

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,
- 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 (2×2):

0 0

0 1

1 0

1 1

Con 3 bits, se pueden obtener ocho condiciones diferentes (2x2x2):

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	
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

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:

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

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.

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

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.

(<http://physics.nist.gov/cuu/Units/binary.html>). 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

Conversión de binario a decimal y viceversa

Caso 1: Un número

Si la información es numérica, basta con convertir la cifra a base $_2$ (binario).

Recordemos que los números comunes están expresados en base $_{10}$, nomenclatura que significa que:

1234 en base $_{10}$ quiere decir que se compone de la suma de:

$$\begin{aligned} &4 \times 10^0 \\ &3 \times 10^1 \\ &2 \times 10^2 \\ &1 \times 10^3 \end{aligned}$$

1001 en base $_2$ quiere decir que se compone de la suma de:

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

En otras palabras, 1001 en base₂ es igual a 9 en base₁₀

Caso2 : Una letra o carácter cualquiera

Existen códigos para computadora que transforman las letras en números y viceversa

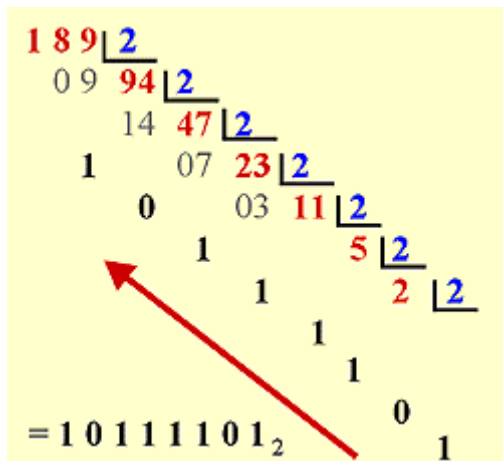
Uno de los mas conocidos es el código ASCII, que consiste en 255 caracteres correspondientes a todas las cifras, las letras y los caracteres comunes de computadora. Cualquier letra se transforma en un número utilizando este código, y los numero resultantes se digitalizan utilizando el procedimiento descrito en el paso anterior.

Veamos ahora cómo llevamos el número binario 10111101₂ a su equivalente en el sistema numérico decimal. Para descomponerlo en factores será necesario utilizar el 2, correspondiente a su base numérica y elevarlo a la potencia que le corresponde a cada dígito, de acuerdo con el lugar que ocupa dentro de la serie numérica. Como exponentes utilizaremos el "0", "1", "2", "3" y así sucesivamente, hasta llegar al "7", completando así la cantidad total de exponentes que tenemos que utilizar con ese número binario. La descomposición en factores la comenzamos a hacer de izquierda a derecha empezando por el mayor exponente, como podrás ver a continuación en el siguiente ejemplo:

$$\begin{aligned}10111101_2 &= (1 \cdot 2^7) + (0 \cdot 2^6) + (1 \cdot 2^5) + (1 \cdot 2^4) + (1 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0) \\&= (128) + (0) + (32) + (16) + (8) + (4) + (0) + (1) \\&= 189_{10}\end{aligned}$$

En el resultado obtenido podemos ver que el número binario 10111101₂ se corresponde con el número entero 189 en el sistema numérico decimal.

Seguidamente realizaremos la operación inversa, es decir, convertir un número perteneciente al sistema numérico decimal (base 10) a un número binario (base 2). Utilizamos primero el mismo número 189 como dividendo y el 2, correspondiente a la base numérica binaria del número que queremos hallar, como divisor. A continuación el resultado o cociente obtenido de esa división (94 en este caso), lo dividimos de nuevo por 2 y así, continuaremos haciendo sucesivamente con cada cociente que obtengamos, hasta que ya sea imposible continuar dividiendo. Veamos el ejemplo:



Una vez terminada la operación, escribimos los números correspondientes a los residuos de cada división en orden inverso, o sea, haciéndolo de abajo hacia arriba. De esa forma obtendremos el número binario, cuyo valor equivale a **189**, que en este caso será: **10111101₂**.