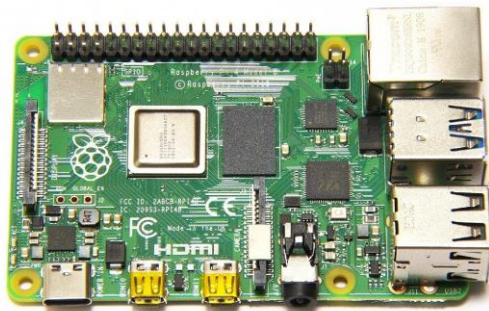
dsPIC33FJ



FPGA



Raspberry *pi*



TIVA C
TM4C123GH6PM

