



REPRESENTACIÓN BINARIA DE LA INFORMACIÓN: EL BIT, EL BYTE.

Para que un sistema informático sea útil es necesario que procese información. La información que se procesa en un sistema informático puede ser de diferentes tipos: textos, gráficos, música, etc.

DATOS: TIPOS DE DATOS E INFORMACIÓN

Debe distinguirse entre dato e información. **Dato** es cualquier número, imagen, sonido, símbolo, nombre, etc que no permite tomar ninguna decisión, porque no tienen un significado concreto, los datos deben procesarse para dotarlos de significado y convertirlos en información. Eso es lo que ocurre cuando trabajamos con un sistema informático, los resultados que nos muestra el ordenador son para nosotros información que nos ayuda en alguna tarea.

El ordenador, para poder procesar los distintos datos, debe convertirlos a un lenguaje numérico binario (0 y 1). Debido a la forma en que están construidos y al uso de los componentes electrónicos sólo dos valores pueden representarse. Para convertir los textos en números se utiliza un código de representación llamado **ASCII** (American Standard Code for Information Interchange) que es un estándar mundial. Una vez pasados a números se deben convertir esos números en valores binarios. Otro tipo de datos como sonidos o imágenes también deben convertirse en valores numéricos.

Para que un ordenador funcione necesita información con la que trabajar. Esta información es de varios tipos dependiendo de su función (pueden ser textos, gráficos, imágenes...) y será manejada por programas o aplicaciones informáticas (como Word, Excel, etc.)

LOS SISTEMAS DE CODIFICACIÓN

Los sistemas de codificación se utilizan para procesar la información que el usuario entiende y el ordenador no. Es evidente que el usuario y el sistema informático trabajan en lenguajes diferentes.

SISTEMAS DE NUMERACIÓN:

Se define un sistema de numeración como el conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.

Estos sistemas se caracterizan por la base a la que hacen referencia y que determina el distinto número de símbolos que lo componen. Nosotros utilizamos base 10 que tiene los siguientes símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.



Los sistemas de numeración que utilizamos son sistemas posicionales, es decir, el valor es relativo según la posición que ocupe.

$$283 = 2 \cdot 10^2 + 8 \cdot 10^1 + 3 \cdot 10^0 = 200 + 80 + 3$$

Todos los sistemas posicionales están basados en el Teorema Fundamental de la Numeración (TFN), que sirve para relacionar una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal

Teorema Fundamental de la Numeración

$$N_i = \sum_{i=-d,n} (\text{dígito})_i * (\text{base})^i$$

$$136_{10} = 1 \times 10^2 + 3 \times 10^1 + 6 \times 10^0$$

En él X es el valor absoluto del dígito en cuestión, i es la posición que ocupa el dígito con respecto al punto decimal y B es la base. El símbolo \sum (sumatoria) indica que para obtener el valor del número será necesario sumar todos los productos.

CODIFICACIÓN NUMÉRICA:

Son tres los sistemas de codificación que utiliza habitualmente un sistema informático:

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



- **BINARIO:** este sistema emplea dos símbolos diferentes: el cero y el uno (0, 1). Es el sistema que maneja el ordenador internamente, ya que lo utilizan sus componentes electrónicos. Cada uno de estos símbolos recibe el nombre de bit, entendiéndose por tal la mínima unidad de información posible. Los símbolos del sistema decimal pueden representarse (codificarse) en binario mediante el TFN. Cada símbolo decimal puede representarse con una combinación de cuatro bits.
- **OCTAL:** es un sistema en base 8 que utiliza los símbolos del 0 al 7 para representar las cantidades, las cuales quedan reproducidas posicionalmente por potencias de 8. El sistema de numeración en base 8 tiene una correspondencia directa con el binario, ya que cada símbolo en base 8 puede representarse mediante una combinación de 3 bits.
- **HEXADECIMAL:** es un sistema de numeración en base 16. utiliza 16 símbolos diferentes, del 0 al 9 y los dígitos valores (o representan, respectivamente los dígitos 10, 11, 12, 13, 14 y 15 del sistema decimal. Este sistema también tiene una correspondencia directa con el sistema binario, ya que cada símbolo en base 16 se puede representar con combinación de 4 bits.

El sistema que maneja internamente el ordenador es el binario, pero en ocasiones por comodidad en el manejo de los datos, se suele utilizar el octal y el hexadecimal, ya que mucha de la información que nos muestra el sistema operativo, como direcciones de memoria, está expresada en hexadecimal.

CAMBIOS DE BASE DE NUMERACIÓN

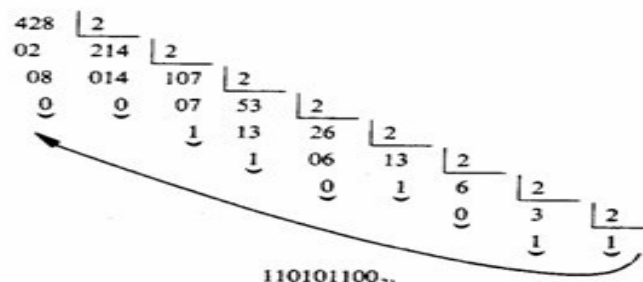
El sistema informático trabaja en el sistema de numeración binario. Nosotros trabajamos en el sistema de numeración decimal. El ordenador no entiende el sistema decimal, pero nosotros tampoco entendemos el binario.

Es necesario saber interpretar el código binario para entender las operaciones que se realizan dentro del ordenador. Operaciones que hay que saber realizar:

CONVERSIÓN DE DECIMAL A BINARIO.

Para pasar un número entero de base decimal a base binaria se divide el número decimal entre 2, el cociente se vuelve a dividir entre 2, y así sucesivamente; los restos obtenidos forman el número en el sistema binario.

EJEMPLO: Convertir el número $428)_{10}$ en su correspondiente binario.



Por tanto, $428)_{10} = 110101100)_2$

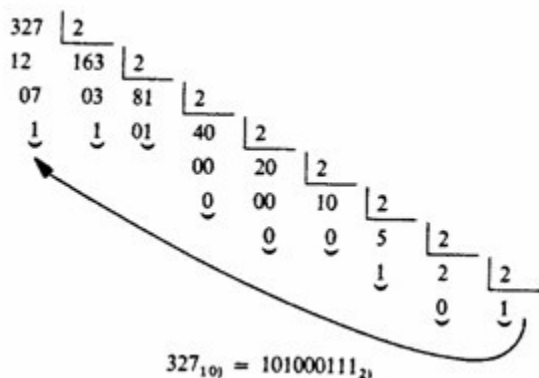


Para saber más:

Si el número decimal tiene parte fraccionaria, la parte entera se convierte de la misma manera que se ha expuesto anteriormente y la parte fraccionaria se multiplica por 2; la parte entera obtenida es la cifra más significativa del número. Si la parte fraccionaria restante se vuelve a multiplicar por 2, la nueva parte entera será la siguiente cifra más significativa, y así sucesivamente.

EJEMPLO:

Convertir el número $327,625)_{10}$ en binario. $327,625)_{10} = 327)_{10} + 0,625)_{10}$ La parte entera es $327)_{10}$ que, pasándola al binario, resulta



Para obtener la parte fraccionaria se procede de la siguiente manera:

$0,625$	$0,250$	$0,500$
$\times 2$	$\times 2$	$\times 2$
$1,250$	$0,500$	$1,000$

Por tanto, la parte fraccionaria será

$$0,625)_{10} = 0,101)_2$$

Entonces,

$$327,625)_{10} = 101000111,101)_2.$$



- CONVERSIÓN DE BINARIO A DECIMAL

Para convertir un número entero de base binaria en base decimal se recurre al polinomio equivalente, operando éste en modo decimal.

EJEMPLO:

$$\begin{aligned}
 1101,011_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} \\
 1 \cdot 2^3 &= 8 \\
 1 \cdot 2^2 &= 4 \\
 1 \cdot 2^0 &= 2 \\
 1 \cdot 2^{-2} &= 0,25 \\
 1 \cdot 2^{-3} &= \underline{0,125} \\
 &= 14,375
 \end{aligned}$$

Por tanto:

$$1101,011_2 = 14,375_{10}$$

- CONVERSIONES OCTAL-DECIMAL

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Esta conversión puede hacerse de la forma genérica (dividiendo entre 8 la parte entera y multiplicando por 8 la parte decimal) o de la siguiente forma:

Para pasar un número del sistema binario al octal se divide el número binario en grupos de tres dígitos partiendo de la coma hacia la izquierda y hacia la derecha, añadiendo ceros si fuese necesario para completar los grupos extremos, su suma ponderada dará un número del sistema octal. El número dado tendrá tantos dígitos en el sistema octal como grupos se hayan formado en el sistema binario,

EJEMPLO:

Sea el número 110011_2 si se divide en grupos de tres dígitos desde el dígito de menor peso.

$$\begin{aligned}
 110_2 &= 6_{10} = 6_8 \\
 011_2 &= 3_{10} = 3_8
 \end{aligned}$$

será:

$$\overbrace{110}^6 \overbrace{011_2}^3 = 63_8$$

EJEMPLO:

Sea el número $1101110,0101_2$ se divide en grupos de tres dígitos a partir de la coma completando con ceros, o sea:



$$1101110,0101)_2 = 101\ 110, 010\ 100)_2$$

Sustituyendo cada grupo por su valor: tendremos:

$$\overbrace{101}^5 \overbrace{110}^6, \overbrace{010}^2 \overbrace{100}^4 = 56,24)_8$$

- CONVERSIÓN DE DECIMAL A OCTAL

Para pasar un número del sistema decimal al sistema octal el método es similar al visto en el sistema binario. Veámoslo con un ejemplo.

EJEMPLO:

Sea el número $12342)_{10}$, que se desea pasar al sistema octal.

12342	8				
43	1542	8			
34	74	192	8		
22	22	32	24	8	
6	6	0	0	3	
\longleftarrow 30066_8					

Por tanto, $12342)_{10} = 30066)_8$.

Para saber más:

En caso de que el número decimal tenga una parte fraccionaria, se procede de la siguiente manera:

Sea:

$$12342,890625)_{10}$$

$$\begin{array}{r} 0.890625 \\ \times 8 \\ \hline \end{array}$$

$$\begin{array}{r} 7,125000 = 7 + 0,125000 \\ \times 8 \\ \hline \end{array}$$

$$1,000000$$

Entonces. $12342,890625)_{10} = 30066,71)_8$.

Si se quiere una aproximación de cuatro cifras y no se llega al producto 0, se toman los cuatro primeros enteros. Esta regla se aplica a las demás conversiones con parte fraccionaria.



- CONVERSIONES HEXADECIMAL-DECIMAL

Para convertir un número del sistema hexadecimal al decimal utilizamos el polinomio equivalente, teniendo en cuenta que los dígitos A, B, C, D, E y F han de ser sustituidos por 10, 11, 12, 13, 14 y 15 respectivamente.

EJEMPLO:

Consideremos $3AC_{16}$) para su transformación en su equivalente decimal; entonces:

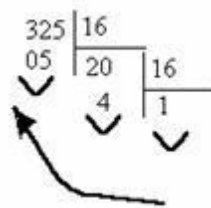
$$3AC_{16}) = 3 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0 = 768 + 160 + 12 = 940_{10}$$

- CONVERSIÓN DE DECIMAL A HEXADECIMAL

La conversión decimal - hexadecimal es parecida a la decimal - octal, solo que las divisiones sucesivas se hacen por 16 en lugar de 8.

EJEMPLO:

Pasar 325_{10}) a hexadecimal.



Por tanto $325_{10}) = 145_{16}$)

- CONVERSIÓN OCTAL-BINARIO-HEXADECIMAL

Para convertir un número del sistema octal al binario se pone el equivalente binario de cada una de las cifras con tres dígitos.

EJEMPLO:

Para obtener el correspondiente en binario del número 52_8), tenemos

$$5 = 101$$

$$2 = 010$$

Por tanto, $52_8) = 101010_2$).

CONVERSIÓN DE HEXADECIMAL A BINARIO

Para pasar un número del sistema hexadecimal al binario tendremos en cuenta que cada grupo de cuatro cifras binarias equivale a una cifra hexadecimal, por lo que las sustituiremos por el correspondiente símbolo hexadecimal. Por consiguiente, para convertir un número hexadecimal en otro binario bastará con sustituir cada valor hexadecimal por los cuatro dígitos binarios equivalentes.



EJEMPLO:

Sea el número hexadecimal $DF4_{16}$, entonces:

$$\begin{aligned} D_{16} &= 13_{10} &= & 1101_2 \\ F_{16} &= 15_{10} &= & 1111_2 \\ 4_{16} &= 4_{10} &= & 0100_2 \end{aligned}$$

Nº	Hexadecimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	0111



Los sistemas de codificación alfanumérica más importantes son:

LA CODIFICACIÓN ALFANUMÉRICA:

Los datos además de numéricos, pueden ser alfabéticos o alfanuméricos. Con los datos alfanuméricos podemos construir instrucciones y programas. Es lógico pensar que el ordenador no solamente procesará datos numéricos, sino también datos alfabéticos y combinaciones de los anteriores.

- **ASCII:** este sistema utiliza una combinación de 7 u 8 bits dependiendo del fabricante para representar cada símbolo. Es el más utilizado el que emplea símbolos diferentes (28). Con este código se pueden representar dígitos del 0 al 9, letras mayúsculas de la A a la Z, letras minúsculas, caracteres especiales y algunos otros denominados de control

Caracteres de control ASCII				Caracteres ASCII imprimibles						ASCII extendido														
DEC	HEX	Símbolo ASCII		DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo			
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ó
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ó	194	C2h	ł	226	E2h	ô
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	ł	227	E3h	ö
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ü	196	C4h	ł	228	E4h	õ
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	à	165	A5h	Ñ	197	C5h	ł	229	E5h	Ö
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	á	166	A6h	°	198	C6h	ł	230	E6h	µ
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	ª	199	C7h	ł	231	E7h	¶
08	08h	BS	(retroceso)	40	28h	(72	48h	H	104	68h	h	136	88h	ê	168	A8h	¿	200	C8h	ł	232	E8h	þ
09	09h	HT	(tab horizontal)	41	29h)	73	49h	I	105	69h	i	137	89h	ë	169	A9h	©	201	C9h	ł	233	E9h	Û
10	0Ah	LF	(salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh	¬	202	CAh	ł	234	EAh	Ü
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ì	171	ABh	½	203	CBh	ł	235	EBh	Ý
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	í	172	ACH	¼	204	CAh	ł	236	ECh	ÿ
13	0Dh	CR	(retorno de carro)	45	2Dh	.	77	4Dh	M	109	6Dh	m	141	8Dh	ì	173	ADh	»	205	CDh	ł	237	EDh	ÿ
14	0Eh	SO	(shift Out)	46	2Eh	-	78	4Eh	N	110	6Eh	n	142	8Eh	Ë	174	AEh	«	206	CEh	ł	238	EEh	·
15	0Fh	SI	(shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	À	175	AFh	»	207	CFh	ł	239	EFh	·
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	É	176	B0h	»	208	D0h	ł	240	F0h	±
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	»	209	D1h	ł	241	F1h	±
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	»	210	D2h	ł	242	F2h	±
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h	»	211	D3h	ł	243	F3h	±
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ö	180	B4h	»	212	D4h	ł	244	F4h	±
21	15h	NAK	(negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	ò	181	B5h	»	213	D5h	ł	245	F5h	±
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	ú	182	B6h	»	214	D6h	ł	246	F6h	±
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	ù	183	B7h	»	215	D7h	ł	247	F7h	±
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	»	216	D8h	ł	248	F8h	±
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	Û	185	B9h	»	217	D9h	ł	249	F9h	±
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	Ü	186	BAh	»	218	DAh	ł	250	FAh	±
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ø	187	BBh	»	219	DBh	ł	251	FBh	±
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch	}	156	9Ch	£	188	BCh	»	220	DCh	ł	252	FCh	±
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	Ø	189	BDh	»	221	DDh	ł	253	FDh	±
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	»	222	DEh	ł	254	FEh	±
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	-				159	9Fh	f	191	BFh	»	223	DFh	ł	255	FFh	±
127	20h	DEL	(delete)																					

- **EBCDIC:** cada símbolo se representa por una combinación de 8 bits agrupados en dos bloques de cuatro. Es el formato extendido del BCD
- **UNICODE:** es un código internacional utilizado hoy en día en la mayoría de los sistemas operativos. Permite que un producto software o página Web específica se oriente a múltiples plataformas, idiomas o países sin necesidad de rediseño. Concretamente ASCII tiene una tabla específica para cada país ya que los diferentes símbolos de todos los países no cabrían en una tabla.



MEDIDAS DE LA INFORMACIÓN

Veamos los más utilizados:

- **bit** (b).- es una señal electrónica que puede estar encendida (1) o apagada (0).
- **Byte** (B).- Formado normalmente por un octeto (8 bits).
- **Kilobyte** (K o KB).- Aunque se utilizan las acepciones utilizadas en el sistema internacional, un Kilobyte no son 1.000 bytes. Debido a lo anteriormente expuesto, un KB (Kilobyte) son 1.024 bytes. Debido al mal uso de este prefijo (Kilo, proveniente del griego, que significa mil), se está utilizando cada vez más el término definido por el IEC (Comisión Internacional de Electrónica) Kibi o KiB para designar esta unidad.
- **Megabyte** (MB).- El MB es la unidad de capacidad más utilizada en Informática. Un MB NO son 1.000 KB, sino 1.024 KB, por lo que un MB son 1.048.576 bytes. Al igual que ocurre con el KB, dado el mal uso del término, cada vez se está empleando más el término MiB.
- **Gigabyte** (GB).- Un GB son 1.024 MB (o MiB), por lo tanto 1.048.576 KB. Llegados a este punto en el que las diferencias sí que son grandes, hay que tener muy en cuenta (sobre todo en las capacidades de los discos duros) que es lo que realmente estamos comprando. Algunos fabricantes utilizan el término GB refiriéndose no a 1.024 MB, sino a 1.000 MB, lo que representa una pérdida de capacidad en la compra. Otros fabricantes sí que están ya utilizando el término GiB. Para que nos hagamos un poco la idea de la diferencia entre ambos, un disco duro de 250 GB (SI) en realidad tiene 232.50 GiB.
- **Terabyte** (TB).- Aunque es aún una medida poco utilizada, pronto nos tendremos que acostumbrar a ella, ya que los discos duros ya se están fabricando en esta medida. Un Terabyte son 1.024 GB.
- Existen unas medidas superiores, como el **Petabyte**, **Exabyte**, **Zettabyte** o el **Yottabite**, que podemos calcular multiplicando por 1.024 la medida anterior.