



Introducción al sistema binario

A finales de la década de 1930, Claude Shannon mostró que utilizando interruptores que se encontraban cerrados para "verdadero" y abiertos para "falso", se podían llevar a cabo operaciones lógicas asignando el número 1 a "verdadero" y el número 0 a "falso".

Este sistema de codificación de información se denominó **binario**. Es la forma de codificación que permite el funcionamiento de los ordenadores. El sistema binario utiliza dos condiciones (representadas por los dígitos 0 y 1) para codificar información.

Desde el año 2000 AC, los seres humanos han contado utilizando 10 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Esto se denomina "base decimal" (o base 10). Sin embargo, las civilizaciones antiguas, e incluso algunas aplicaciones actuales, utilizaron y continúan utilizando otras bases numéricas:

- Sexagesimal (60), utilizada por los Sumerios. Esta base se utiliza actualmente en nuestro sistema de cronometraje, para los minutos y los segundos,
- Vigesimal (20), utilizada por los mayas,
- Duodecimal (12), utilizada en el sistema monetario del Reino Unido e Irlanda hasta 1971: una "libra" valía veinte "chelines", y un "chelín" valía doce "peniques". El actual sistema de cronometraje también se basa en doce horas (algo que se traduce en el uso que se le da en Norteamérica),
- Quinario (5), utilizado por los mayas,
- Binario (2), utilizado por la tecnología digital.

El bit (dígito binario)

El término **bit** (abreviado con la minúscula *b*) significa **dígito binario**, y corresponde al número 0 o 1 en la numeración binaria. Es la unidad de información más pequeña que puede manipular una máquina digital. Es posible representar esta información binaria:

- con una señal eléctrica o magnética que, más allá de un cierto nivel, representa el 1,
- a través de la aspereza o profundidad de los hoyos de una superficie,
- utilizando circuitos eléctricos, componentes eléctricos que poseen dos condiciones estables (una que representa al 1 y la otra al 0).

Por lo tanto, el bit se puede establecer con uno de dos estados: tanto con 1 como 0. Con dos bits, se pueden obtener 4 condiciones diferentes (2x2):

0	0
0	1
1	0
1	1



Con 3 bits, se pueden obtener ocho condiciones diferentes (2^3):

Valor binario de 3 bits	Valor decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Con un grupo n de bits, es posible representar 2^n valores.

Valores de los bits

En un número binario, el **valor** de un bit depende de su posición, empezando desde la derecha. Como las decenas, centenas y millares en un número decimal, el valor de un bit se incrementa por dos a medida que va desde la derecha hacia la izquierda, como se muestra en el siguiente cuadro:

Numero binario	1	1	1	1	1	1	1	1
Valor	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

Conversión de binario a decimal

Para convertir una cadena binaria en un número decimal se debe multiplicar cada bit por su valor, y luego sumar los productos. De esta forma, la cadena binaria 0101 en número decimal sería:

$$\begin{aligned} & 2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1 \\ & = 8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 \\ & = 5 \end{aligned}$$



Conversión de decimal a binario

Para hacer la conversión de decimal a binario, hay que ir dividiendo el número decimal entre dos y anotar en una columna a la derecha el resto (un 0 si el resultado de la división es par y un 1 si es impar).

La lista de ceros y unos leídos de abajo a arriba es el resultado.

Ejemplo: vamos a pasar a binario 79_{10}

79 **1** (impar). Dividimos entre dos:
 39 **1** (impar). Dividimos entre dos:
 19 **1** (impar). Dividimos entre dos:
 9 **1** (impar). Dividimos entre dos:
 4 **0** (par). Dividimos entre dos:
 2 **0** (par). Dividimos entre dos:
 1 **1** (impar).

Por tanto, $79_{(10)} = 1001111_{(2)}$

El byte

El **byte** (abreviado con la mayúscula *B*) es una unidad de información compuesta por 8 bits. Se puede utilizar para almacenar, entre otras cosas, un carácter, como por ejemplo una letra o un número.

Agrupar números en cúmulos de 8 facilita su lectura, así como agrupar números en grupos de tres hace más legibles los millares cuando se trabaja en base decimal. Por ejemplo, el número "1.256.245" se lee mejor que "1256245".

Por lo general, una unidad de información de 16 bits se denomina **palabra**.

Una unidad de información de 32 bits se denomina **palabra doble** (o también, *dword*).

Para un byte, el menor número posible es 0 (representado por ocho ceros: 00000000), y el mayor es 255 (representado por ocho unos: 11111111), que permite la creación de 256 valores diferentes.

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

Kilobytes y megabytes

Durante mucho tiempo, la informática fue una ciencia inusual ya que utilizaba diferentes valores para sus unidades, diferentes a las del sistema métrico (también llamado "sistema Internacional"). Los usuarios de ordenadores aprendían con rapidez que 1 kilobyte estaba compuesto por 1024 bytes. Por este motivo, en diciembre de 1998, la Comisión Electrotécnica Internacional intervino en el tema. La lista siguiente incluye las unidades estandarizadas de la IEC:

- Un kilobyte (kB) = 1000 bytes
- Un megabyte (MB) = 1000 kB = 1.000.000 bytes
- Un gigabyte (GB) = 1000 MB = 1.000.000.000 bytes
- Un terabyte (TB) = 1000 GB = 1.000.000.000.000 bytes



¡Advertencia! Algunos programas (e incluso algunos sistemas operativos) aún utilizan la notación anterior a 1998, que mostraremos a continuación:

- Un kilobyte (kB) = 2^{10} bytes = 1024 bytes
- Un megabyte (MB) = 2^{20} bytes = 1024 kB = 1.048.576 bytes
- Un gigabyte (GB) = 2^{30} bytes = 1024 MB = 1.073.741.824 bytes
- Un terabyte (TB) = 2^{40} bytes = 1024 GB = 1.099.511.627.776 bytes

El IEC también definió el kilo binario (kibi), al mega binario (mebi), al giga binario (gibi) y al tera binario (tebi).

Se definieron de la siguiente manera:

- Un kibibyte (kiB) vale 2^{10} = 1024 bytes
- Un mebibyte (MiB) vale 2^{20} = 1.048.576 bytes
- Un gibibyte (GiB) vale 2^{30} = 1.073.741.824 bytes
- Un tebibyte (TiB) vale 2^{40} = 1.099.511.627.776 bytes

En algunos idiomas, como el francés y el finlandés, la palabra byte no empieza con la letra "b", pero la mayor parte de la comunidad internacional prefiere el término en inglés "byte". Esto da las siguientes notaciones para kilobyte, megabyte, gigabyte, y terabyte:

kB, MB, GB, TB



Observe el uso de la mayúscula *B* para distinguir *Byte* de *bit*.



Operaciones binarias

En el sistema binario se pueden realizar operaciones simples tales como adición, sustracción y multiplicación.

Adición en el sistema binario

La adición en el sistema binario sigue las mismas reglas que en el sistema decimal: Se comienza agregando los bits que tienen menor valor (aquellos que se encuentran en la derecha) y se lleva el valor al siguiente lugar cuando la suma de dos bits en la misma posición es más grande que el valor mayor de la unidad (en sistema binario: 1). Luego, este valor se transporta al bit de la siguiente posición.

Por ejemplo:

	0	1	1	0	1
+	0	1	1	1	0
-	-	-	-	-	-
	1	1	0	1	1