

# *Sistemas embebidos II*

## *Sesión 4*

# Elementos del lenguaje (Entidad)

La entidad define la interfaz de un componente de hardware, describiendo sus entradas y salidas. Es el bloque externo del diseño y describe qué hace el sistema sin entrar en detalles de cómo lo hace. Dentro de la entidad, se declara el puerto (las señales de entrada y salida).

**Module Definition**

Entity name:

Architecture name:

**I/O Port Definitions**

Port Name	Direction	Bus	MSB	LSB
	<div><div>in</div><div>in</div><div>out</div><div>inout</div></div>	<input type="checkbox"/>	0	0

?

OK Cancel

```
entity AND_GATE is
    port(
        A : in  std_logic;
        B : in  std_logic;
        Y : out std_logic
    );
end AND_GATE;
```

# Elementos del lenguaje (Arquitectura)

La arquitectura describe el funcionamiento interno o la implementación de la entidad. Aquí se detalla el cómo se implementa el comportamiento del diseño especificado en la entidad. Puede contener descripciones comportamentales, estructurales o una mezcla de ambas.

```
architecture Behavioral of AND_GATE is
begin
    Y <= A and B;  -- Comportamiento de la compuerta AND
end Behavioral;
```

# Elementos del lenguaje (Proceso)

```
process (A, B)
begin
    if (A = '1' and B = '1') then
        Y <= '1';
    else
        Y <= '0';
    end if;
end process;
```

Los procesos son bloques secuenciales dentro de una arquitectura que permiten describir comportamientos que dependen del tiempo o de eventos. Un proceso puede contener sentencias condicionales (if, case), bucles y otras estructuras secuenciales. Los procesos son muy útiles para describir circuitos secuenciales como flip-flops o máquinas de estados.

# Elementos del lenguaje (Señales y Variables)

Las señales se utilizan para la comunicación entre diferentes procesos o entre la entidad y la arquitectura. Las variables (variable), por otro lado, son locales a un proceso y se usan para almacenar valores temporales dentro de un proceso. Las señales tienen un retraso inherente (simulan el retardo de propagación en el hardware), mientras que las variables son instantáneas dentro del proceso.

```
signal temp: std_logic;  
variable counter: integer := 0;
```

# Elementos del lenguaje (Tipos de Datos)

VHDL incluye diversos tipos de datos para modelar señales y variables, los más comunes son:

- `std_logic`: Representa un bit y puede tomar varios valores como '0', '1', 'Z' (impedancia alta), 'X' (indeterminado), etc.
- `std_logic_vector`: Un vector de varios bits.
- `integer`: Números enteros.
- `boolean`: Verdadero o falso.

```
architecture ejemplo of tipos_de_datos is
    -- Declaración de señales con diferentes tipos de datos

    -- BIT: un único bit ('0' o '1')
    signal bit_example : BIT := '1';

    -- BIT_VECTOR: vector de bits
    signal bit_vector_example : BIT_VECTOR(3 downto 0) := "1010";

    -- STD_LOGIC: más flexibilidad que BIT ('0', '1', 'Z', 'X', etc.)
    signal std_logic_example : STD_LOGIC := '1';

    -- STD_LOGIC_VECTOR: vector de señales STD_LOGIC
    signal std_logic_vector_example : STD_LOGIC_VECTOR(7 downto 0) := "11001100";

    -- INTEGER: enteros
    signal integer_example : INTEGER := 10;

    -- REAL: número real con decimales
    signal real_example : REAL := 3.14;

    -- SIGNED: número con signo (positivo o negativo) en forma binaria
    signal signed_example : SIGNED(7 downto 0) := to_signed(-15, 8);

    -- UNSIGNED: número sin signo (solo positivo) en forma binaria
    signal unsigned_example : UNSIGNED(7 downto 0) := to_unsigned(15, 8);

    -- BOOLEAN: valores lógicos (TRUE o FALSE)
    signal boolean_example : BOOLEAN := TRUE;
```



# Elementos del lenguaje (Concurrente vs Secuencial)

Concurrente: Las sentencias concurrentes se ejecutan simultáneamente y son naturales para describir lógica combinacional. Ejemplos de sentencias concurrentes incluyen asignaciones simples o instancias de componentes.

Secuencial: Dentro de los procesos, las sentencias se ejecutan de manera secuencial, lo cual es útil para describir el comportamiento temporal o secuencial de un sistema.

# Elementos del lenguaje (Asignación de Señales)

La asignación de valores a las señales se hace usando el operador `<=` (asignación concurrente). Dentro de un proceso, las variables se asignan con el operador `:=`, pero las señales siguen usando `<=`.

```
signal Result: std_logic;  
Result <= A and B;  -- Asignación de señal
```

- Concurrente: Las sentencias concurrentes se ejecutan simultáneamente y son naturales para describir lógica combinatorial. Ejemplos de sentencias concurrentes incluyen asignaciones simples o instancias de componentes.
- Secuencial: Dentro de los procesos, las sentencias se ejecutan de manera secuencial, lo cual es útil para describir el comportamiento temporal o secuencial de un sistema.



# Elementos del lenguaje (Componentes e Instancias)

Los componentes permiten la reutilización de otros diseños dentro de un diseño más grande, similar a módulos en otros lenguajes de hardware. Puedes declarar e instanciar componentes dentro de la arquitectura.

```
component AND_GATE is
    port (A : in std_logic; B : in std_logic; Y : out std_logic);
end component;

U1: AND_GATE port map (A => signal_A, B => signal_B, Y => signal_Y);
```

# Elementos del lenguaje (ejemplo)

- Entidad compuertas\_and\_or:
  - Define las entradas (a, b, c, d) y salidas (and\_out, or\_out, sum\_out, real\_val).
  - a, b, c, d son señales de tipo STD\_LOGIC que se usan para las compuertas lógicas.
  - and\_out y or\_out son salidas de tipo STD\_LOGIC para los resultados de las compuertas AND y OR, respectivamente.
  - sum\_out es una salida de tipo INTEGER que mostrará el resultado de una operación aritmética.
  - real\_val es una salida de tipo REAL que mostrará el resultado de una operación con números reales.
- Arquitectura behavioral:
  - Se definen señales internas and\_result, or\_result, integer\_sum, y real\_example.
  - Compuerta AND: Calcula el resultado de la operación a and b y lo asigna a and\_result.
  - Compuerta OR: Calcula el resultado de c or d y lo asigna a or\_result.

# Elementos del lenguaje (ejemplo)

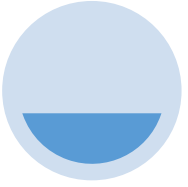
- Proceso:
  - Dentro del proceso: Se asignan los resultados de las compuertas AND y OR a las salidas `and_out` y `or_out`, respectivamente.
  - Se realiza una operación de suma con valores enteros concatenando `a` y `b`, convirtiendo la concatenación a un entero y asignando el resultado a `sum_out`.
  - Se realiza una operación aritmética con valores reales y se asigna el resultado a `real_val`.

# Filtros

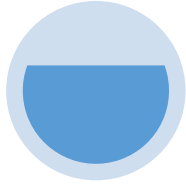
Es un circuito electrónico que posee una entrada y una salida  
En la entrada se introduce señales alternas de diferentes frecuencias y en la salida se extraen señales atenuadas en mayor o menor medida según la frecuencia de la señal.



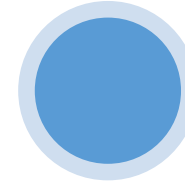
# Filtros



Activos  
Pasivos



Analógicos  
Digitales

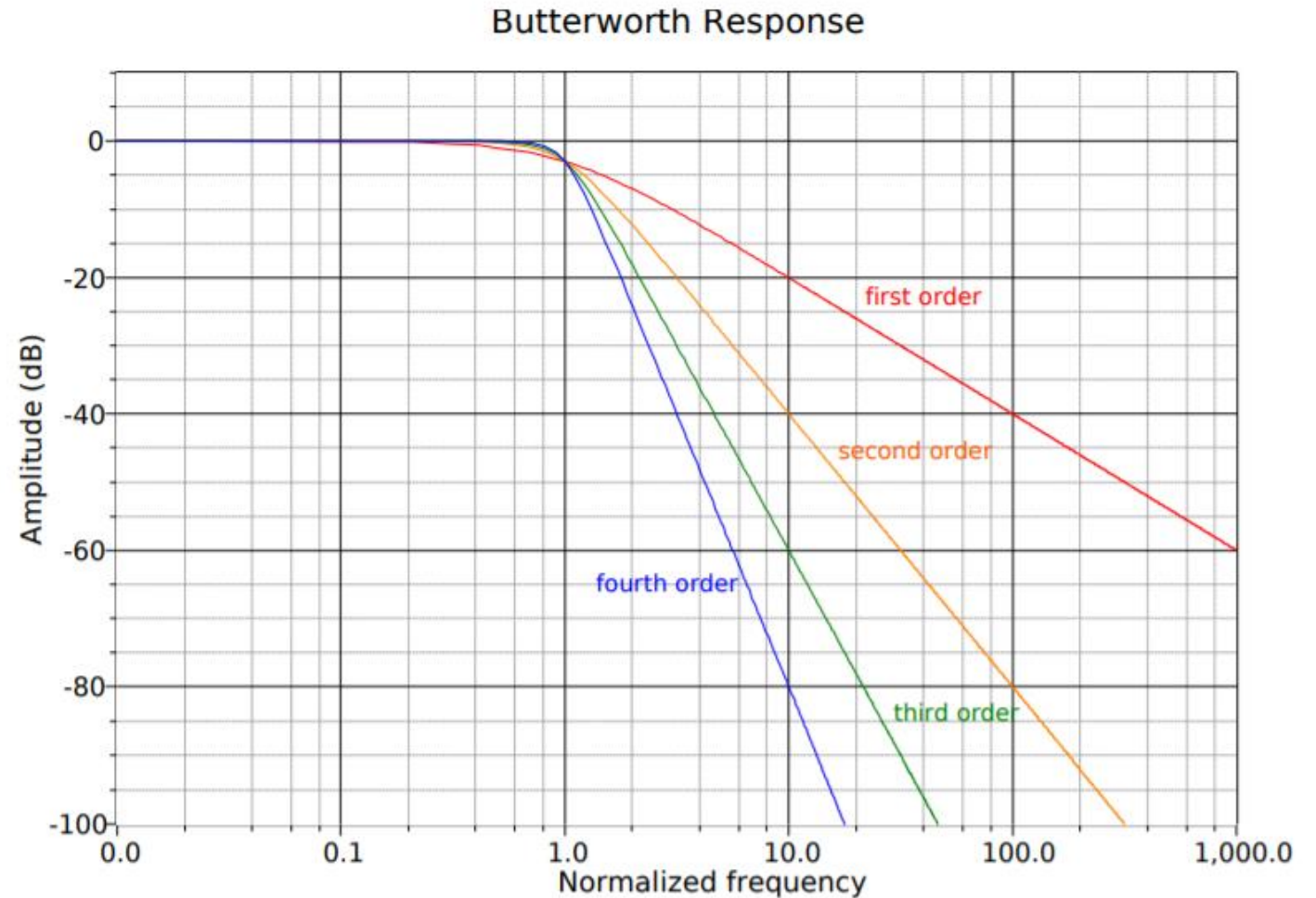


Filtro de paso bajo  
Filtro de paso alto  
Filtro de paso  
banda  
Filtro de banda  
eliminada  
Filtro multibanda  
Filtro variable

# Filtros - Orden

Un filtro es un circuito con al menos un elemento reactivo. Un circuito con solo un elemento reactivo es un “filtro de primer orden”, si el circuito tiene dos elementos reactivos es un “filtro de segundo orden”...

La diferencia que existe entre un filtro de primer orden y un filtro de orden mayor es la curva de respuesta de frecuencia.

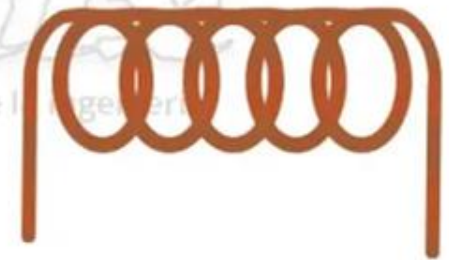


# Filtros - Orden

Resistencia

Capacitor

Bobina



Ingeniería  
Mezclada  
La enciclopedia de la ingeniería

# Filtros - Frecuencia

Se refiere a la cantidad de repeticiones de un fenómeno periódico en una unidad de tiempo.

$$1 \text{ Hz} = \left[ \frac{1}{\text{s}} \right]$$



$$f = 0.5 \text{ Hz}$$
$$T = 2.0 \text{ s}$$



$$f = 1.0 \text{ Hz}$$
$$T = 1.0 \text{ s}$$

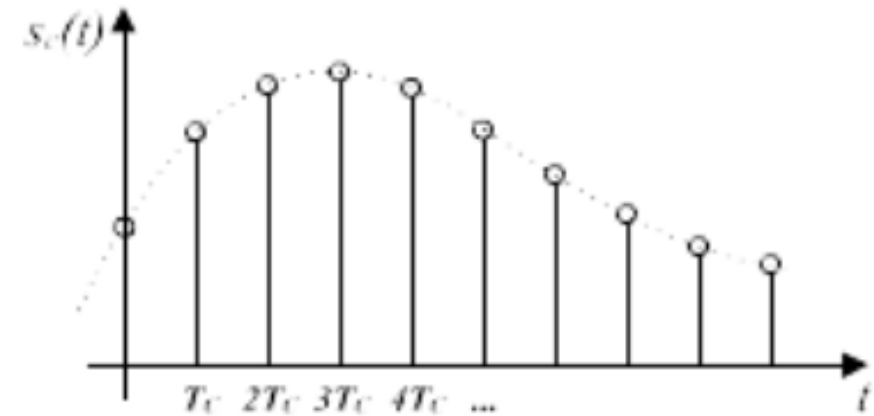
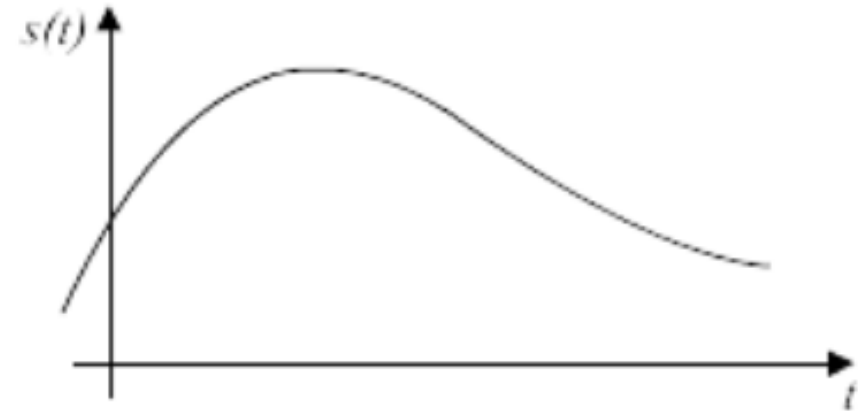


$$f = 2.0 \text{ Hz}$$
$$T = 0.5 \text{ s}$$



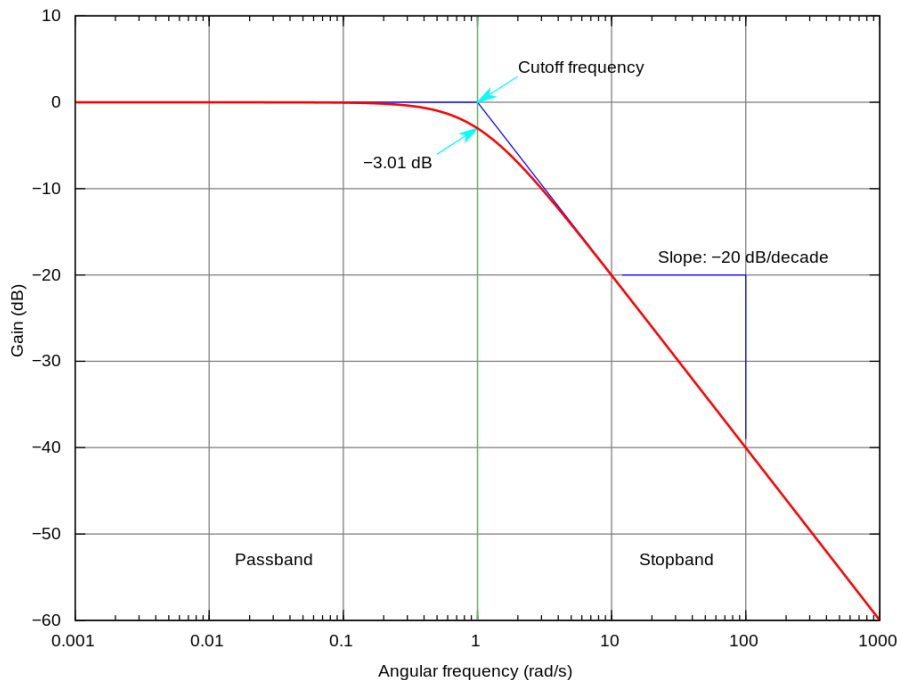
# Filtros – S/s

Es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta,



# Filtros - Frecuencia de corte

Es un límite en la respuesta frecuencial de un sistema en el cual la energía que fluye a través de este se comienza a reducir (atenuar o reflejar) en lugar de pasar a través de él.

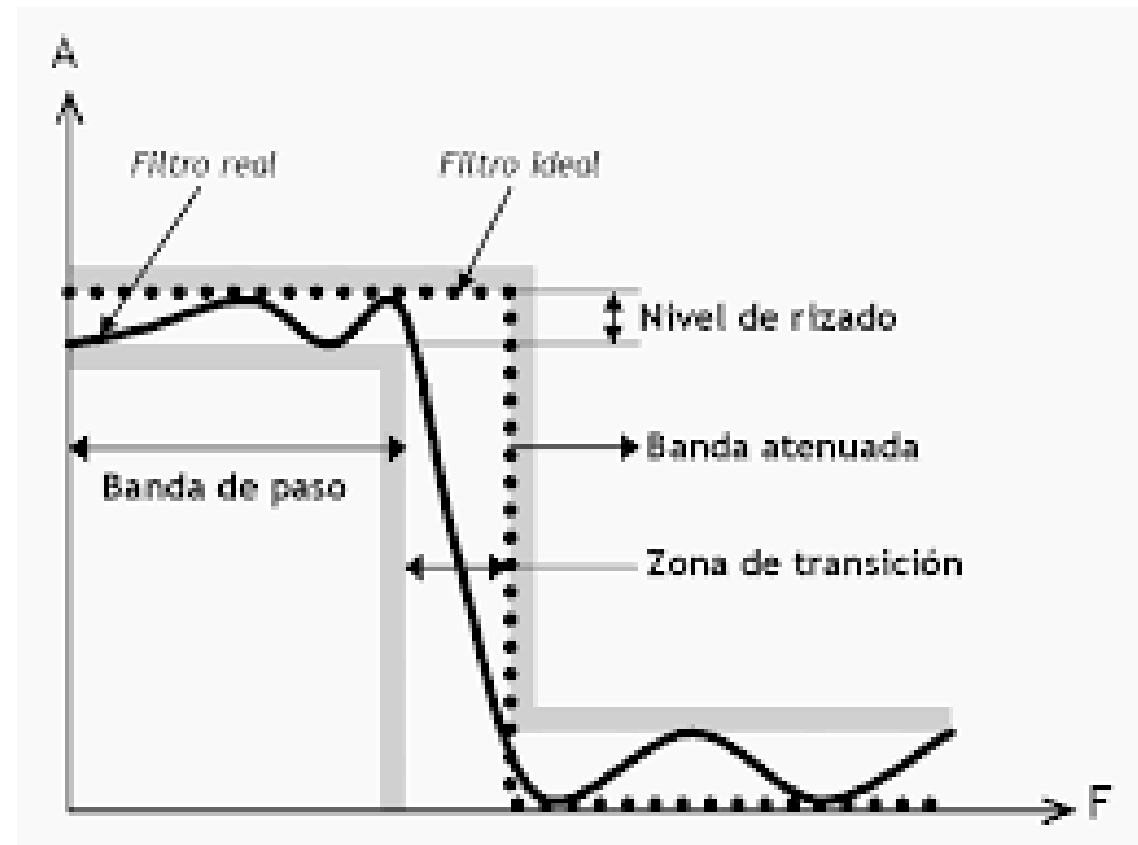


Es la frecuencia para la que la ganancia en tensión del filtro cae de 1 a 0.707 (en decibelios (dB) se diría que la ganancia del filtro se reduce en 3dB)

En los filtros pasa banda y elimina banda existirán dos frecuencias de corte diferentes, la inferior y la superior.

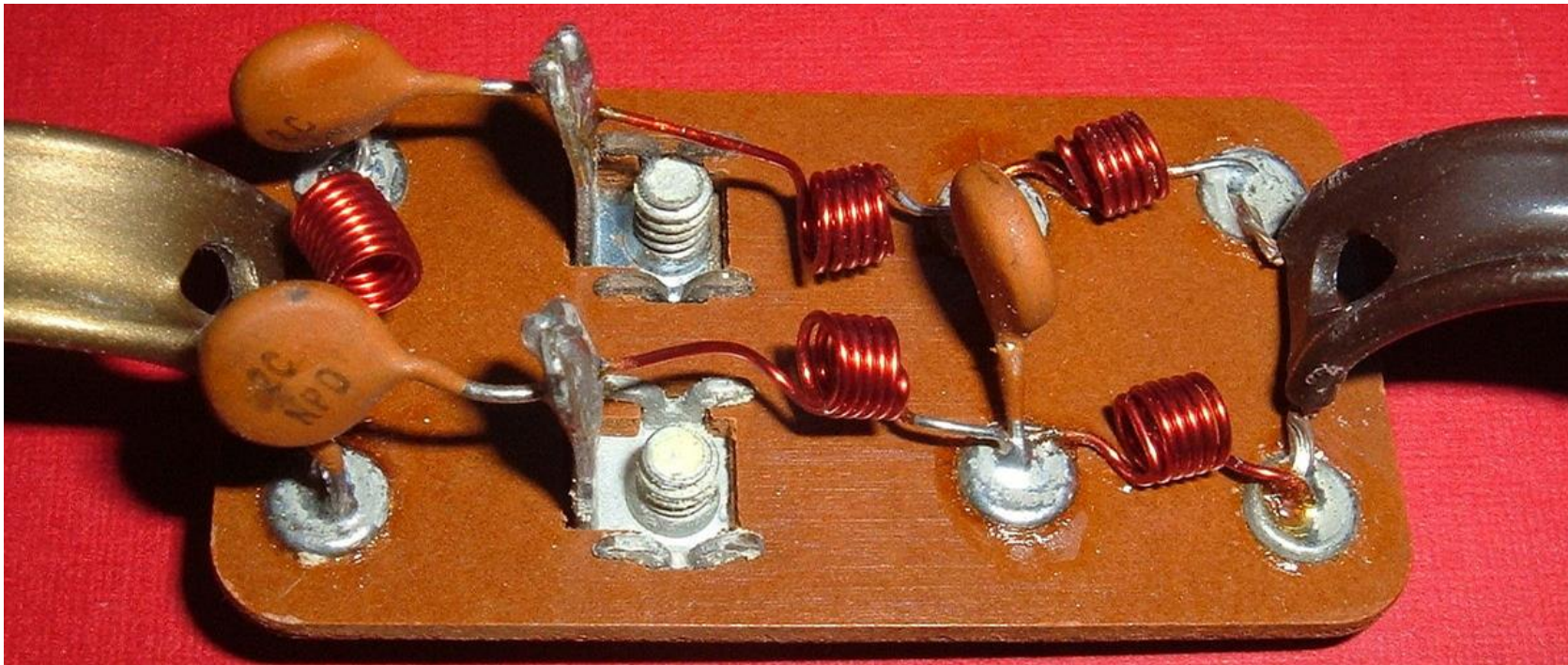
# Filtros - Banda de Paso

Es el margen de frecuencias para las cuales la señal puede pasar

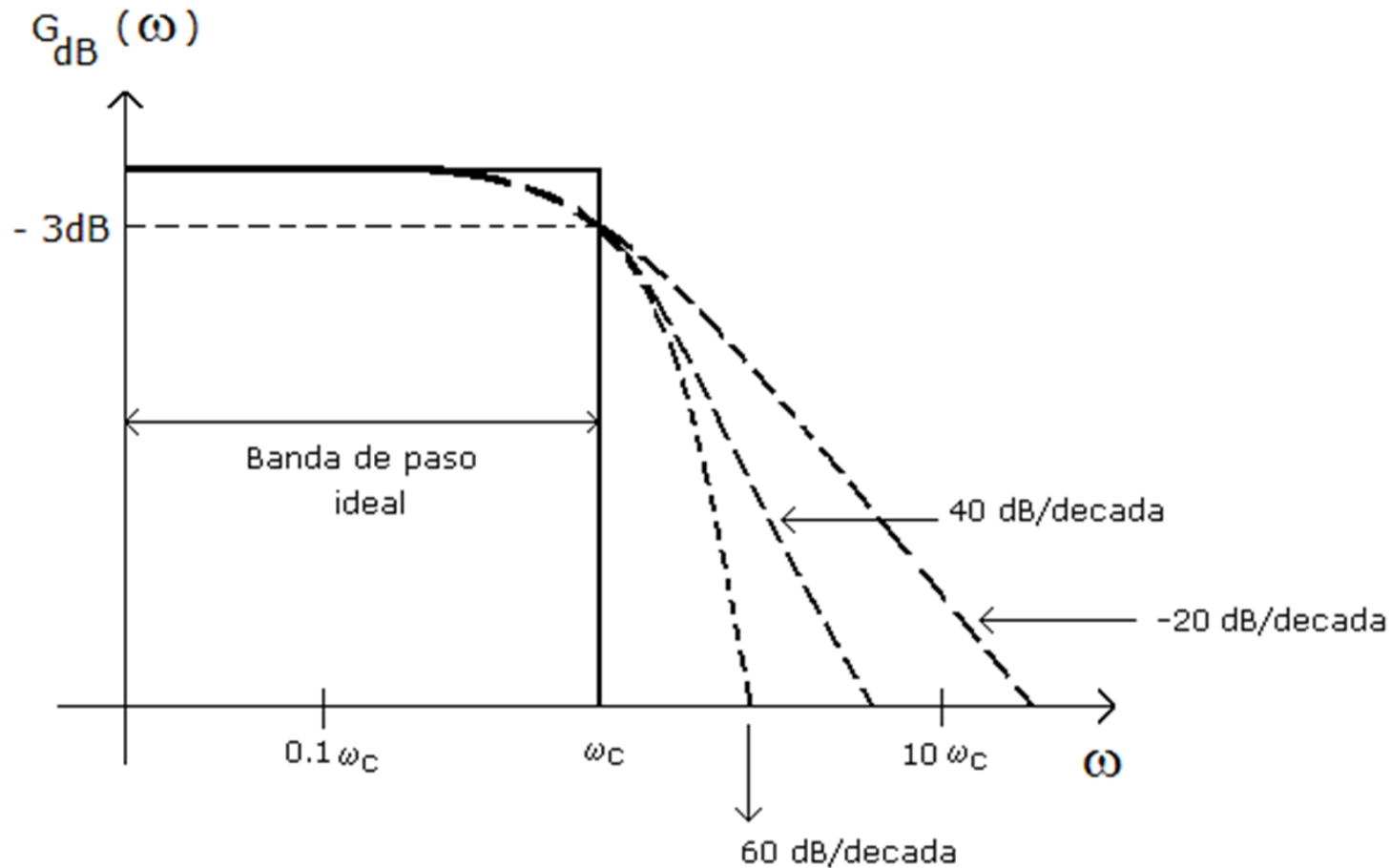


# Filtros - Varios

- **Banda de rechazo:** Rango de frecuencias de un filtro en el cual las frecuencias de una señal no pueden pasar.
- **Década:** Dos frecuencias están separadas una década si una de ellas es de valor diez veces mayor que la otra.
- **Banda atenuada:** Es el rango de frecuencias que el filtro atenúa más de 3dB.



# Filtros - dB



Es una unidad que se utiliza para expresar la relación entre dos valores de presión sonora, o tensión y potencia eléctrica. La unidad básica es el belio (o bel) de símbolo B, pero dada la amplitud de los campos que se miden en la práctica, se utiliza su submúltiplo, el decibelio. El nombre se le ha dado en homenaje a Alexander Graham Bell. Es una expresión que no es lineal, sino logarítmica, adimensional y matemáticamente escalar. Ni el belio, ni el decibelio son unidades del Sistema internacional de unidades.<sup>4</sup>

# Filtros - dB

$$\text{dB} = 10 \log_{10} \frac{P_1}{P_2} \text{ (si lo que se comparan son potencias),}$$

$$\text{dB} = 20 \log_{10} \frac{V_1}{V_2} \text{ (si lo que se comparan son voltajes),}$$

$$\text{dB} = 20 \log_{10} \frac{I_1}{I_2} \text{ (si lo que se comparan son intensidades de corriente).}$$

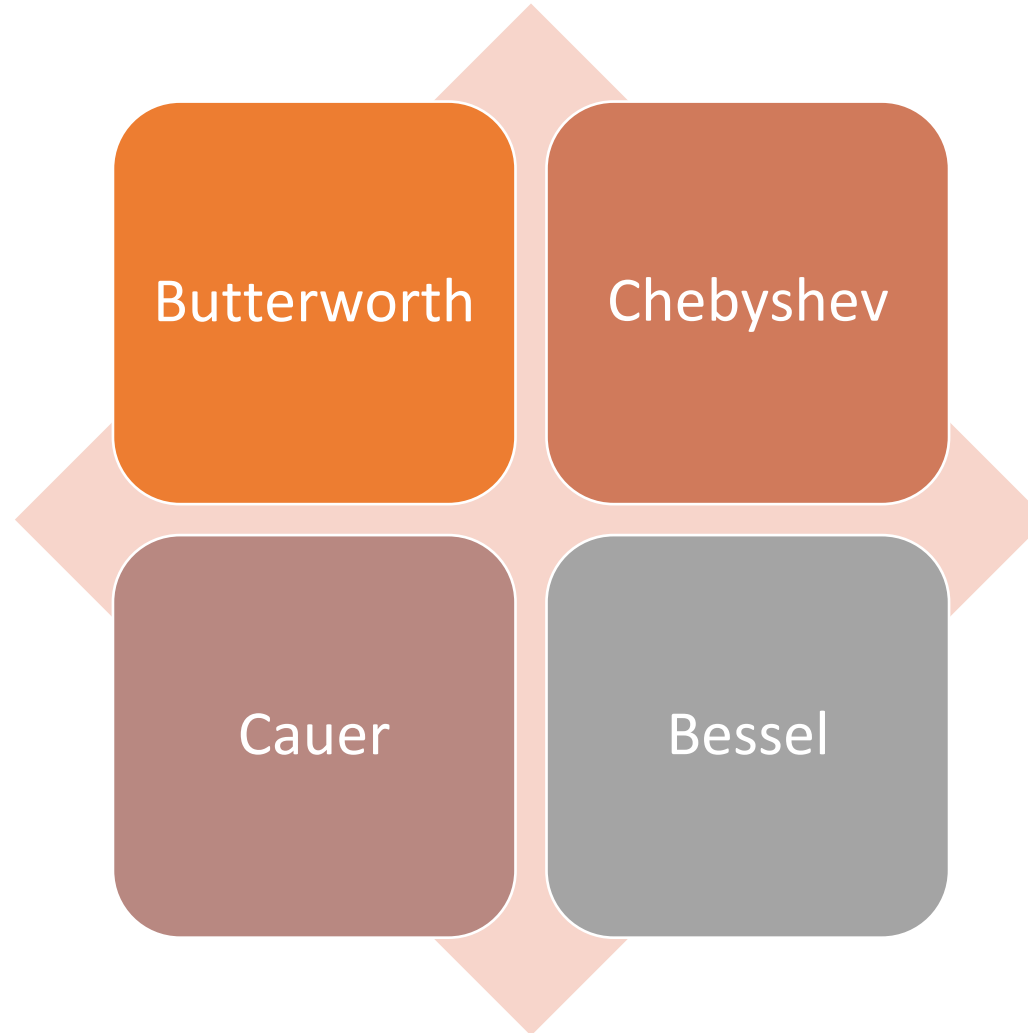
# Filtros - Función de transferencia

Con independencia de la realización concreta del filtro, salvo que debe ser lineal, (analógico, digital o mecánico) su forma de comportarse se describe por su función de transferencia. Esta determina la forma en que la señal aplicada cambia en amplitud y en fase, para cada frecuencia, al atravesar el filtro.

Se puede llegar a expresar matemáticamente la función de transferencia en forma de fracción mediante las transformaciones en frecuencia adecuadas. Se dice que los valores que hacen nulo el numerador son los ceros y los que hacen nulo el denominador son polos.

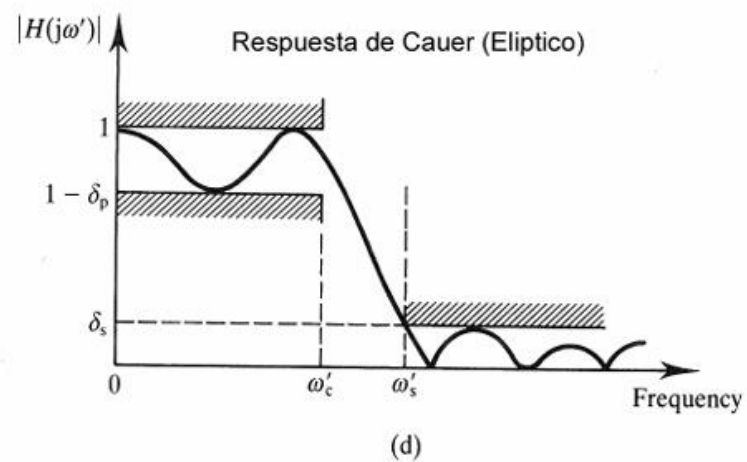
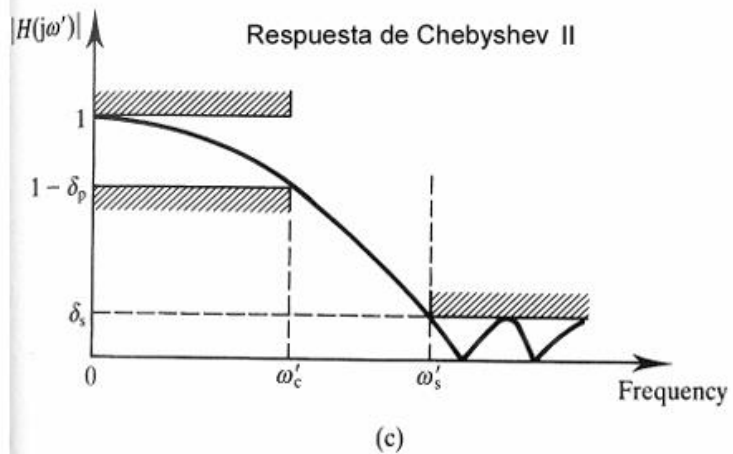
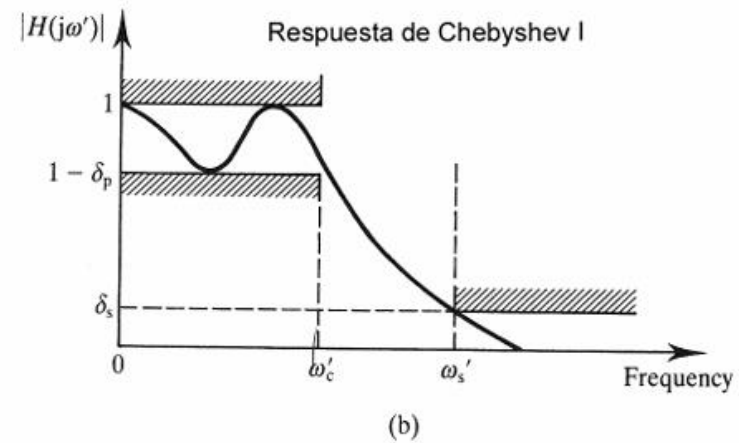
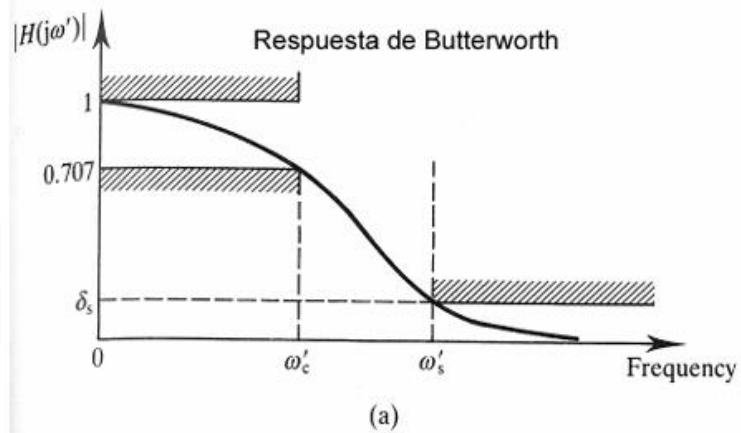
$$H(f) = \frac{\textit{numerador}(f)}{\textit{denominador}(f)}$$

# Filtros - Tipologías

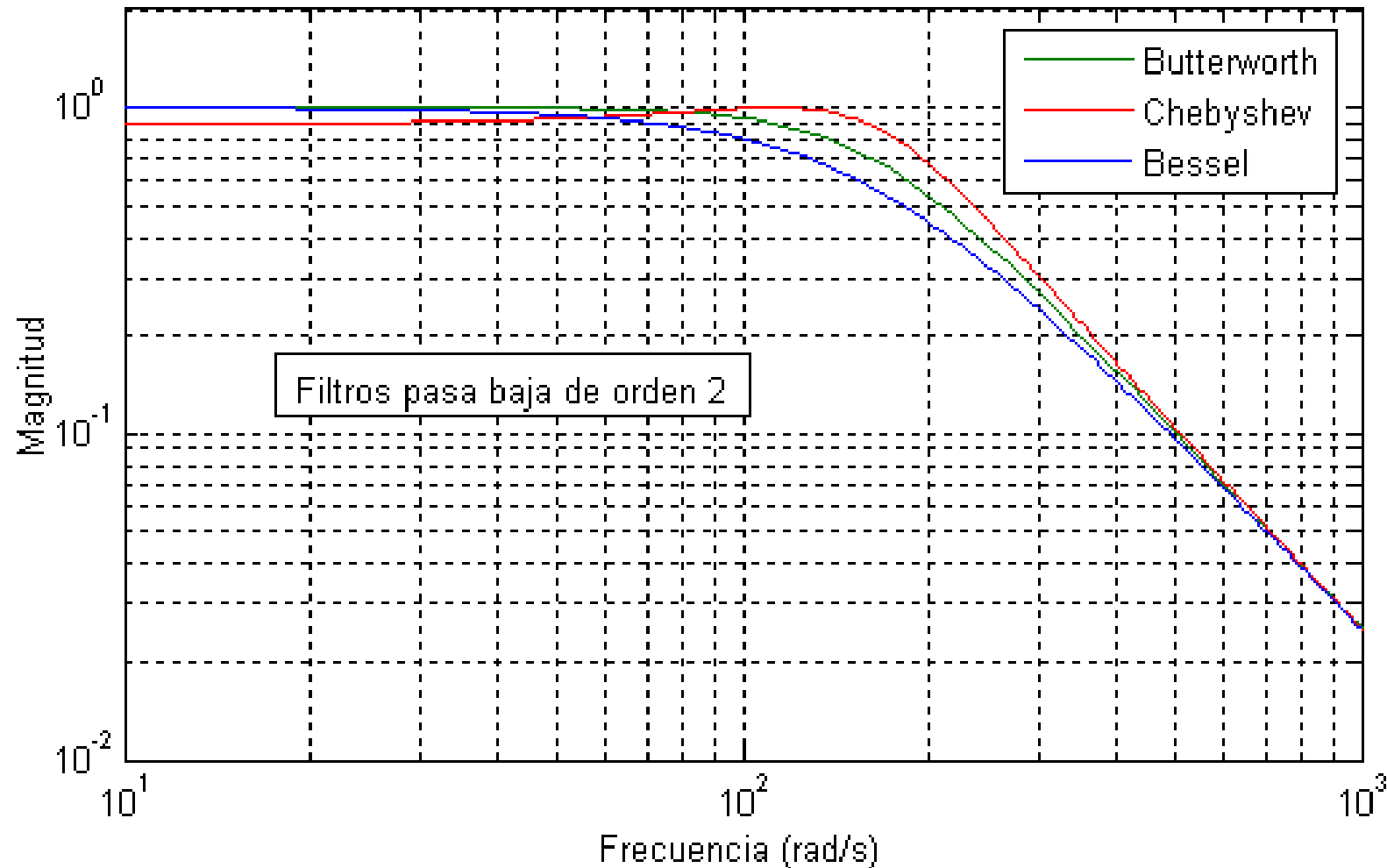




# Filtros - Tipologías

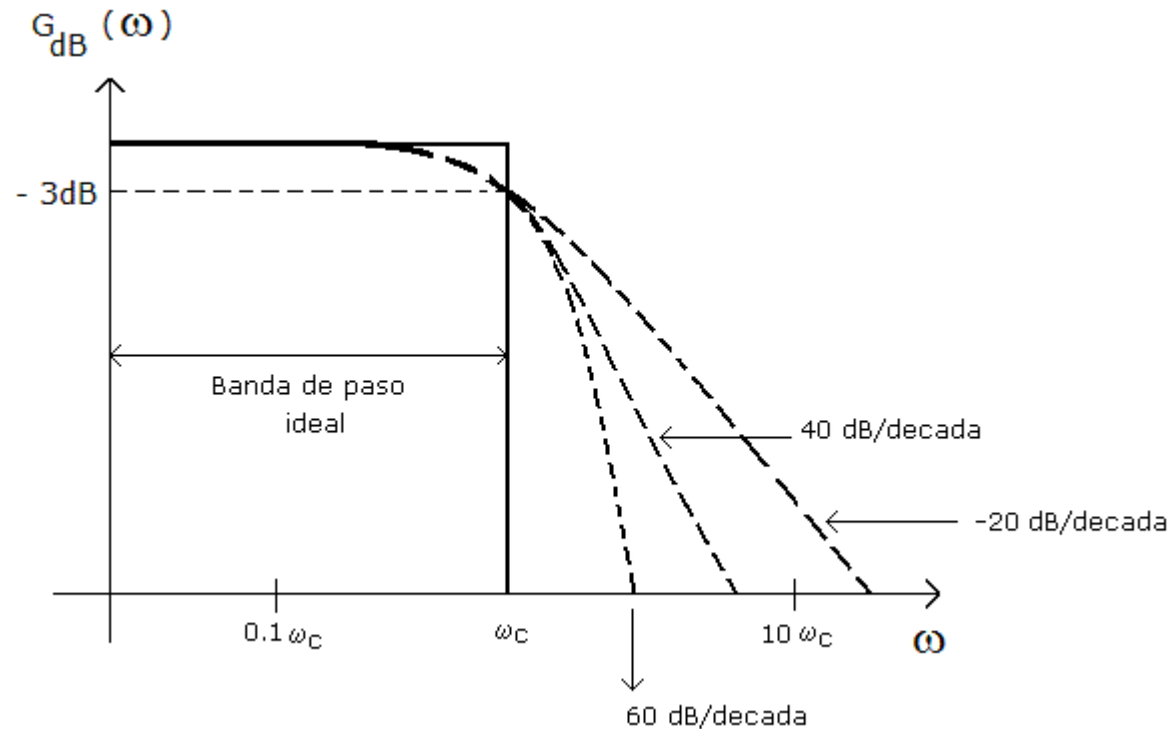


# Filtros – Tipos de filtros



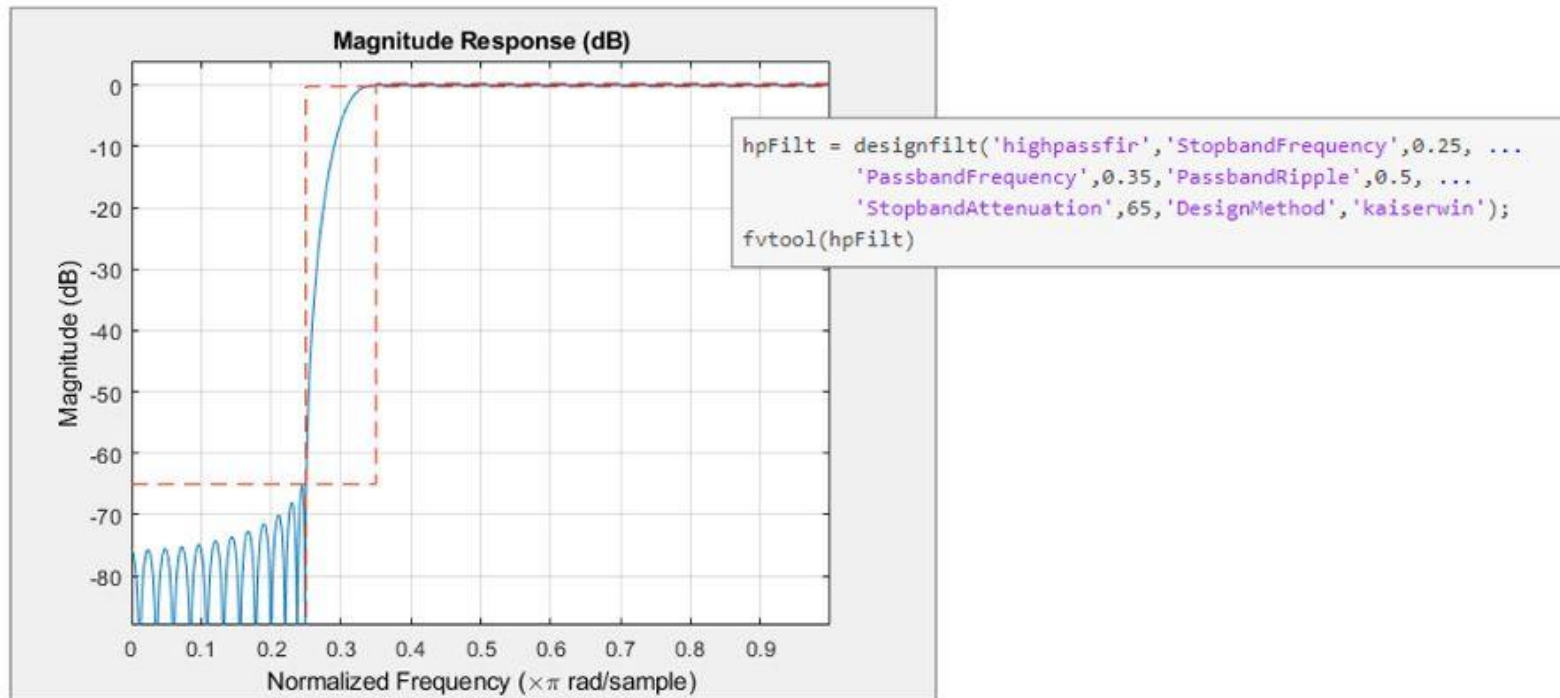
# Filtros – Tipos de filtros

**Filtro pasa bajos:** Son aquellos que introducen muy poca atenuación a las frecuencias que son menores que la frecuencia de corte. Las frecuencias que son mayores que la de corte son atenuadas fuertemente.



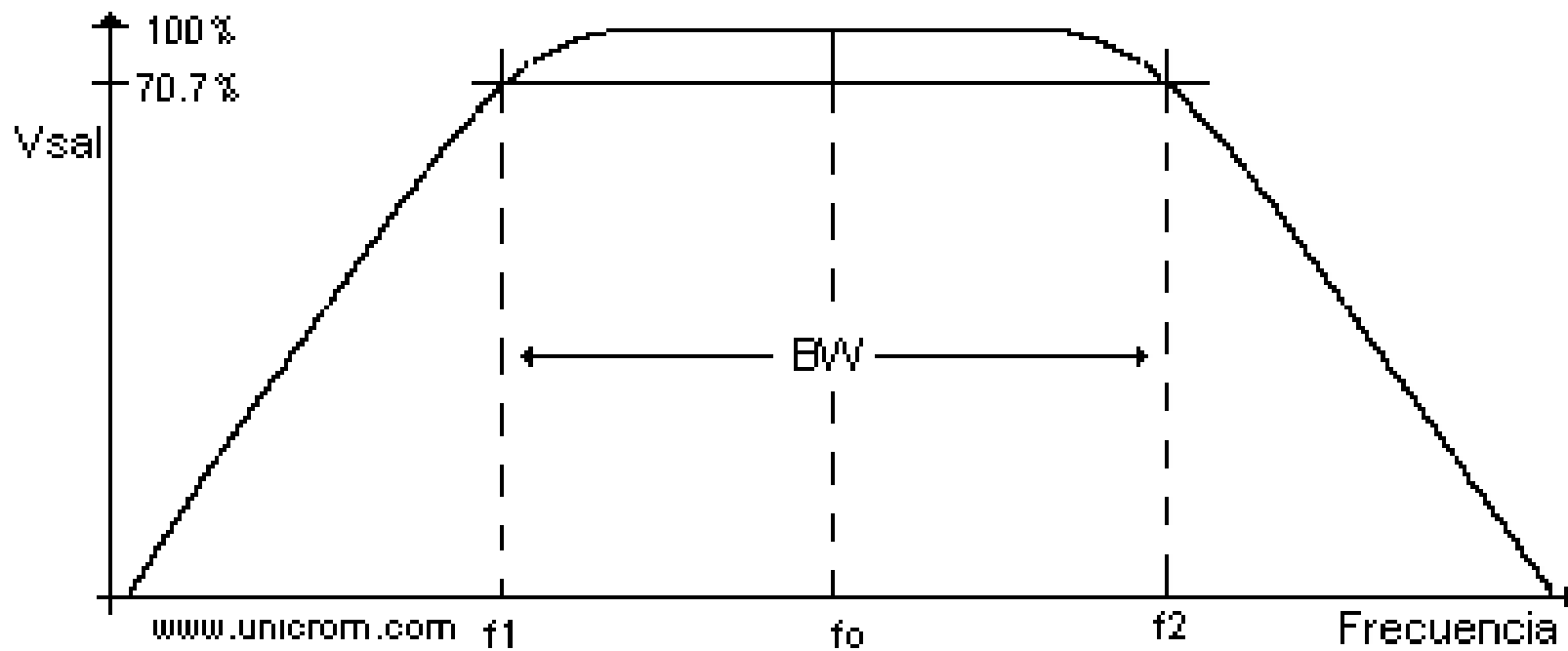
# Filtros – Tipos de filtros

**Filtro pasa altos:** Este tipo de filtro atenúa levemente las frecuencias que son mayores que la frecuencia de corte e introducen mucha atenuación a las que son menores que dicha frecuencia.



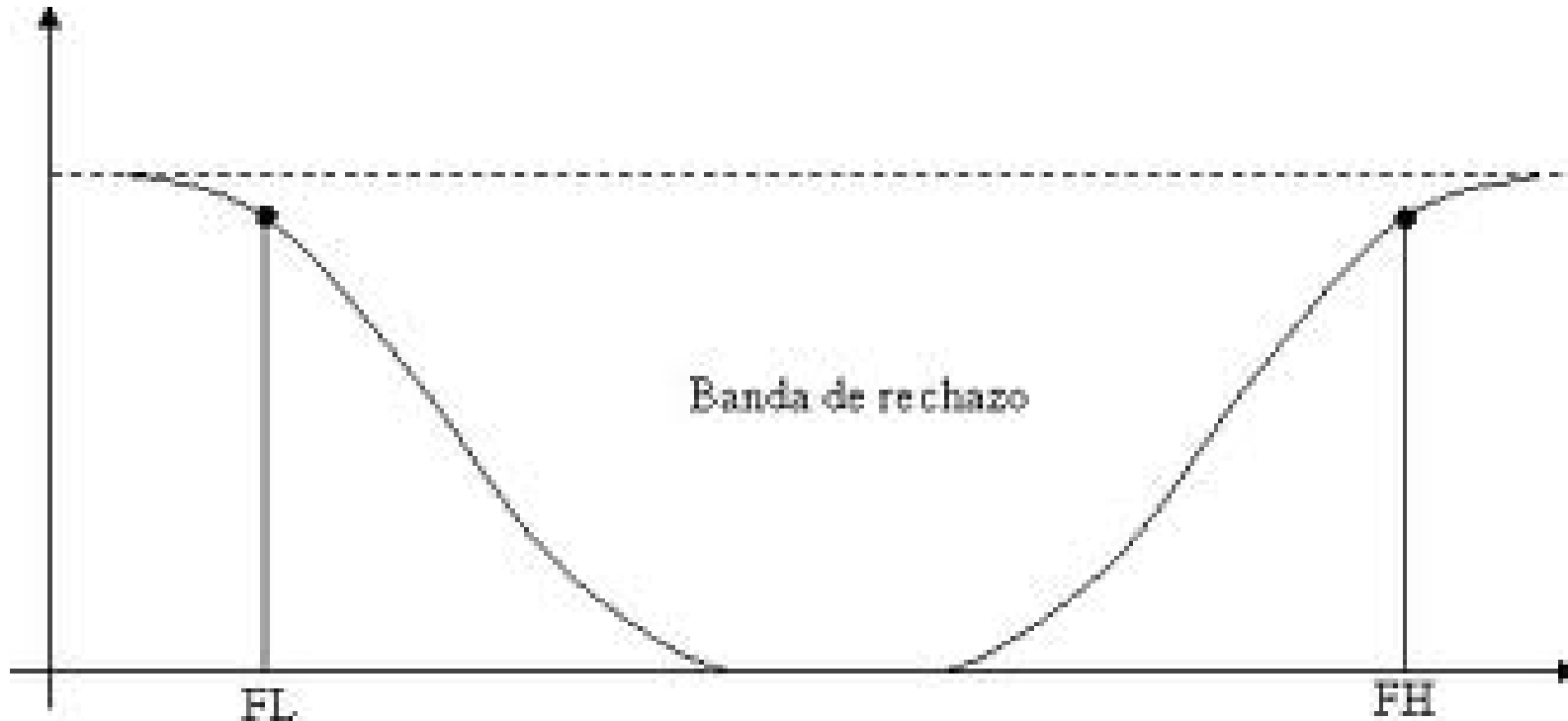
# Filtros – Tipos de filtros

**Filtro pasa banda:** En este filtro existen dos frecuencias de corte, una inferior y otra superior. Este filtro sólo atenúa grandemente las señales cuya frecuencia sea menor que la frecuencia de corte inferior o aquellas de frecuencia superior a la frecuencia de corte superior. por tanto, sólo permiten el paso de un rango o banda de frecuencias sin atenuar.

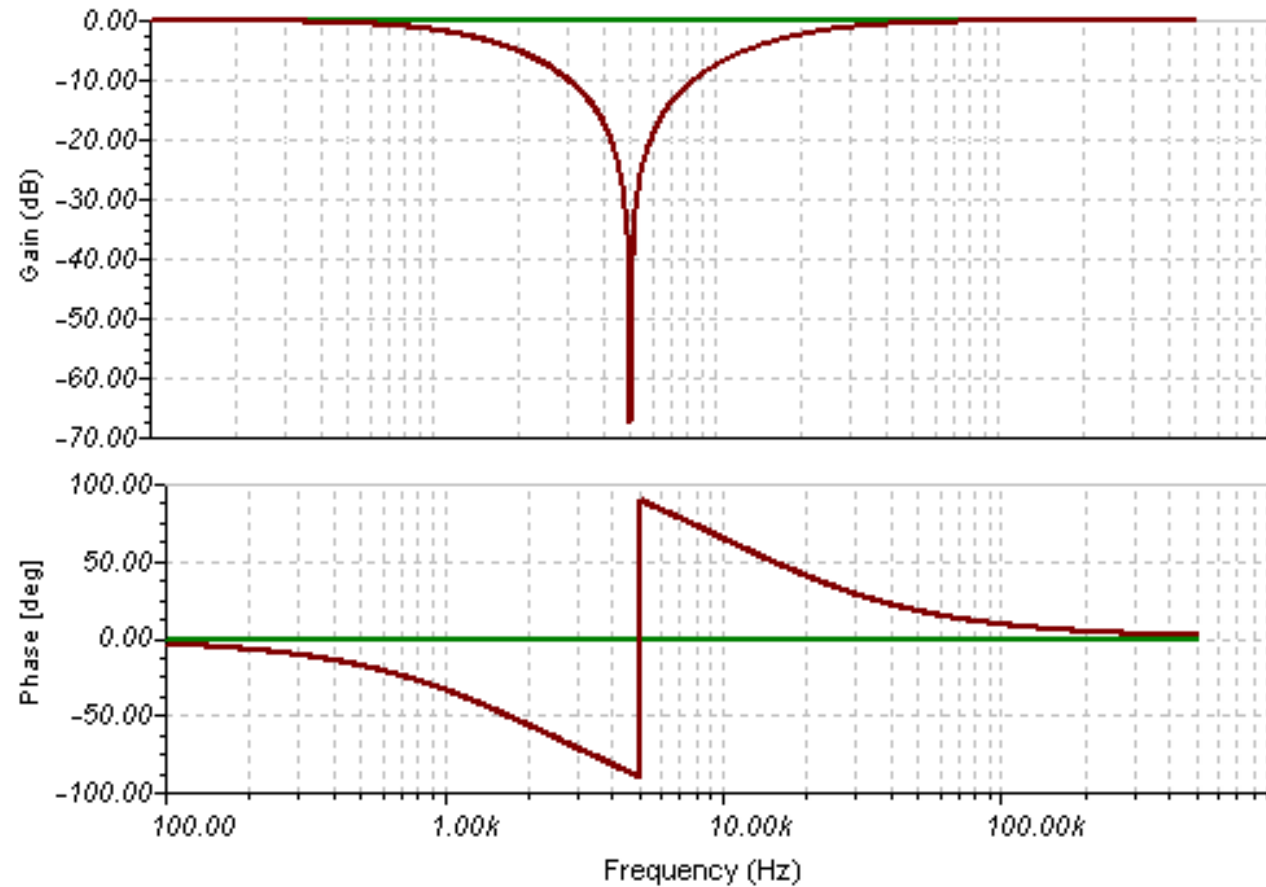


# Filtros – Tipos de filtros

**Filtro elimina banda:** Este filtro elimina en su salida todas las señales que tengan una frecuencia comprendida entre una frecuencia de corte inferior y otra de corte superior. Por tanto, estos filtros eliminan una banda completa de frecuencias de las introducidas en su entrada



# Filtros – Tipos de filtros






# Filtros – Tipos de filtros

## Filtro eléctrico:

- Elemento que discrimina una determinada frecuencia o gama de frecuencias de una señal eléctrica que pasa a través de él, pudiendo modificar tanto su amplitud como su fase

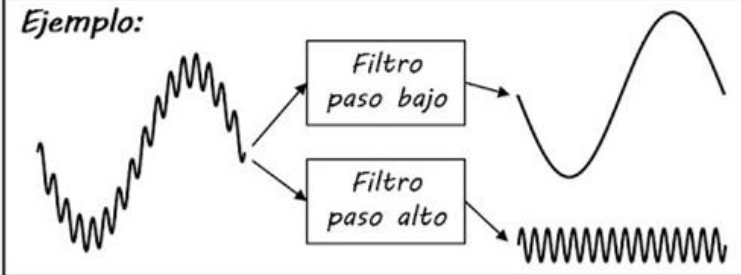
## Señal eléctrica a distintas frecuencias:

- Frecuencia  $f < f_{ci}$  
- Frecuencia  $f_{ci} < f < f_{cs}$  
- Frecuencias  $f_{cs} < f$  

## Clases previas recomendadas:

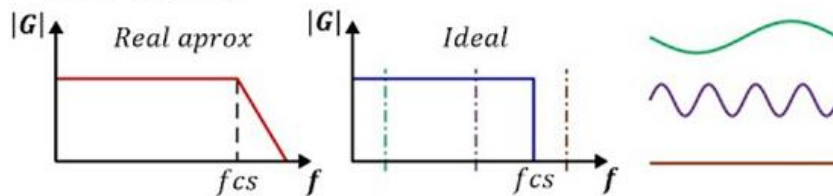
- Diagramas de bode
- Amplificadores operacionales (Realimentación negativa)

## Ejemplo:

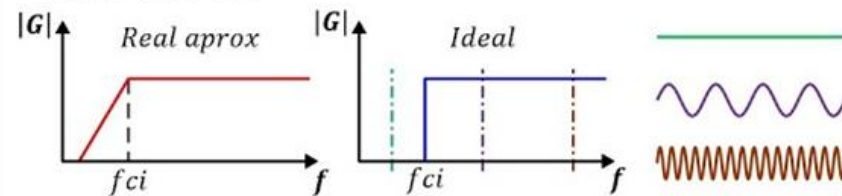


## Filtros en función de la respuesta en frecuencia:

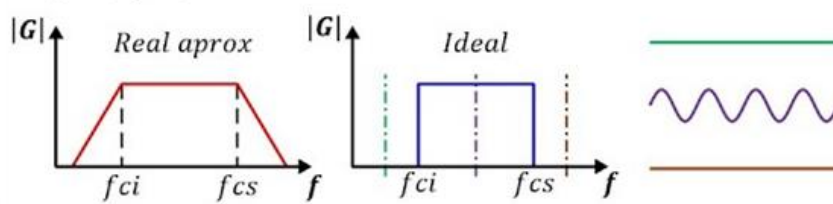
### - Filtro paso bajo



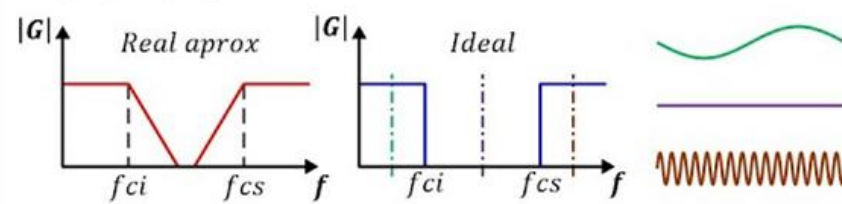
### - Filtro paso alto



### - Filtro paso banda



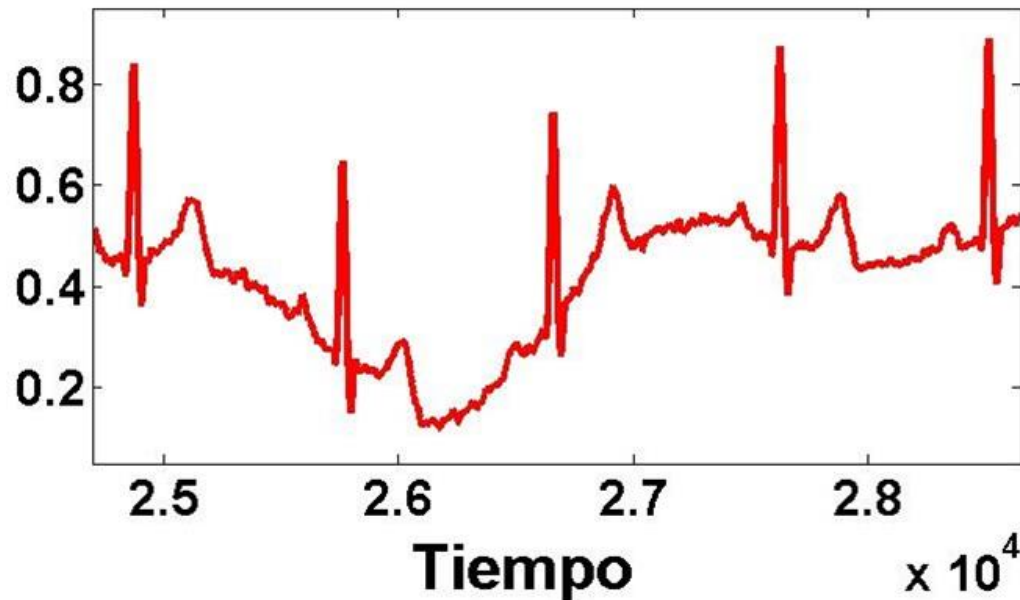
### - Filtro Notch



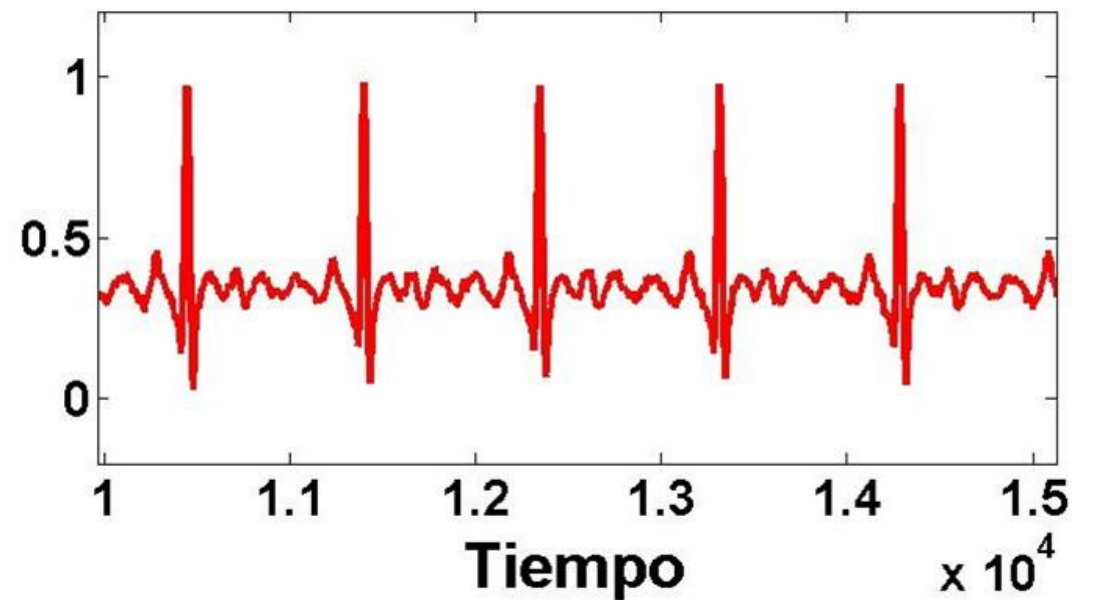


# Filtros – Tipos de filtros

1. ECG Original



2. ECG Filtrado por FFT

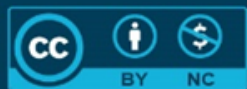


<https://pysdr.org/es/content-es/filters.html>

<https://es.mathworks.com/help/signal/filter-design.html>



Esta licencia permite a otros distribuir, remezclar, retocar, y crear a partir de esta obra de manera no comercial y, a pesar que sus nuevas obras deben siempre mencionar a la IU Digital y mantenerse sin fines comerciales, no están obligados a licenciar obras derivadas bajo las mismas condiciones.



# ¡GRACIAS

**IU**Digital  
de Antioquia  
INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA