



Bits y bytes

INTRODUCCIÓN

Todos los componentes que constituyen el hardware del ordenador, funcionan con **circuitos electrónicos digitales**. La electrónica digital trabaja con dos niveles de tensión 0V ó 5V, equivalentes a los valores 0 y 1 del sistema binario.

En informática, la unidad mínima de almacenamiento es el bit (b). La palabra bit proviene de las palabras inglesas **binary digit** (Dígito binario).

Un **bit** (1 b) puede almacenar solamente un 1 o un 0. Podemos imaginar un bit como una bombilla, que puede tener dos estados:



apagada

0



encendida

1

Combinaciones de bits Al combinar bits, las posibles combinaciones de 1 y 0 se multiplican, si con 1b tenemos dos combinaciones posibles, "0" y "1", si unimos 2 bits tendremos 4, que son, "00", "01", "10" y "11". Es decir, con 2 bits podemos asignar 4 valores diferentes.

Ejemplo: Si asignamos colores a cada dos bits podríamos tener un dibujo con puntos (pixels) de cuatro colores distintos "00" = Azul; "01" = Verde; "10" = Rojo y "11" = Amarillo

A través de secuencias de bits, se puede codificar cualquier valor discreto como números, palabras, e imágenes. **Con 3 bits se pueden conseguir $2^3 = 8$ combinaciones posibles** y con 4 bits (llamado nibble) se pueden representar hasta $2^4 = 16$ valores diferentes.

R G B	Color
0 0 0	Negro
0 0 1	Azul
0 1 0	Verde
0 1 1	Cian
1 0 0	Rojo
1 0 1	Magenta
1 1 0	Amarillo
1 1 1	Blanco

Byte

Si unimos 8 bits (8 b) obtenemos un byte (B), que también es una unidad muy pequeña de almacenamiento. Con un byte (B) podemos contar desde 00000000 (0 en decimal) hasta 11111111 (255 en decimal) o podemos tener 256 combinaciones diferentes, por ejemplo, para asignar a colores o caracteres.



Múltiplos del byte Por lo tanto, para almacenar gran cantidad de información en informática utilizamos los múltiplos del byte¹:

Bit (b) = 1 bit.

Byte (B) = 8 bits.

Kibibyte (KiB) = 1.024 bytes.

Mebibyte (MiB) = 1.024 KB = 1.048.576 bytes.

Gibibyte (GiB) = 1.024 MB = 1.073.741.824 bytes.

Tebibyte (TiB) = 1.024 GB = 1.099.511.627.776 bytes...

Ordenados en una tabla:

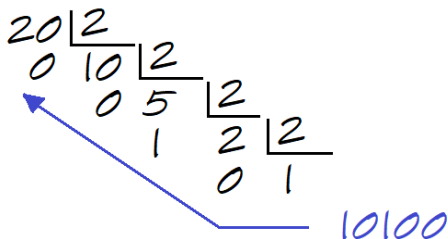
Símbolo	Nombre	Valor
b	Bit	1 bit
B	Byte	8 bits
KiB	Kibibyte	1024 bytes
MiB	Mebibyte	1024 kibibytes
GiB	Gibibyte	1024 mebibytes
TiB	Tebibyte	1024 gibibytes
PiB	Pebibyte	1024 tebibytes
EiB	Exabibyte	1024 pebibytes

Al trabajar sólo con dos niveles de tensión podemos representar todos los datos mediante el sistema numérico binario, nombre que recibe por el número de elementos a utilizar para mostrar todos los números. Nosotros estamos acostumbrados a utilizar el sistema decimal y cualquier otro sistema nos resulta difícil de comprender, y por ello debemos aprender a “traducir” cualquier número dado en el sistema binario a sistema decimal y viceversa

SISTEMA BINARIO

De decimal a binario: Bastará con dividir el número dado en decimal por la base del sistema binario (la base es el número de elementos que utiliza) de forma sucesiva (sin sacar decimales) hasta que el cociente sea menor que la base.

Ejemplo: Pasar el número 20_{10} al sistema binario.



Realizando la lectura como indica la flecha obtenemos: $20_{10} = 10100_2$

¹ Para reforzar lo visto en clase y ver la diferencia entre kilobytes y kibibytes te aconsejamos leer el artículo de tecnología.org [Kibibytes, mebibytes e xibibytes](http://tecnología.org/Kibibytes,mebibytesexibibytes) | tecnología.org



De binario a decimal: Cada posición tiene un determinado “peso”, es decir, comenzamos siempre por la derecha (denominada bit menos significativo por ser la parte de menor peso) y cogiendo la base de sistema utilizado se eleva a cero la primera cifra, la segunda a uno... hasta la última de la izquierda (bit más significativo). Los valores obtenidos son los pesos de las diferentes posiciones y bastará con multiplicar cada cifra con su respectivo peso y sumar los resultados.

Ejemplo: Pasar el número 10100_2 al sistema decimal.

$$10100|_2 = 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 20|_{10}$$

El resultado es: $10100|_2 = 20|_{10}$

Ejercicios a realizar:

1.- Pasa a codificación binaria los siguientes números dados en decimal.

- 30
- 128
- 1024
- 64
- 135

2.- Pasa a codificación decimal los siguientes números dados en binario.

- 10110010
- 0011
- 10
- 10010001
- 11111100