

**EC1281**  
**LABORATORIO DE MEDICIONES ELÉCTRICAS**  
**PRELABORATORIO N° 2**  
**PRÁCTICA N° 3: SIMULACIÓN DE CIRCUITOS**

**¿QUÉ ES SPICE?**

Es un programa para simular circuitos que incluyen tanto componentes básicos como circuitos electrónicos integrados, desarrollado como una tesis doctoral en la Universidad de Berkeley en 1973.

La codificación inicial se realizó en Fortran.

Los datos e instrucciones se introducían en el sistema mediante una lista organizada, denominada "Netlist".

El cálculo de las variables del circuito se realizaba aplicando el método de nodos.

Por ser el resultado de un desarrollo académico, el programa SPICE básico es un programa de libre distribución.

## ¿CÓMO EVOLUCIONÓ SPICE?

A lo largo de los años se fué mejorando la forma de realizar los cálculos, en 1989 se codifica en C, se desarrolla una interfaz gráfica y se continúan agregando numerosas funciones.

Posteriormente las compañías que desarrollan software tomaron el núcleo inicial del programa, y elaboraron alrededor del mismo interfaces de entrada y salida que simplifican enormemente el uso del programa, pero que convierten el producto final en un software comercial, que debe comprarse: ISPICE, HSPICE, PSPICE...

En el laboratorio tenemos PSPICE Evaluation 9.1.

En la actualidad AIMSPICE sigue siendo de libre distribución, pero la entrada de datos es a través de "Netlist"

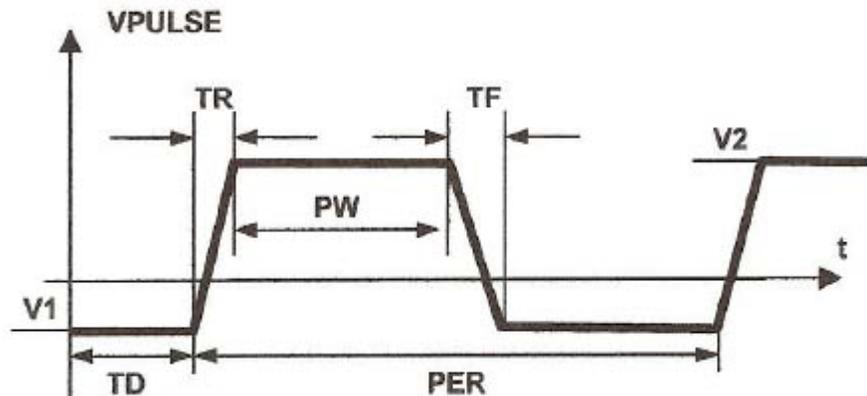
**El nombre SPICE es un acrónimo:**

**Simulation Program with Integrated Circuit Emphasis**

## **PASOS A SEGUIR PARA SIMULAR UN CIRCUITO EN SPICE**

- \* Abrir el programa y seleccionar Schematics. Aparece una página donde se va a dibujar el circuito.
- \* Seleccionar los componentes que se desean y colocarlos en la página.
- \* Ubicar los componentes en la posición deseada.
- \* Definir los parámetros de los componentes en las correspondientes cajas de diálogo (valor de las resistencias, voltaje de las fuentes, etc.)
- \* Conectar los componentes para determinar el circuito. No deben quedar componentes sin conexión y debe identificarse un punto como tierra GND.
- \* Seleccionar las variables que se quieren como resultados (voltaje en una resistencia, corriente por una rama, etc.)
- \* Seleccionar el tipo de análisis que se quiere realizar. Nosotros vamos a utilizar Bias Point detail, Transient y AC Sweep. Cuando se abre la ventana hay que seleccionar los parámetros de la simulación correspondiente.
- \* Guardar el archivo con un nombre que lo identifique.
- \* Correr la simulación. Los resultados aparecen en una gráfica.

## EJEMPLO: LA FUENTE VPULSE EN SPICE



Descripción del generador VPULSE en *PSpice*

V1 = Voltaje inferior (0V)

V2 = Voltaje superior (10V)

PER = Período (unas 16 veces la constante de tiempo esperada)

PW = Ancho del pulso (unas 8 veces la constante de tiempo esperada)

TD = Tiempo de retardo (0 s)

TR y TF = Tiempos de subida y bajada (se puede colocar 1ns en cada parámetro para que la forma de onda no sea totalmente vertical, lo cual puede crear problemas de conmutación)

## VALORES PARA LOS CIRCUITOS

Circuito 3.2.a:  $V_p = 5 \text{ V}$ ;  $f = 1 \text{ KHz}$ ;  $R = 1 \text{ K}\Omega$ ,  $C = 100 \text{ nF}$

Circuito 3.2.b:  $V_p = 5 \text{ V}$ ;  $f = 1 \text{ KHz}$ ;  $R = 1 \text{ K}\Omega$ ,  $L = 100 \text{ mH}$

Circuito 3.3:  $V_p = 1 \text{ V}$ ;  $R = 1 \text{ K}\Omega$ ,  $C = 100 \text{ nF}$ ;  $L = 100 \text{ mH}$

Circuito 3.4:  $R_1 = 1 \text{ K}\Omega$ ;  $R_2 = 10 \text{ K}\Omega$ ;  $R_3 = 910 \Omega$ ; Fuentes  $15\text{V}$  y  $-15\text{V}$ .

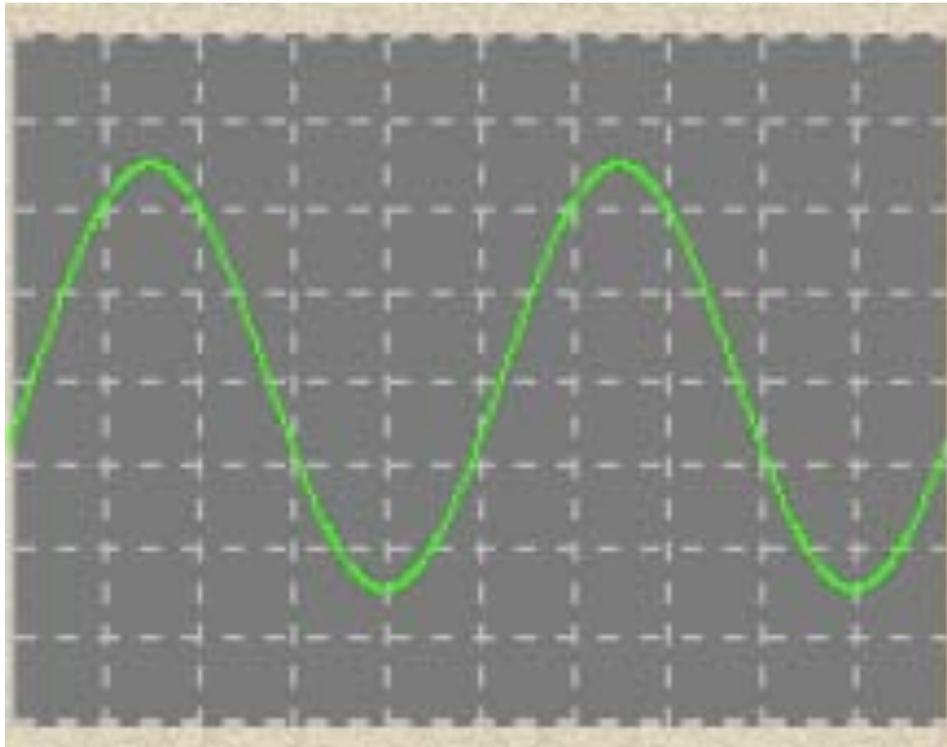
Circuito 3.5:  $R_1 = 1 \text{ K}\Omega$ ;  $R_2 = 10 \text{ K}\Omega$ ;  $R_3 = 910 \Omega$ ;  $C = 10 \text{ nF}$ ,  $15\text{V}$  y  $-15\text{V}$ .

Rango de frecuencias para los análisis AC Sweep:  $10 \text{ Hz}$  a  $1 \text{ MHz}$ .

## **CRONOGRAMA DE TRABAJO PARA LA PRÁCTICA N° 3**

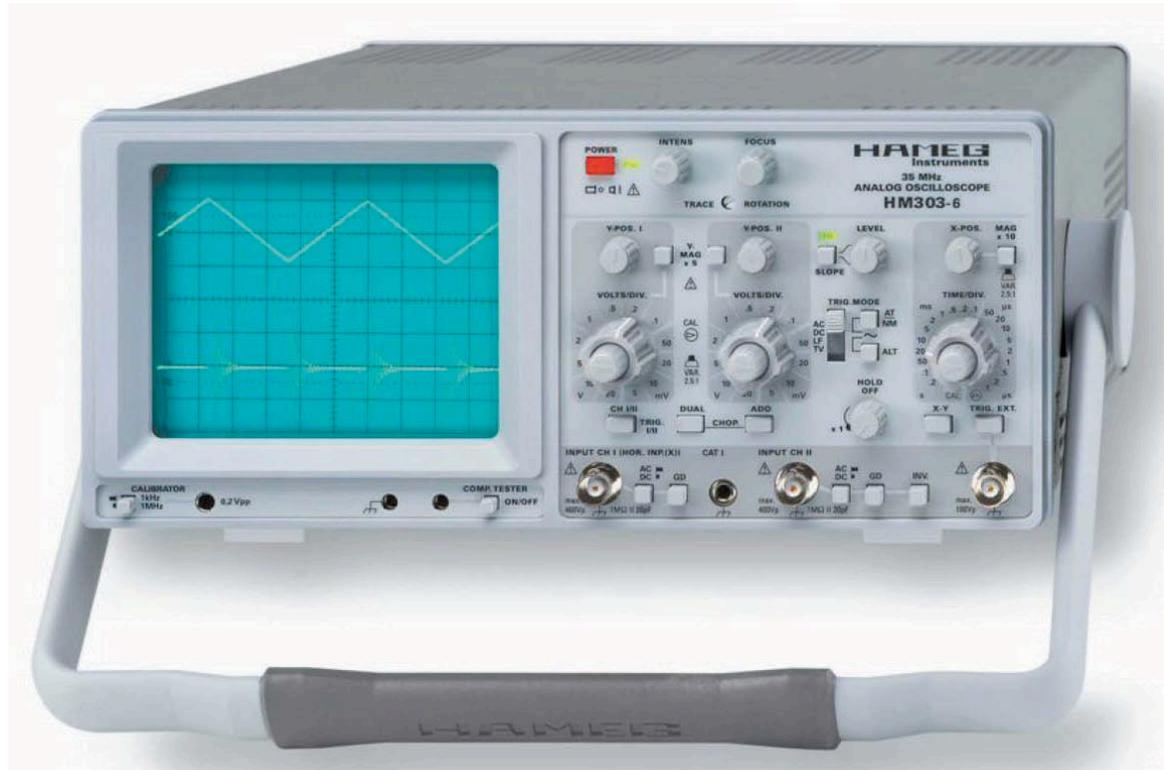
Familiarización con el programa SPICE y ejecución de la simulación de los primeros circuitos	20 minutos
Simulación de los circuitos RC y RL	30 minutos
Simulación de las configuraciones con el circuito RLC	60 minutos
Simulaciones con el amplificador operacional	60 minutos

# **PRÁCTICA N° 4 EL OSCILOSCOPIO**

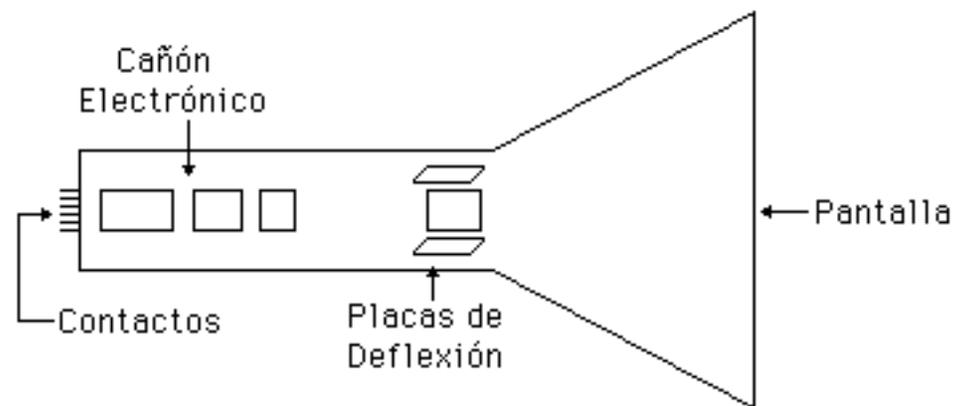
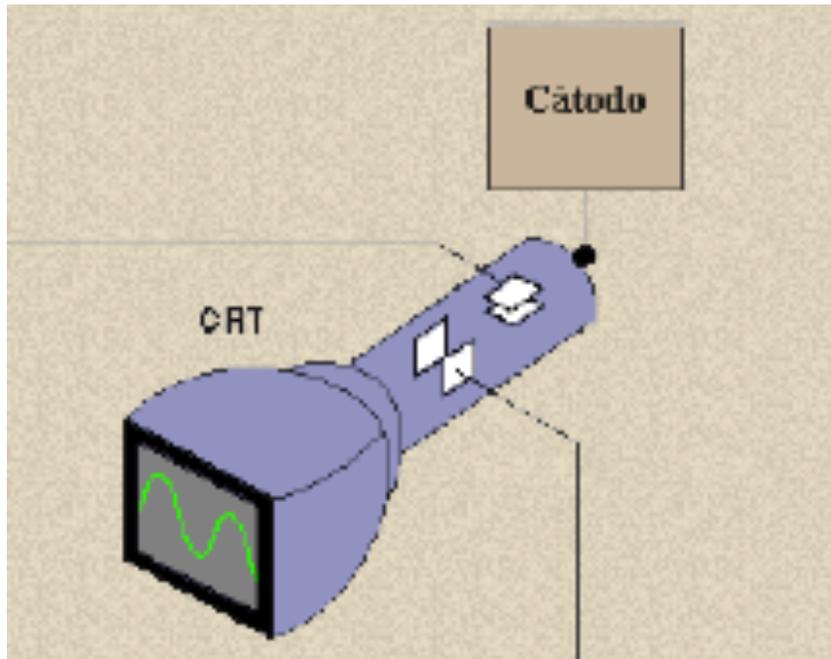


**Señal sinusoidal en la pantalla de un osciloscopio**

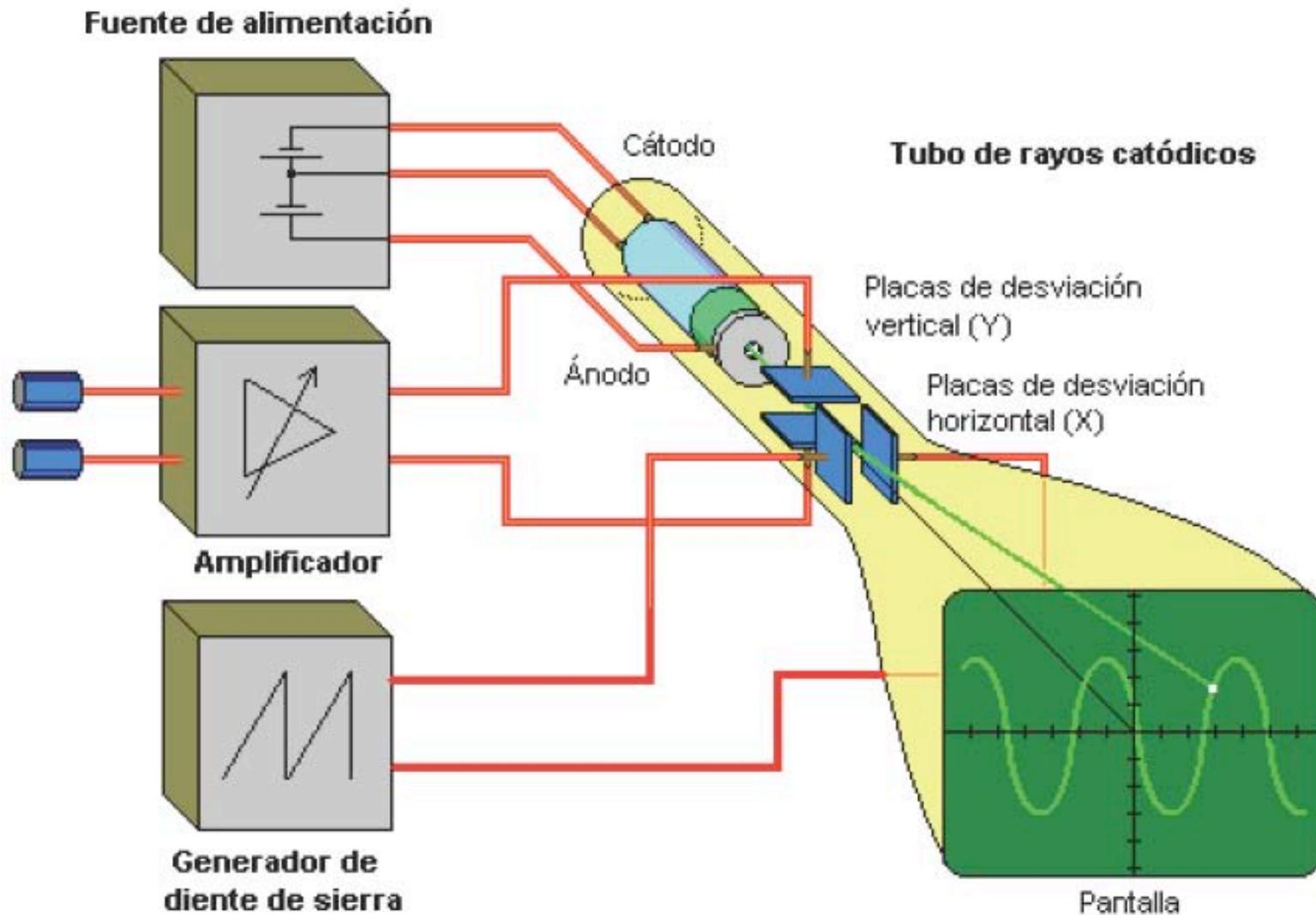
# OSCILOSCOPIO ANALÓGICO



# TUBO DE RAYOS CATÓDICOS

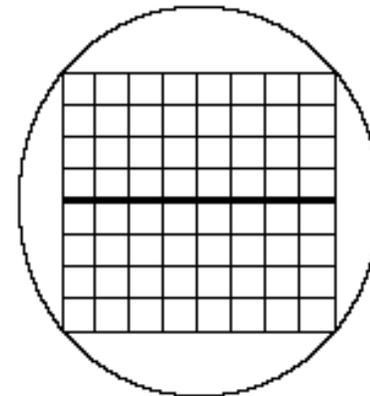
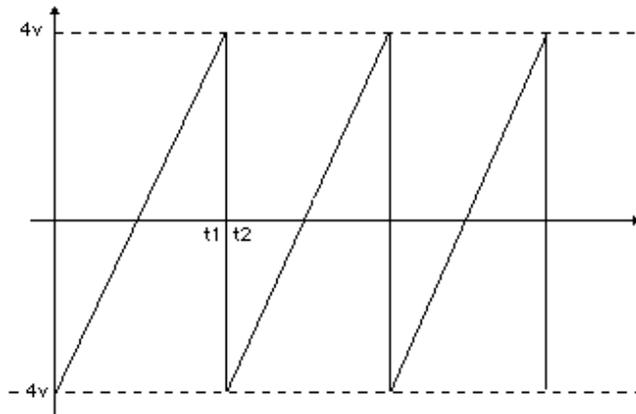


# DIAGRAMA DE BLOQUES DE UN OSCILOSCOPIO ANALÓGICO

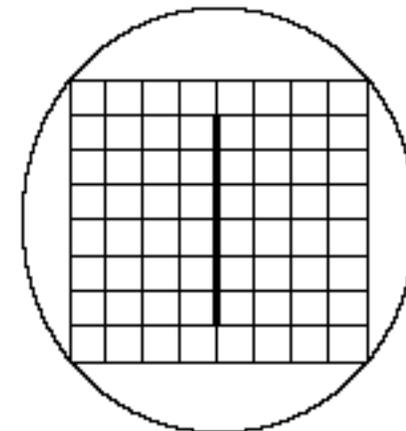
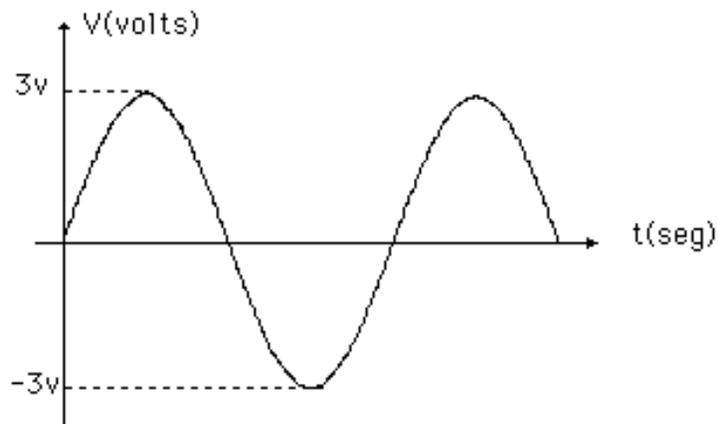


# PRESENTACIÓN DE LAS FIGURAS EN LA PANTALLA DE UN OSCILOSCOPIO ANALÓGICO

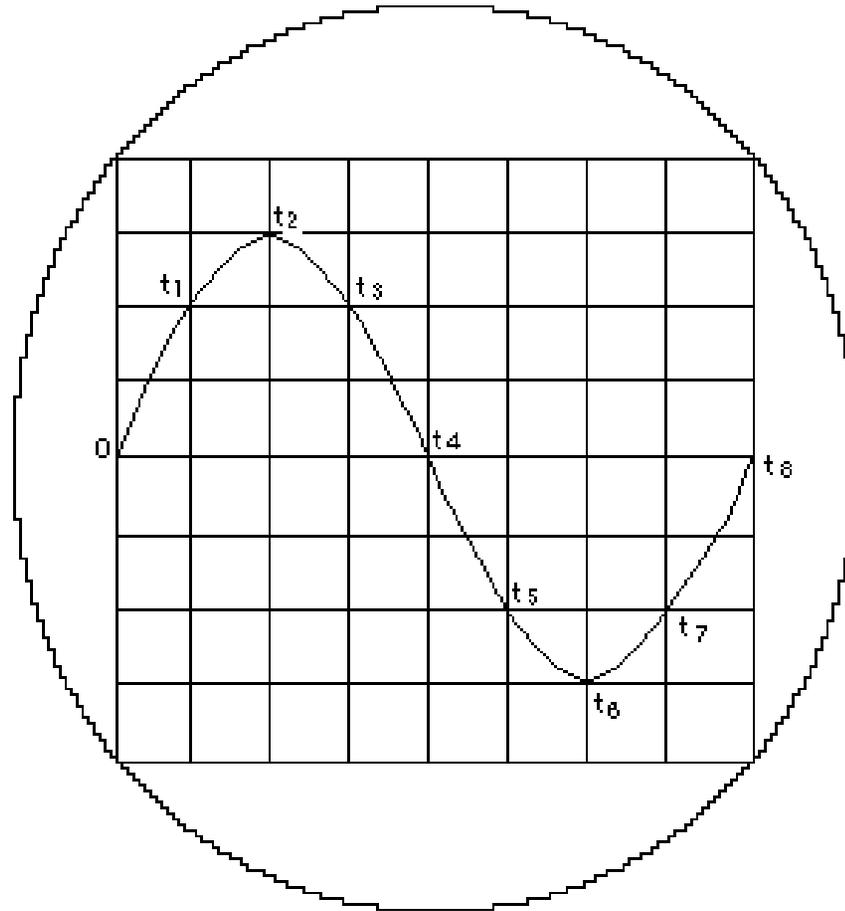
## Aplicación de la diente de sierra a las placas de deflexión horizontal



## Aplicación de una señal sinusoidal a las placas de deflexión vertical

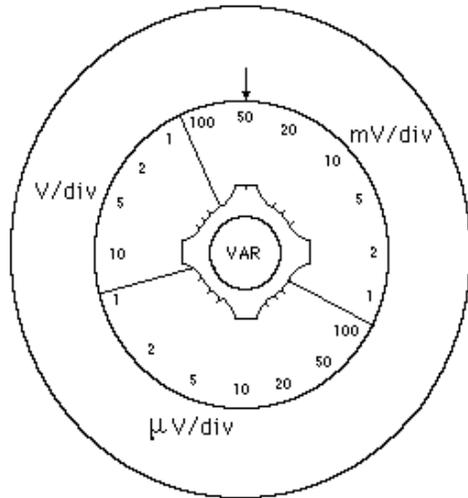


## Señales aplicadas simultáneamente a la placas de deflexión

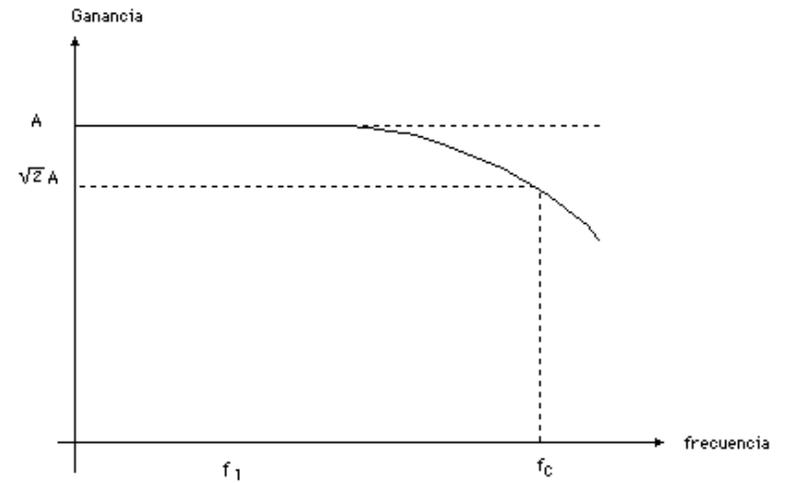


# Amplificador vertical

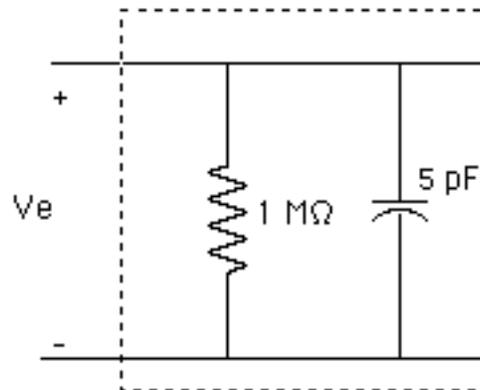
Perilla de calibración



Respuesta en frecuencia



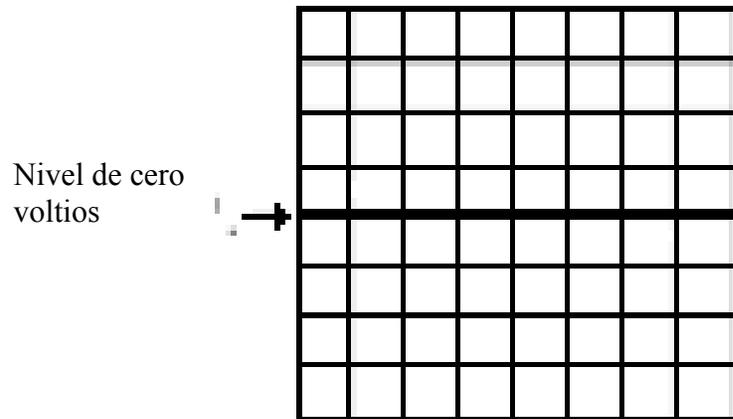
Impedancia de entrada



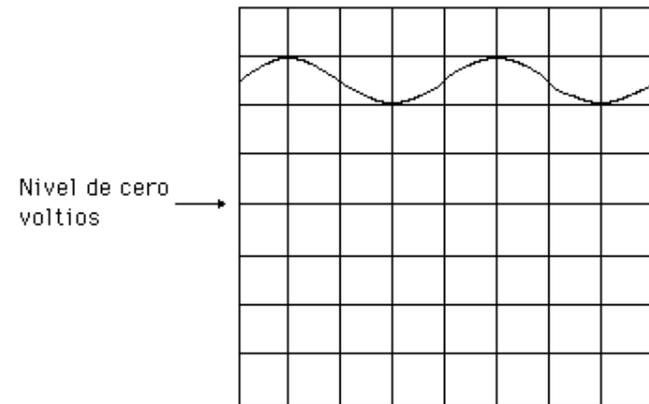
# Amplificador vertical : Acoplamiento de la señal de entrada

$$f(t) = 2,5V + 0,5V(\text{sen } \omega t)$$

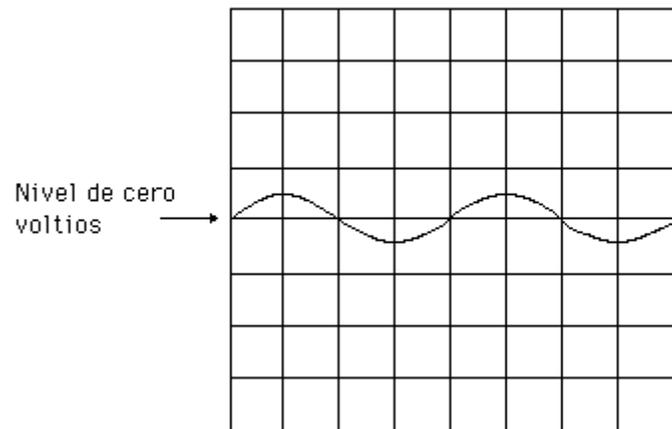
Acoplamiento GND



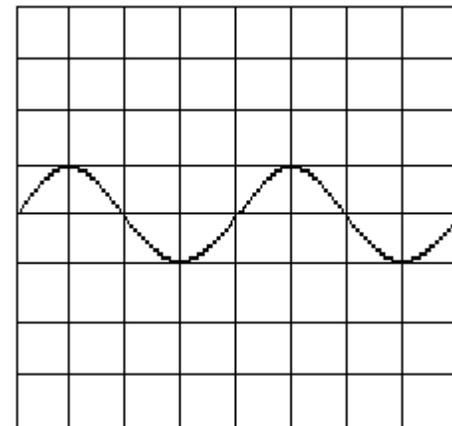
Acoplamiento DC, 1V/div



Acoplamiento AC, 1V/div

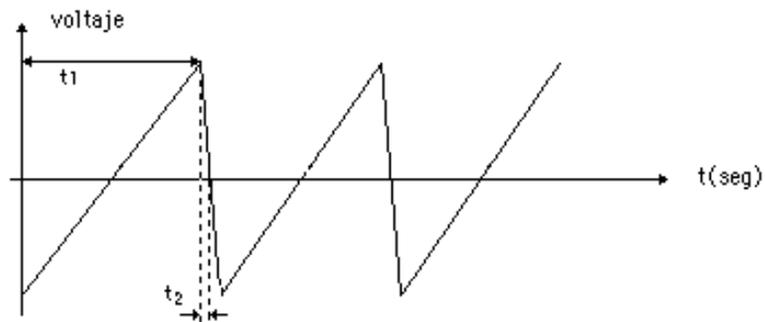


Acoplamiento AC, 0.5V/div

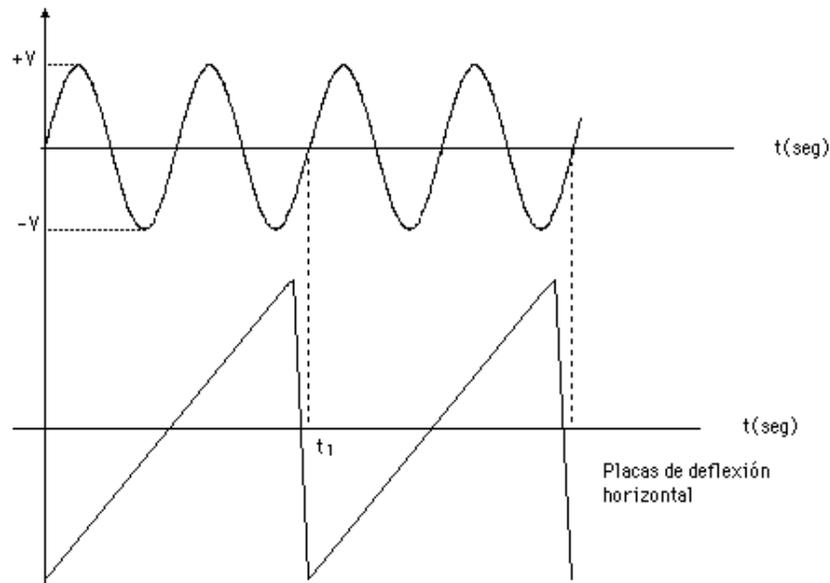


# Amplificador Horizontal - Base de tiempo

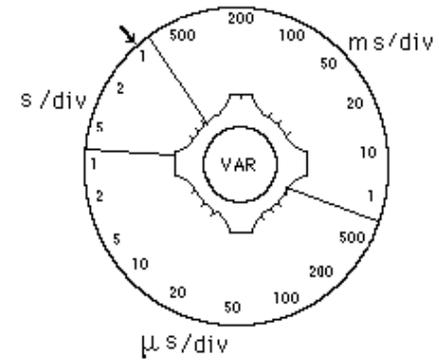
## Diente de sierra



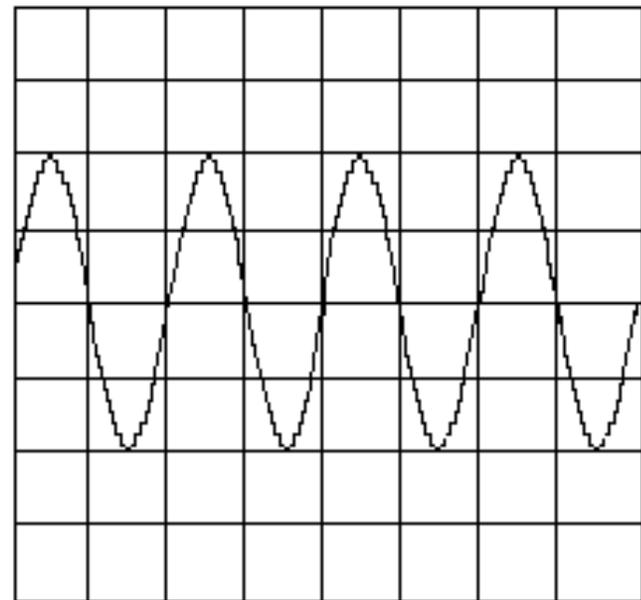
## Formas de onda



## Perilla de calibración

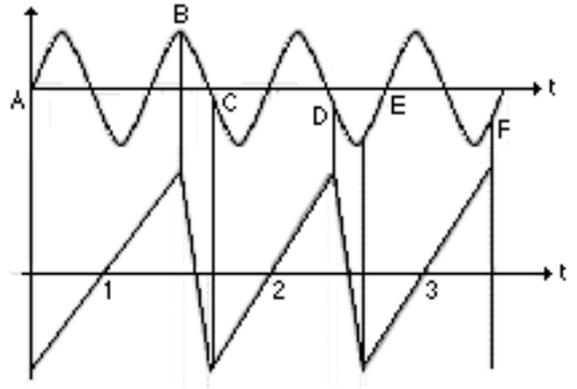


## Pantalla osciloscopio

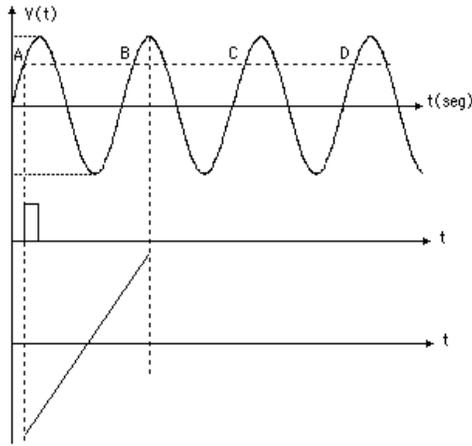


# Circuito de disparo

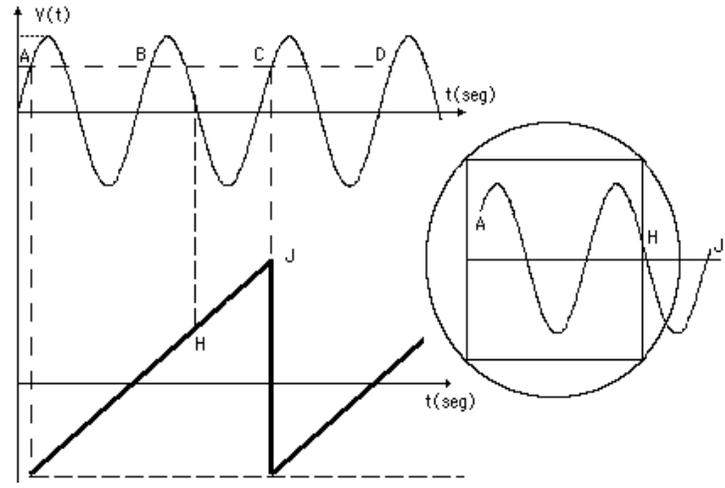
Diente de sierra y señal no sincronizadas



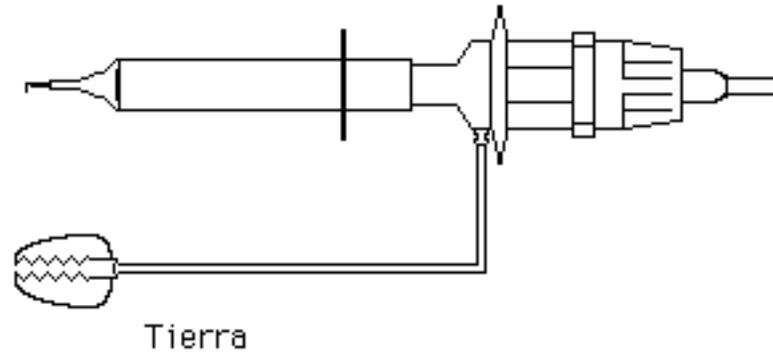
Operación del circuito de disparo



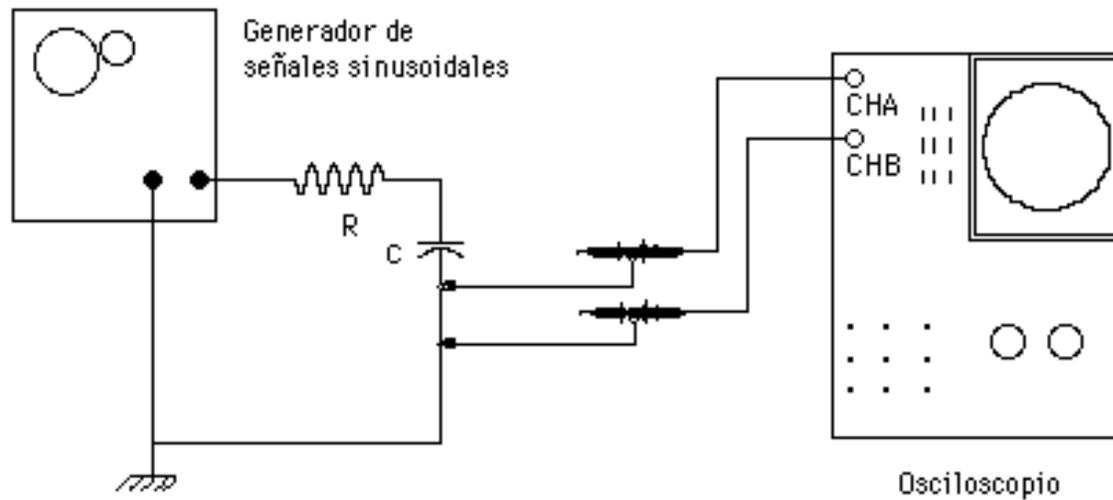
Sincronización de la diente de sierra para observar una señal estable



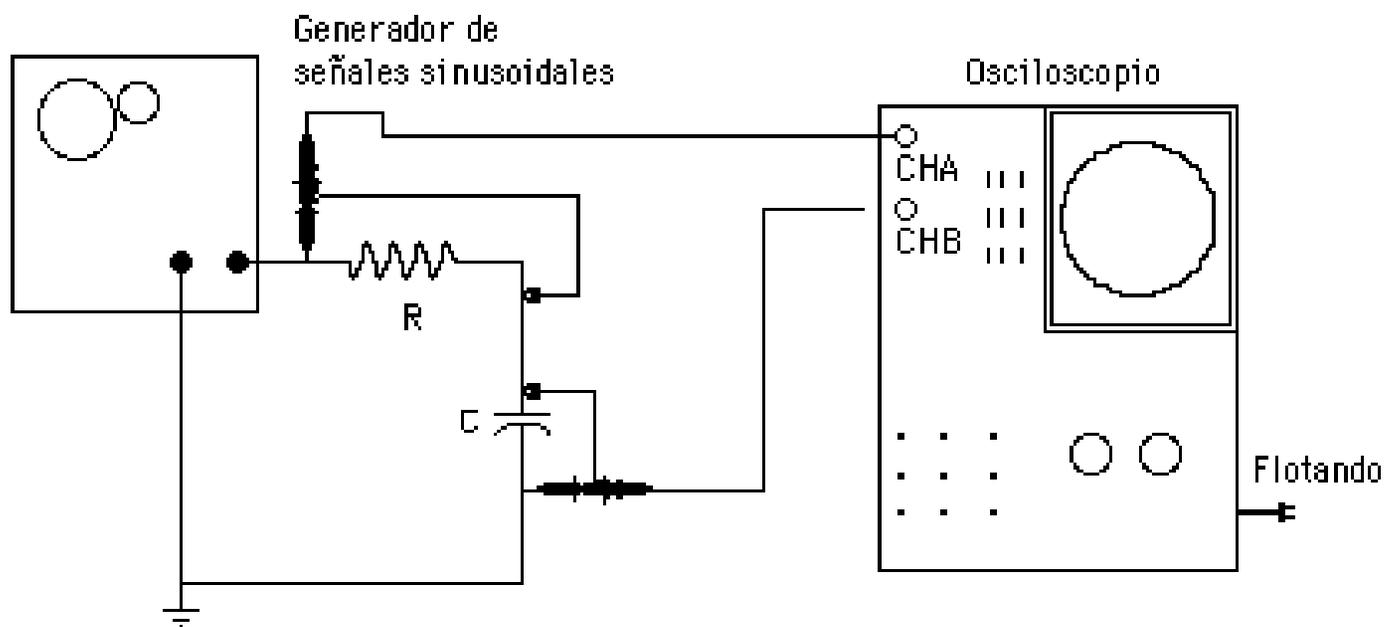
# PUNTAS DE PRUEBA DEL OSCILOSCOPIO



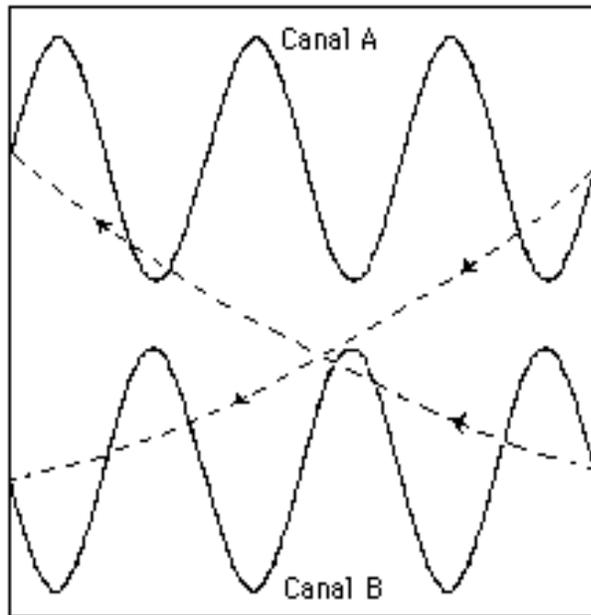
## Osciloscopio con conexión a tierra (aterrado)



## Osciloscopio sin conexión a tierra (flotando)

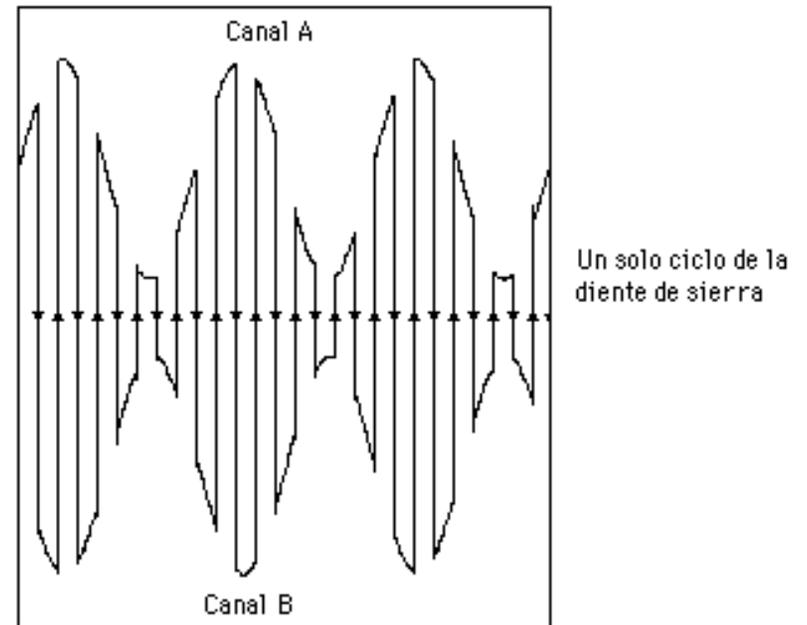


# SISTEMAS PARA PRESENTAR DOS O MAS SEÑALES SIMULTÁNEAS SOBRE LA PANTALLA DE UN OSCILOSCOPIO CON UN SOLO CAÑÓN



**MODO ALTERNADO (ALT)**

Para señales de frecuencias altas



**MODO CORTADO (CHOP)**

Para señales de frecuencias bajas

# EL OSCILOSCOPIO DIGITAL



## ¿QUÉ ES UN CIRCUITO DIGITAL?

Es un circuito electrónico en el cual las señales tienen exclusivamente **dos niveles** de voltaje, y la información se codifica utilizando este tipo de señales como los dos estados de un sistema binario, con lo cual se pueden realizar operaciones muy complejas, utilizando el Álgebra Booleana.

Los dos valores de las señales se identifican como:

ALTO y BAJO

VERDADERO y FALSO (ÁLGEBRA BOOLEANA)

UNO LÓGICO y CERO LÓGICO

1 y 0

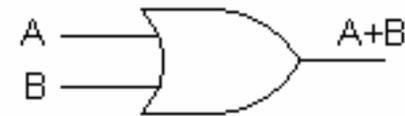
# ALGEBRA BOOLEANA

El Algebra Booleana define las operaciones entre elementos que pueden tener solo dos valores, y que producen resultados que solo pueden tener también uno de esos dos valores.

Los operadores básicos son AND (\*), OR (+) y NOT.

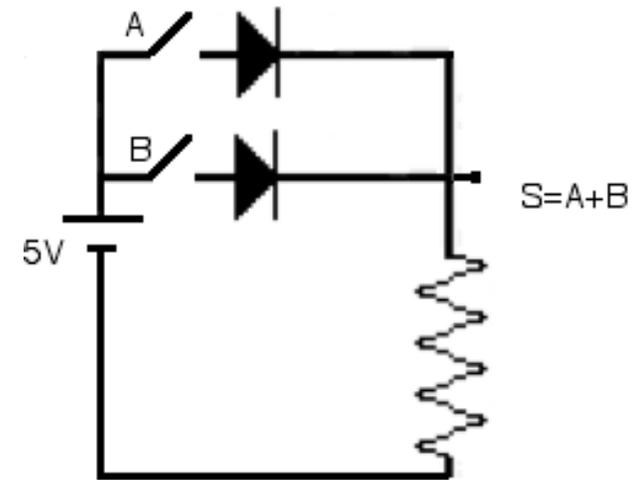
Ejemplo: El operador OR

A	B	A+B (OR)
0	0	0
0	1	1
1	0	1
1	1	1



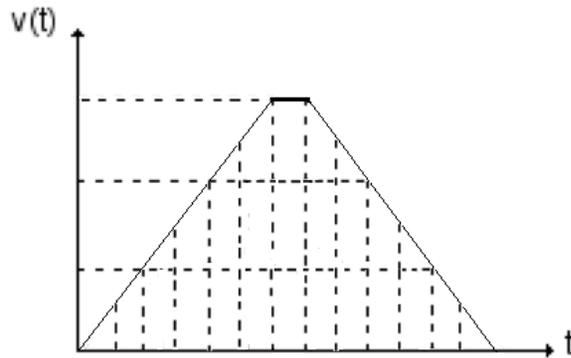
# ALGEBRA BOOLEANA Y CIRCUITOS ELECTRÓNICOS

SwA	SwB	S= A+B (OR)
Abierto	Abierto	0V
Abierto	Cerrado	5V
Cerrado	Abierto	5V
Cerrado	Cerrado	5V

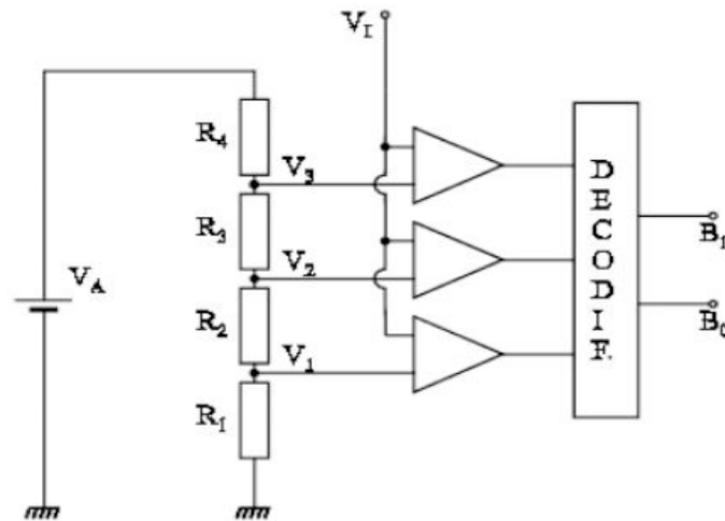
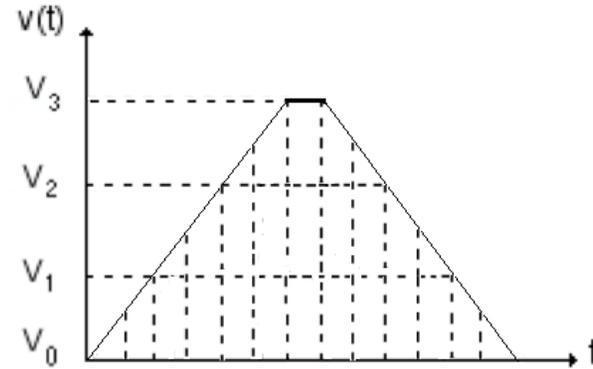


# CONVERSIÓN ANALÓGICA DIGITAL

Señal analógica

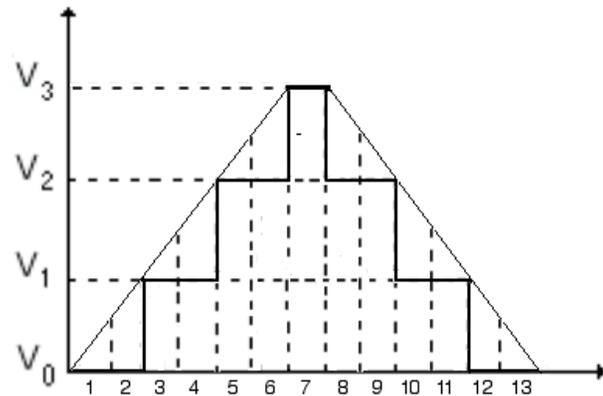


Definición de niveles



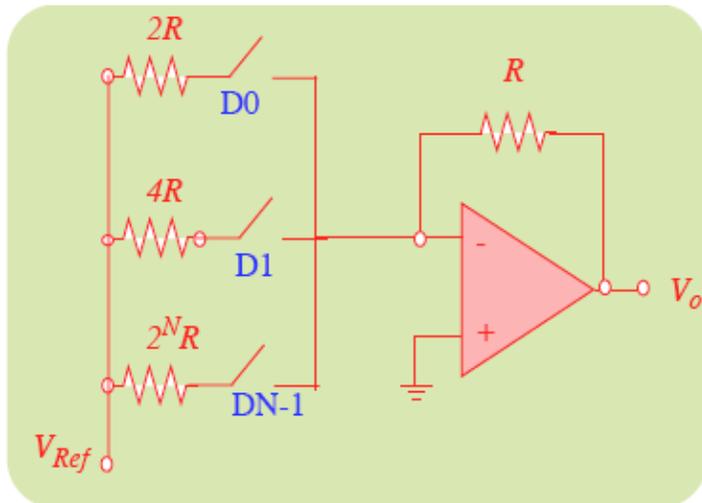
Conversor análogo-digital

# INFORMACIÓN DIGITALIZADA



**Señal discretizada**

INTERVALO	SALIDA OPERACIONALES			VALOR DIGITAL	
	A3	A2	A1	B2	B1
1 (0001)	V <sup>-</sup>	V <sup>-</sup>	V <sup>-</sup>	0	0
2 (0010)	V <sup>-</sup>	V <sup>-</sup>	V <sup>-</sup>	0	0
3 (0011)	V <sup>-</sup>	V <sup>-</sup>	V <sup>+</sup>	0	1
4 (0100)	V <sup>-</sup>	V <sup>-</sup>	V <sup>+</sup>	0	1
5 (0101)	V <sup>-</sup>	V <sup>+</sup>	V <sup>+</sup>	1	0
6 (0110)	V <sup>-</sup>	V <sup>+</sup>	V <sup>+</sup>	1	0
7 (0111)	V <sup>+</sup>	V <sup>+</sup>	V <sup>+</sup>	1	1
8 (1000)	V <sup>-</sup>	V <sup>+</sup>	V <sup>+</sup>	1	0
9 (1001)	V <sup>-</sup>	V <sup>+</sup>	V <sup>+</sup>	1	0
10 (1010)	V <sup>-</sup>	V <sup>-</sup>	V <sup>+</sup>	0	1
11 (1011)	V <sup>-</sup>	V <sup>-</sup>	V <sup>+</sup>	0	1
12 (1100)	V <sup>-</sup>	V <sup>-</sup>	V <sup>-</sup>	0	0
13 (1101)	V <sup>-</sup>	V <sup>-</sup>	V <sup>-</sup>	0	0



**Convertidor digital analógico**