



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Representación de la información digital

Apellidos, Nombre	Rebollo Pedruelo, Miguel (mrebollo@dsic.upv.es)
Departamento	Sistemas Informáticos y Computación
Centro	Facultad de Administración y Dirección de Empresas



1. Resumen

Los ordenadores son máquinas digitales y como tales operan con información representada en formato binario. La unidad elemental de información es el bit. Una misma secuencia de bits puede ser interpretada de muchas formas distintas por el ordenador: como una cantidad numérica, como caracteres de un alfabeto, como datos (por ejemplo, una foto en una cámara digital o una canción en un reproductor de MP3) y también como las instrucciones de un programa.

Habitualmente, el ordenador manipula los bits en bloques de tamaño fijo. Un bloque de 8 bits se denomina byte. A partir de ahí, los múltiplos superiores son el kilobyte, megabyte, gigabyte, etc.

En este artículo encontrarás los fundamentos sobre sistemas de representación de información digital que permiten a los ordenadores y a cualquier dispositivo electrónico representar y procesar la información.

2. Objetivos

Cuando se hayan asimilado los contenidos de este documento, el alumno debe poder

- Explicar en términos generales cómo se representa la información digital
- Convertir números decimales en binarios y viceversa
- Interpretar las tablas de código ASCII
- Usar UNICODE (UTF-8) para la codificación de caracteres en el navegador
- Calcular la correspondencia entre las magnitudes empleadas como múltiplos de bits.

3. Introducción

Los ordenadores son máquinas que son capaces de procesar información, de convertir información de entrada en información de salida. En realidad es una máquina muy simple: todas las operaciones que realizan son operaciones matemáticas, operaciones aritméticas (como una suma) y lógicas (operaciones que tienen como resultado únicamente dos posibles valores: verdadero o falso). Y una de las características que hace diferentes a los ordenadores de otro tipo de máquinas es que se trata de dispositivos digitales, basados en circuitos electrónicos, en contraposición a los dispositivos analógicos (por ejemplo, las primeras máquinas de calcular mecánicas)..

Un dispositivo digital está basado en circuitos electrónicos que son capaces de procesar información representada únicamente por dos estados. A estos estados se les puede denominar como "verdadero" o "falso", o como 1 ó 0. Estos valores realmente representan dos voltajes diferentes en el circuito.¹

Los sistemas de numeración binaria no nacieron con la informática, ni siquiera con la electrónica. Los primeros documentos que lo muestran datan del siglo III a.C. y se deben a un matemático indio, Pingala, que empleó este sistema para representar sílabas

¹ por ejemplo, los voltajes típicos empleados en un ordenador suelen ser 5v. y 12v. para representar los ceros y los unos respectivamente.



largas y cortas en la escritura sánscrita. En 1854, el matemático George Boole desarrolló un sistema de lógica basado en dos estados, conocido como Álgebra de Boole, que sentó las bases para la construcción de los circuitos electrónicos.

4. Bits: dígitos binarios

En el mundo de la informática (y de la electrónica en general), la información está formada por elementos discretos, contables, que se pueden separar y aislar. El elemento más pequeño de información que se puede representar en un circuito es el **bit** (*Binary digIT* -dígito binario-). En cambio, el mundo analógico es un mundo continuo: la luz, el sonido y otras magnitudes que podemos percibir a través de nuestros sentidos están formados por ondas.

Un bit puede almacenar únicamente dos valores, que podemos representar como 0 ó 1, encendido o apagado, blanco o negro, alto o bajo... Así, podemos considerar que un ordenador no es más que una máquina formada por millones de interruptores, cada uno de los cuales es capaz de almacenar un bit. El ordenador, entonces, toma una entrada representada como una secuencia de bits, los manipula, realizando operaciones aritméticas y lógicas sobre ellos, y proporciona al usuario una secuencia de bits como resultado de salida.

Pero como los humanos somos bastante malos reconociendo secuencias de ceros y unos como información inteligible, utilizamos dispositivos de entrada y de salida que se encargan de traducir las secuencias de bits en algo comprensible para nosotros: una canción, una imagen, una película, un texto impreso o, ¿por qué no?, un olor o una sensación (por ejemplo, la vibración producida por el mando de una consola).

Ahora bien ¿qué es lo que entiende un ordenador ante una secuencia como 00101111? La respuesta depende de qué se esté representado con esa combinación de bits.

4.1. Bits como números

La mayoría de los sistemas empleados para contar se basan en el sistema de **numeración decimal**.² Eso quiere decir que empleamos 10 dígitos distintos (del 0 al 9) y cualquier número se construye por la combinación de n dígitos. El número de cifras es lo que se llama la base de numeración. El sistema decimal es un sistema en **base 10**.

- Si disponemos de 1 dígito decimal, podemos contar de 0 a 9, es decir, 10 combinaciones distintas.
- Si disponemos de 2 dígitos decimales, podemos contar de 0 a 99. Son 100 (10^2) combinaciones distintas
- Si disponemos de 3 dígitos, podemos contar de 0 a 999, o sea, 1.000 (10^3) combinaciones distintas.
- En general, si disponemos de n dígitos, podremos diferenciar 10^n combinaciones distintas y podremos contar de 0 a 10^n-1 .

² Aunque el sistema decimal es el más habitual en las lenguas indo-europeas, existen culturas que emplean otros sistemas de numeración. Uno de ellos es el sistema vigesimal (base 20), que encontramos en el euskera y que se ha extendido a otras lenguas de forma esporádica a través de los celtas, como en el caso del francés, el danés o el galés.



La descomposición de un número en el sistema decimal según su posición se hace de la siguiente forma:

$$1437 = 1 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$$

Ecuación 1. Descomposición de un número en el sistema decimal

Es algo tan natural que apenas os damos cuenta. Pero veamos qué ocurre cuando se cambia el sistema de numeración.

El **sistema binario** es un sistema en **base 2**. Esto quiere decir que sólo tenemos 2 cifras para representar la información: el 0 y el 1. Cualquier número se forma usando únicamente estas dos cifras: 0, 1, 10, 11, 100, 101 y así sucesivamente. Igual que en el sistema decimal:

- si tenemos un dígito podemos contar de 0 a 1. Eso nos da un total de 2 posibilidades
- con dos dígitos, las posibilidades son 00, 01, 10 y 11, es decir, 4 posibilidades distintas (2^2)
- con 3 dígitos las combinaciones posibles son 000, 001, 010, 011, 100, 101, 110 y 111, que da un total de 8 combinaciones (2^3)

Y así sucesivamente. De la misma forma que en el sistema decimal, si disponemos de n dígitos binarios podremos representar 2^n combinaciones diferentes. Y la descomposición de un número binario sigue las mismas reglas que en el caso de la numeración decimal:

$$1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

Ecuación 2. Descomposición de un número en el sistema decimal

En general, dada una base de numeración b , con n dígitos pueden formarse un total de b^n combinaciones diferentes.



Actividad: Comprueba esta afirmación con al menos dos bases de numeración y 3 números distintos .

Como puede observarse, el número de combinaciones posibles en binario son potencias de dos. Si seguimos la secuencia:

$2^1 = 2$	$2^6 = 64$
$2^2 = 4$	$2^7 = 128$
$2^3 = 8$	$2^8 = 256$
$2^4 = 16$	$2^9 = 512$
$2^5 = 32$	$2^{10} = 1024$

¿Te resultan familiares estos valores? En efecto, la capacidad de almacenamiento de discos duros, memorias USB, la RAM del ordenador o incluso la velocidad de transferencia a través de Internet se mide con estos valores.



Puede realizarse transformaciones de un sistema de numeración en otro de forma sencilla. A esta operación se la denomina **cambio de base**.

La transformación de un número binario en su correspondiente decimal es inmediata a través de la descomposición del número binario según el **sistema posicional** que corresponde a la base 2, tal y como muestra la Ecuación 2. Simplemente hay que realizar la operación planteada:³

$$1011_{(2)} = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 8 + 0 + 2 + 1 = 11_{(10)}$$

La transformación de decimal a binario es algo más compleja. Se trata de un proceso iterativo, por el que debemos realizar divisiones sucesivas hasta que no se pueda seguir. El resultado de la conversión nos lo da el resto de las correspondientes divisiones más el cociente de la última división en orden inverso. Si observas la Figura 1, las divisiones sucesivas entre 2 nos dan como resto 1, 1 y 0 y el último cociente es $2/2 = 1$. Así pues, $11_{(10)} = 1011_{(2)}$.

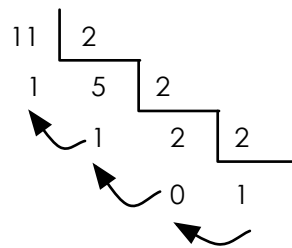


Figura 1. Conversión de decimal a binario

Otros sistemas de numeración empleados frecuentemente en Informática son el **octal** (base 8, dígitos del 0 al 7) y el **hexadecimal** (base 16, dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Estos sistemas de numeración tienen una traducción muy rápida al sistema binario, ya que un número en octal puede representarse por 3 bits y un número en hexadecimal por 4 bits (ver Tabla 1). Las transformaciones son muy simples, ya que de cualquier a de estas bases a binario la traducción es directa dígito a dígito (no es necesario el método de las divisiones de la Figura 1); y de binario a octal o hexadecimal basta con agrupar los dígitos binarios de izquierda a derecha de 3 en 3 o de 4 en 4 respectivamente y aplicar la conversión (también se puede hacer por descomposición, pero es más rápido traducir directamente de la tabla).

Como muestra del uso de estos otros sistemas de numeración, simplemente indicar que una dirección IPv6 (las direcciones IP que identifican a un equipo dentro de Internet) se representa mediante 32 dígitos hexadecimales; en total, 128 bits ($32 \times 4 = 128$), por ejemplo

2001:0db8:85a3:08d3:1319:8a2e:0370:7334

Para acabar, simplemente indicar que las operaciones matemáticas básicas están también definidas en binario y que se pueden hacer sumas, restas, multiplicaciones y divisiones tal y como las resolvemos en el sistema decimal.

³ Recuerda que cualquier número elevado a 0 es 1.

El subíndice que aparece entre paréntesis indica la base de numeración. Fíjate que en este ejemplo puede resultar confuso omitir la base.



decimal	binario	octal	hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	a
11	1011	13	b
12	1100	14	c
13	1101	15	d
14	1110	16	e
15	1111	17	f

Tabla 1. Correspondencia entre bases de numeración



Actividad: Expresa en binario los números decimales siguientes:

194, 25, 67, 99, 32, 17.

Expresa en decimal, octal y hexadecimal los siguientes números binarios:

01001011, 10100110

4.1. Bits como caracteres

Una secuencia de bits puede representar también los **caracteres** de algún alfabeto. Sin embargo, en este caso el código binario asignado a cada letra no es más que una convención, un acuerdo, y podría ser cualquier otra asignación. De hecho, como vamos a ver, existen varias maneras de representar los caracteres y todas ellas se consideran distintos estándares.



Una de las empleadas en el alfabeto occidental es el **código ASCII**. Según este código (ver Tabla 2) cada carácter se representa mediante 1 byte (8 bits), con lo que tenemos $2^8 = 256$ caracteres distintos. Eso podría parecer suficiente ya que nuestro alfabeto tan sólo tiene 28 letras. Sin embargo, debes tener en cuenta que cada símbolo que aparece en el teclado debe aparecer en esta tabla. Así, las letras mayúsculas y minúsculas tienen códigos distintos, los números tienen otro código que no tiene nada que ver con su valor numérico (piensa en el número como grafía) y todos los caracteres que puedes ver en tu teclado tienen su correspondiente código. Y también algunos que no son imprimibles, como el espacio en blanco, el [Enter], la tecla de retroceso y algunas otras (los códigos 0 a 31 de la Tabla 2).

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
0 0 NUL	16 10 DLE	32 20 (espacio)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (56 38 8
9 9 TAB	25 19 EM	41 29)	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D -	61 3D =
14 E SO	30 1E RS	46 2E .	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?
ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F □

Tabla 2. Código ASCII



El código ASCII puede parecer suficiente si utilizamos un único alfabeto. Pero ¿qué ocurre si necesitamos incluir símbolos de alfabetos distintos en un mismo documento?. Por ejemplo, ¿has entrado alguna vez en una página web de un país con un alfabeto distinto? Una web en japonés, ruso, árabe, hebreo... Probablemente el navegador muestre una serie de símbolos extraños sin sentido. Para evitar este problema, y poder usar una única conversión plurilingüe existe el formato UNICODE o los UTF-8 y UTF-16.

El formato **UNICODE** es una representación variable que emplea 1 ó 2 bytes para representar cada carácter. En total, hay 65.000 combinaciones posibles, lo que es suficiente para representar todos los caracteres existentes en cualquier idioma del mundo. De esta manera podemos emplear una única codificación para acceder a cualquier texto en formato electrónico.



Actividad: comprueba la codificación que empleas en tu navegador habitual (MS Explorer, Firefox, Chrome, Safari...) ¿usas una representación UNICODE? Puedes ajustarlo desde la opción Ver > Codificación del texto (puede cambiar según el navegador). Comprueba su funcionamiento entrando en páginas plurilingües.

4.1. Bits como datos

Además de números o caracteres, cualquier información que procese el ordenador debe representarse en binario. Un fichero del procesador de textos, una presentación, una foto de las vacaciones, un archivo de audio de una canción en mp3, una película, un juego... todos ellos son ejemplos de información binaria.

En este caso, tenemos una gran variedad de formas de codificar y representar la información. Por ejemplo, una combinación de bits puede ser el color de un pixel en la pantalla del ordenador, un punto sobre el papel, una nota de una melodía o las coordenadas del puntero del ratón en la pantalla. Pero incluso para un mismo tipo de archivo una lista de ceros y unos puede interpretarse de formas distintas. Una imagen puede estar almacenada como un bmp, jpg, gif, png, raw y muchos otros. Y cada uno de ellos, una misma combinación de bits se interpreta de distintas maneras.

Esta variedad en la representación de la información es la que provoca que, si tratamos de abrir un fichero con un programa distinto al que lo ha creado, éste nos indique que no puede abrir el fichero. O que si se ha producido un error en la transmisión de datos (por ejemplo, descargando un fichero), éste sea ilegible para el ordenador, ya que la secuencia binaria no tiene ningún sentido. Es lo que ocurre, por ejemplo, cuando un DVD está estropeado o si no llega señal correcta a la TDT: vemos cuadrados extraños en la imagen o escuchamos sonidos ininteligibles porque no es posible representar la información correspondiente.

4.2. Bits como instrucciones

Pero los bits no se emplean para representar sólo los datos. El ordenador es una máquina digital todo lo que maneja está codificado en binario y que modifica esta información ejecutando las instrucciones de un programa. Y estas instrucciones también deben estar codificadas en binario.

En este caso, cómo se traducen las secuencias de bits en instrucciones es algo que se realiza dentro del procesador y que está codificado directamente en sus circuitos.



5. Múltiplos de bits

Igual que para cualquier medida, sobre las magnitudes básicas se definen **prefijos** que indican múltiplos de la misma. Por ejemplo, en el Sistema Internacional, (S.I.) tenemos los prefijo *deca*, *hecto*, *kilo*, etc. que anteponeamos a algunas magnitudes como las de longitud (metro) o masa (gramo).

En el caso de la numeración binaria, las magnitudes, su símbolo y su significado son los que se muestran en la Tabla 3. A partir de los exabytes, se emplea el alfabeto griego en orden inverso para denotar las siguientes potencias.

Sistema Internacional			Prefijos CEI		
Nombre	Símbolo	Valor	Nombre	Símbolo	Valor
bit	bit	0 ó 1	bit	bite	0 ó 1
byte	byte	8 bits	byte	byte	8 bits
kilobyte	KB	1.000 bytes	kibibyte	KiB	1.024 bytes
megabyte	MB	1.000 KB	mebibyte	MiB	1.024 KiB
gigabyte	GB	1.000 MB	gibibyte	GiB	1.024 MiB
terabyte	TB	1.000 GB	tebibyte	TiB	1.024 GiB
petabyte	PB	1.000 TB	pebibyte	PiB	1.024 TiB
exabyte	EB	1.000 PB	exbibyte	EiB	1.024 PiB
zettabyte	ZB	1.000 EB	zebibyte	ZiB	1.024 EiB
yottabyte	YB	1.000 ZB	yobibyte	YiB	1.024 ZiB

Tabla 3. Prefijos binarios y su significado

El problema de usar los prefijos del sistema internacional para las magnitudes binarias es el **desajuste** que se produce por usar bases de numeración distintas: el sistema internacional emplea potencias de 10 mientras que el sistema binario usa potencias de 2. A eso se debe que, cuando compras un disco duro de 1TB, realmente es 931GB. No hemos perdido 70GB por el camino. Lo que ocurre es que en el caso de los bits, el cambio de órdenes de magnitud se hace multiplicando por 1.024 (2^{10}), mientras que en el sistema decimal esos mismos prefijos indican que se debe multiplicar por 1.000 (10^3). De ahí las diferencias entre la capacidad real y la capacidad "nominal" del disco duro.

Para tratar de evitarlo, la Comisión Electrotécnica Internacional (CEI) ha propuesto unos prefijos distintos para los múltiplos decimales y binarios. Simplemente varía el prefijo para indicar si que es binario. Por ejemplo, en lugar de kilo es kibi (kilo-binario), en lugar de mega, mebi y así sucesivamente. La lista de prefijos completos y su valor real



está también en la Tabla 3. Así, para ser correctos podemos decir que tenemos un disco de 1 terabyte (1 TB) o de 931 gibibytes (931 GiB); ambos son equivalentes.

Sin embargo, esta propuesta no se ha popularizado y en la práctica se sigue empleando los prefijos del sistema internacional incluso para hacer referencia a las magnitudes binarias, asumiendo el redondeo que esto ocasiona.

6. Cierre

Los ordenadores son máquinas que manipulan información digital, cuyos valores vienen dados únicamente por dos posibles estados representados como 0 y 1. A este sistema de numeración se le conoce como sistema binario y a cada uno de estos dígitos se les llama como *bit*.

Como números, una secuencia de bits es un número expresado en base 2. De ahí que la mayoría de las magnitudes relacionadas con la informática se expresen como potencias de 2. Los números pueden convertirse fácilmente de una base de numeración a otra siguiendo unas reglas sencillas. Otras dos bases de numeración que tienen una especial importancia en el campo de la informática son la octal (base 8) y la hexadecimal (base 16). Su interés radica en la facilidad de la conversión de ambas bases con el sistema binario.

Como caracteres, los bits se agrupan de 8 en 8 (un byte). En este caso tenemos $2^8=256$ combinaciones distintas: suficientes para representar todos los símbolos del alfabeto. Esta codificación sigue un estándar conocido como código ASCII. Sin embargo, no es suficiente si necesitamos representar simultáneamente caracteres de varios alfabetos. En ese caso, existen otras codificaciones, como UNICODE, que permite representar hasta 65.000 caracteres impresos distintos.

Además, de números o caracteres, las secuencias de bits pueden representar cualquier otro tipo de datos, como ficheros de audio, de vídeo, imágenes o datos privados de determinadas aplicaciones. Y no sólo datos. Las series de bits representan también las instrucciones que es capaz de ejecutar el procesador siguiendo el esquema y la codificación que tiene guardados en sus circuitos.

Los bits se agrupan en unidades mayores. Enumeradas de menor a mayor son bit, byte, kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB), petabyte (PB) y exabyte (EB). A partir de esas magnitudes, se emplean las letras del alfabeto griego en orden inverso como prefijos: zettabyte (ZB), yottabyte (YB), etc.

7. Bibliografía

BEEKMAN, George: *Introducción a la informática*.- Ed. Pearson, Madrid, 2005

Publicado en España bajo licencia Creative Commons Reconocimiento 3.0 (CC BY). Información sobre los términos de la licencia disponible en la página

<http://creativecommons.org/licenses/by/3.0/es/deed.es>