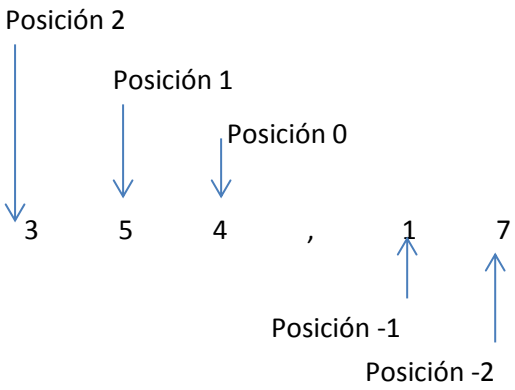
SISTEMAS DE NUMERACIÓN

Es un conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen dicha representación.

Un elemento importante en el sistema de numeración es la **base**, que determina el número de símbolos para representar los números. Así, el sistema más utilizado diariamente es el **sistema decimal**, que utiliza **10** caracteres. En la siguiente tabla se muestra los diferentes sistemas de numeración más representativos.

Sistemas	Base	Dígitos
Binario	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F Siendo las letras la representación de los valores comprendidos entre el 10 y el 15 del sistema decimal.

Es importante destacar que en todos los sistemas tiene importancia la posición del dígito frente a la coma decimal:



Con tantos sistemas de numeración, nosotros podemos representar un número cualquiera en todos ellos, de tal manera que por ejemplo en octal tenga un valor y al pasarlo a decimal tenga otro. ¿Cómo pasamos un número de un sistema a otro? Mediante la conversión entre sistemas.

- Conversión de una base cualquiera a base decimal.

Mediante el teorema fundamental de la numeración podemos convertir una cantidad expresada en cualquier base a base decimal:

$$\textbf{Número decimal} = \sum(\text{dígito})_i \ X \ (\text{base})^i$$

Siendo i la posición del dígito respecto a la coma decimal.

Ejemplo:

El número $23'5_8$ que está en octal lo vamos a pasar a decimal:

$\boldsymbol{2}x8^1 + \boldsymbol{3}x8^0 + \boldsymbol{5}x8^{-1} = 16 + 3 + 0'125 = 19'625_{10}$

El número $101'01_2$ que está en binario lo vamos a pasar a decimal:

$1x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} = 4 + 0 + 1 + 0 + 0.25 = 5'25_{10}$

El número $B3'F_{16}$ que está en hexadecimal lo vamos a pasar a decimal:

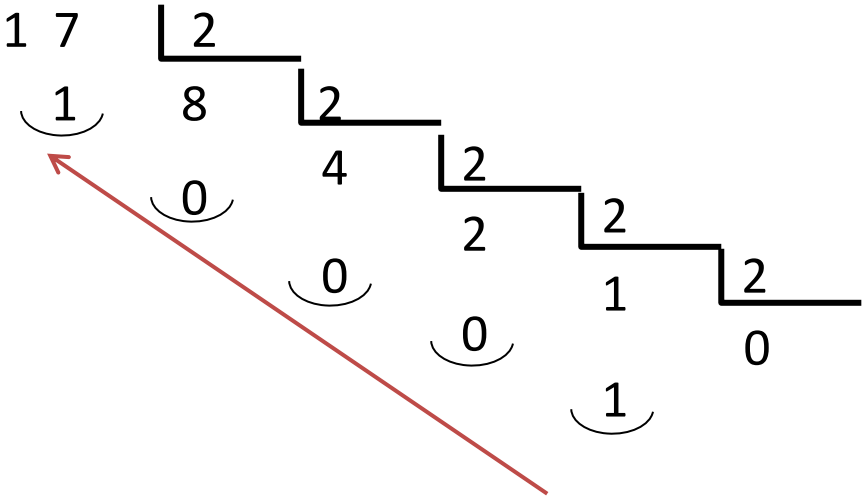
$11x16^1 + 3x16^0 + 15x16^{-1} = 176 + 3 + 0'9375 = 179'9375_{10}$

- Conversión de una base decimal a otra cualquiera.

La técnica más utilizada para convertir un número de decimal a otro sistema es multiplicar su parte decimal y dividir su parte entera por la base del sistema al que queremos convertir el número.

Vamos a convertir el número $17{,}125_{10}$ a binario.

Lo primero es coger la parte entera (17) y dividirla por la base del sistema al que queremos convertir. En este caso al ser binario, lo dividimos por 2 (si fuera a octal por 8; hexadecimal por 16). Hay que ir dividiendo el cociente una y otra vez hasta que sea menor que el divisor (en nuestro caso 2)

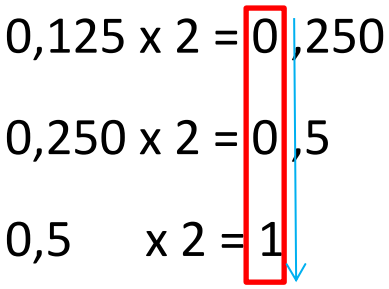


La parte entera (17) en binario será los valores de los restos en orden inverso.

1 0 0 0 1

Lo siguiente es convertir la parte decimal del número (0,125). Para ello iremos multiplicando la cantidad por la base (2). Sobre la parte decimal del resultado volvemos a multiplicar por la base, y así una y otra vez hasta que o bien en una de las multiplicaciones el resultado no tenga

parte decimal (es decir, sólo tenga parte entera) o veamos que se produce un ciclo de resultados (es decir, un bucle infinito).



El número convertido será los dígitos de las partes enteras del resultado de las multiplicaciones de arriba abajo.

0 0 1

Por tanteo, el número $17{,}125_{10}$ en decimal equivale al $10001'001_2$ en binario.

- Conversión de octal/hexadecimal-binario.

Cuando tenemos un dígito a octal y lo queremos pasar a hexadecimal o viceversa, primero lo convertiremos a binario y después a hexadecimal u octal. Por tanto, el convertir a binario nuestro número será un paso intermedio.

Tabla de equivalencia de los números de octal a binario:

Dígito octal	Dígito binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tabla de equivalencia de los números de hexadecimal a binario:

Dígito hexadecimal	Dígito binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Nótese que para poder representar los dígitos en octal necesito 3 dígitos en binario y para hacer lo propio en hexadecimal necesito 4. Aquí es donde radica el truco de este método.

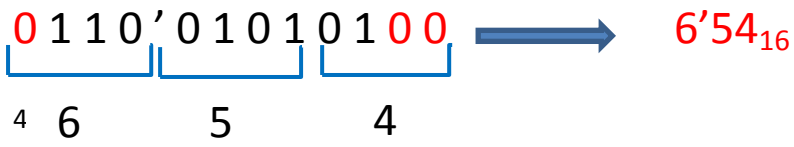
Pongamos el caso que quiero convertir $6'25_8$ que está en octal a hexadecimal. Bien, como hemos dicho anteriormente, primero tenemos que convertir el número a binario. Para ello lo único que tenemos que hacer es ir dígito a dígito de nuestro número en octal y convertirlo a binario fijándonos en la tabla de equivalencia.

$$6_8 = 110 \quad 2_8 = 010 \quad 5_8 = 101$$

Ahora juntamos los números resultantes en binario en su orden y separados por la coma decimal donde tercie.

$$1\,1\,0\,'\,0\,1\,0\,1\,0\,1_2$$

Ahora, una vez que tenemos nuestro número en binario, lo que vamos a hacer es pasarlo a hexadecimal. Para ello nos posicionamos en la coma decimal y empezamos a agrupar los dígitos de cuatro en cuatro, primero la parte entera. Una vez agrupada la parte entera hacemos lo propio con la parte decimal. Recuerda siempre empezando a partir de la coma decimal y si al final te falta algún dígito para hacer un grupo de 4, rellénalo con tantos ceros como necesites. Una vez tengas hecho los grupos lo único que tenemos que hacer es irnos a la tabla de equivalencia y convertir cada grupo de 4 dígitos a su valor en hexadecimal. Como resultado obtendremos el valor en hexadecimal.



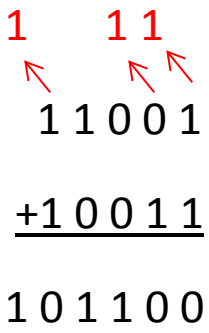
- Operaciones aritméticas en binario: sumas y restas.

1. *Suma:*

Al igual que en la decimal, hay que tener en cuenta que influye la posición del número respecto a la coma decimal. De tal manera que el primer dígito a la izquierda de la coma representa las unidades, el segundo las decenas, el tercero las centenas, etc. Cuando sumas un número y no cabe en esa unidad, se pasa el sobrante a la siguiente unidad superior (es decir, de las unidades a las decenas, de las decenas a las centenas y así sucesivamente).

Hay que tener en cuenta las tablas de la suma:

Tabla del 0	Tabla del 1
0 + 0 = 0	1 + 0 = 1
0 + 1 = 0	1 + 1 = 0 y me llevo 1 a la siguiente unidad

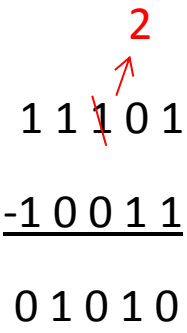


2. *Resta:*

Como sucede en la suma, hay que tener en cuenta la posición de los dígitos respecto a la coma decimal. Cuando el dígito superior es menor que el dígito inferior, se produce un acarreo de tal manera que traes una unidad del dígito superior.

Hay que tener en cuenta las tablas de la resta:

Tabla del 0	Tabla del 1
0 - 0 = 0	1 - 0 = 1
0 - 1 = no cabe (me llevo 1 de la unidad superior)	1 - 1 = 0



Representación interna de la información

Para almacenar los caracteres que forman el alfabeto se utilizan los códigos de E/S que traducen la información o los datos que nosotros podemos entender a una representación que la máquina puede interpretar y procesar. Los códigos estandarizados que se utilizan son el BCD, EBCDIC, ASCII y Unicode.

Actualmente, el Unicode (Unicode Standard) es el más extendido, se usa en los ordenadores bajo Windows y en los navegadores Internet Explorer y Netscape a partir de su versión 4. Utiliza 16 bits, lo que permite codificar todos los caracteres de cualquier lenguaje, hasta 65 536.

La información almacenada en un ordenador ocupa un espacio, es decir, una serie de bits. Pero como el bit es una unidad muy pequeña se utilizan unidades más grandes. Para empezar el byte u octeto, formado por 8 bits. Y también sus múltiplos:

Nombre (símbolo)	Sistema Internacional de Unidades (SI) Estándar (uso decimal)	Prefijo binario (uso binario)	Nombre (símbolo)
Kilobyte (kB)	$1000^1 = 10^3$ bytes	$1024^1 = 2^{10}$ bytes	Kibibyte (kib)
Megabyte (MB)	$1000^2 = 10^6$ bytes	$1024^2 = 2^{20}$ bytes	Mebibyte (Mib)
Gigabyte (GB)	$1000^3 = 10^9$ bytes	$1024^3 = 2^{30}$ bytes	Gibibyte (Gib)
Terabyte (TB)	$1000^4 = 10^{12}$ bytes	$1024^4 = 2^{40}$ bytes	Tebibyte(Tib)
Petabyte (PB)	$1000^5 = 10^{15}$ bytes	$1024^5 = 2^{50}$ bytes	Pebibyte (Pib)
Exabyte (EB)	$1000^6 = 10^{18}$ bytes	$1024^6 = 2^{60}$ bytes	Exbibyte (Eib)
Zettabyte (ZB)	$1000^7 = 10^{21}$ bytes	$1024^7 = 2^{70}$ bytes	Zebibyte (Zib)
Yottabyte (YB)	$1000^8 = 10^{24}$ bytes	$1024^8 = 2^{80}$ bytes	Yobibyte (Yib)

A modo de ejemplo, podemos ver, en la siguiente tabla, la correspondencia entre números y caracteres del código ASCII.

Código decimal	Carácter ASCII	Código decimal	Carácter ASCII	Código decimal	Carácter ASCII	Código decimal	Carácter ASCII	Código decimal	Carácter ASCII	Código decimal	Carácter ASCII
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	_
48	0	64	@	80	P	96	`	112	p		