

# TEMA 1:

# LA INFORMACIÓN Y LOS SISTEMAS DE NUMERACIÓN



## **Índice**

1. Introducción
2. El sistema decimal
3. El sistema binario
4. El sistema octal
5. El sistema hexadecimal
6. Conversión entre los distintos sistemas
  - 6.1. De Decimal a Binario
  - 6.2. De Binario a Decimal
  - 6.3. De Binario a Octal
  - 6.4. De Binario a Hexadecimal
  - 6.5. Octal a Binario
  - 6.6. Octal a Hexadecimal
  - 6.7. Hexadecimal a Binario
  - 6.8. Hexadecimal a Octal
7. Suma de números binarios
8. Producto de números binarios
9. Representación interna de la información
10. Arquitecturas de 4, 8, 16, 32 y 64 bits
11. Representación de datos alfabéticos y alfanuméricos
  - 11.1. ASCII
  - 11.2. UNICODE

## TEMA 1: LA INFORMACIÓN Y LOS SISTEMAS DE NUMERACIÓN

### 1. Introducción.

El ser humano siempre ha tenido la necesidad de comunicarse. Además, han tenido que comunicarse a grandes distancias y con rapidez. Desde la antigüedad se han ido ingeniando diferentes métodos para conseguirlo, señales de humo, espejos y poco a poco han ido evolucionando llegando al Morse, teléfono, etc...

Los ordenadores debido a sus construcciones solamente nos podemos comunicar con ellos en forma binaria. Es decir, un ordenador está compuesto de circuitos electrónicos los cuales solo pueden reconocer si hay o no hay corriente en esos circuitos, por lo tanto, solo reconocen dos valores, "1" si hay tensión en un punto o un "0" si no hay tensión denominándolo *lógica positiva* (la más utilizada) o el caso contrario un "0" cuando hay tensión y un "1" si no hay tensión denominándolo *lógica negativa*.

El ordenador para comunicarse con nosotros no utiliza el sistema binario, utiliza otros *sistemas de numeración*<sup>1</sup> que son el octal, hexadecimal y decimal.

### 2. El sistema decimal

Es uno de los sistemas denominados posicionales, que utiliza un conjunto de símbolos cuyo significado o valor depende de su posición relativa al punto decimal. Los símbolos que utiliza para la representación son: 0 1 2 3 4 5 6 7 8 9

Un determinado valor o cantidad, que denominaremos número decimal, se puede expresar de la siguiente forma según el teorema fundamental de la numeración:

$$N^o = \sum_{i=m}^n (Dígito)_i * (base)^i$$

Donde:

Base = 10

i = Posición respecto a la coma

m = Número de dígitos a la derecha de la coma

n = Número de dígitos a la izquierda de la coma menos 1

Dígito = Cada uno de los que componen el número

<sup>1</sup> **Sistema de numeración** es el conjunto de símbolos y reglas que se utilizan para la representación de datos numéricos o cantidades.

### 3. El sistema binario

Es el sistema que utilizan internamente los circuitos digitales que configuran el hardware de los ordenadores.

Los símbolos que utiliza para la representación son: 0 1

Cada cifra o dígito de un número representado en este sistema se denomina *bit*.

### 4. El sistema octal

Se trata de un sistema de numeración de base 8 que utiliza los siguientes símbolos:

0 1 2 3 4 5 6 7

Es un sistema de los llamados posicionales.

### 5. El sistema hexadecimal

Es un sistema de numeración de base 16 en el que se utilizan los siguientes símbolos:

0 1 2 3 4 5 6 7 8 9 A B C D E F

También es uno de los sistemas denominados posicionales.

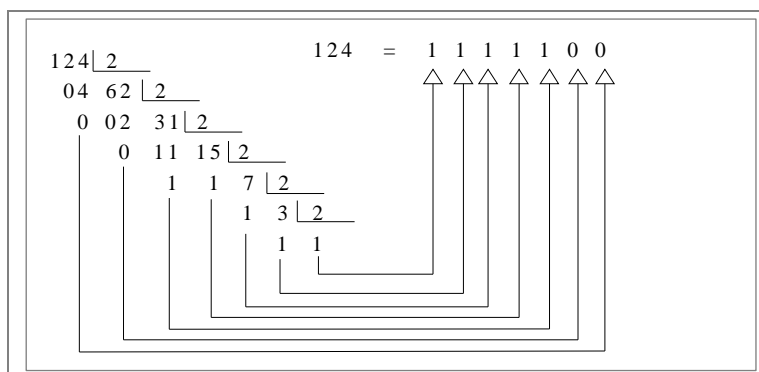
A los símbolos A B C D E F se le asignan los valores absolutos decimales 10, 11, 12, 13, 14, 15, respectivamente.

### 6. Conversión entre los distintos sistemas.

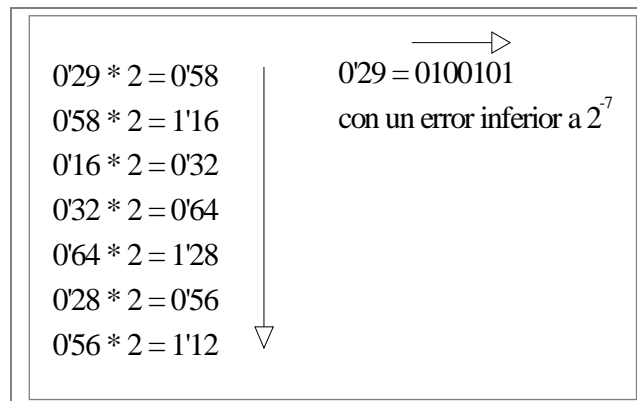
#### 6.1. De Decimal a Binario

La primera forma tendremos que tener en cuenta si es la parte entera o decimal de un número concreto por ejemplo vamos a convertir el número 124'29

Cogemos primero la parte entera 124 y hacemos divisiones sucesivas entre 2



Ahora vamos a trabajar con la parte decimal



Por tanto el  $124.29_{(10)} = 1111100.0100101_{(2)}$

## 6.2. De Binario a Decimal

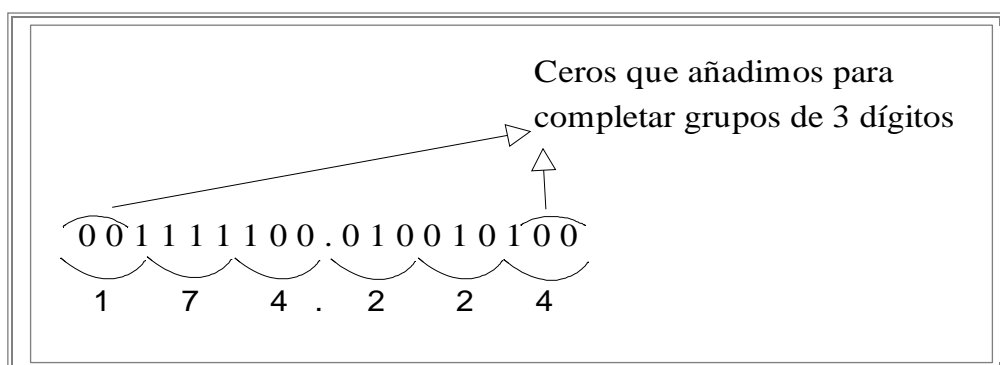
Método de las sumas de las potencias de 2. Este método es la aplicación directa del teorema fundamental de la numeración.

$$\begin{aligned}
 1111100 &= 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = \\
 &= 64 + 32 + 16 + 8 + 4 + 0 + 0 = \underline{124}
 \end{aligned}$$

Si el número binario tuviese parte decimal se haría igual con los exponentes en negativo, estando el exponente -1 inmediatamente después del exponente 0.

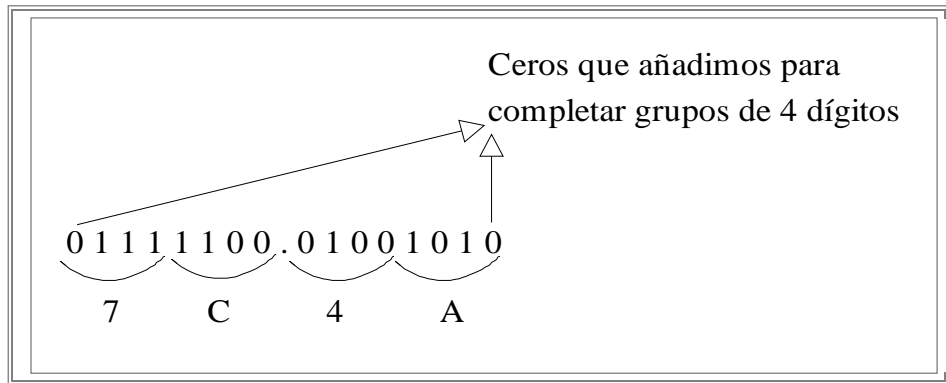
## 6.3. De Binario a Octal

Para convertir número binarios a octal se agrupan los dígitos binarios de 3 en 3 tomando como punto de inicio el punto decimal (tanto a derecha como a izquierda).



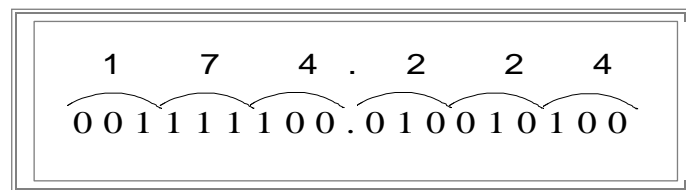
#### 6.4. De Binario a Hexadecimal

Para convertir números binarios a hexadecimales se realiza el siguiente proceso: Se agrupan de 4 en 4 a partir del punto decimal hacia la izquierda y hacia la derecha sustituyendo cada cuarteto por su correspondiente dígito hexadecimal.



#### 6.5. Octal a Binario

Tenemos que seguir los pasos inversos a la conversión de binario a octal, es decir, cada dígito lo representaremos con tres dígitos en binario.

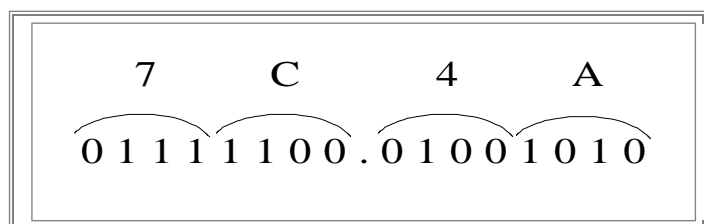


#### 6.6 Octal a Hexadecimal

Primero realizamos la conversión de octal a binario y después de binario a hexadecimal.

#### 6.7. Hexadecimal a Binario

Esta conversión se hace de forma inversa a como se hizo la conversión de binario a hexadecimal, es decir, se representan los caracteres en base 16 en conjuntos de 4 dígitos.



## 6.8. Hexadecimal a Octal

Para realizar la conversión a octal primero tenemos que hacer la conversión de hexadecimal a binario y a continuación la conversión de binario a octal.

Fuente: Apuntes de Luis Estañ

## 7. Suma de números binarios

La tabla de sumar para números binarios es la siguiente:

+	0	1
0	0	1
1	1	10

Las posibles combinaciones al sumar dos bits son:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 10$

Al sumar  $1 + 1$  es 10, es decir, llevamos 1 a la siguiente posición de la izquierda (acarreo). Esto es equivalente, en el sistema decimal a sumar  $9 + 1$ , que da 10: cero en la posición que estamos sumando y un 1 de acarreo a la siguiente posición.

Ejemplo:

$$\begin{array}{r}
 1 \\
 10011000 \\
 + 00010101 \\
 \hline
 10101101
 \end{array}$$

Fuente: [http://es.wikipedia.org/wiki/Sistema\\_binario](http://es.wikipedia.org/wiki/Sistema_binario)

## 8. Producto de números binarios

La tabla de multiplicar para números binarios es la siguiente:

.	0	1
0	0	0
1	0	1

El algoritmo del producto en binario es igual que en números decimales; aunque se lleva a cabo con más sencillez, ya que el 0 multiplicado por cualquier número da 0, y el 1 es el elemento neutro del producto.

Por ejemplo, multipliquemos 10110 por 1001:

$$\begin{array}{r}
 10110 \\
 * 1001 \\
 \hline
 10110 \\
 00000 \\
 00000 \\
 10110 \\
 \hline
 11000110
 \end{array}$$

Fuente: [http://es.wikipedia.org/wiki/Sistema\\_binario](http://es.wikipedia.org/wiki/Sistema_binario)

## 9. Representación interna de la información

Bit es el acrónimo binary digit (dígito binario). Un bit es un dígito del sistema de numeración binario.

Los circuitos digitales internos que componen las computadoras utilizan el sistema de numeración Binario para la interpretación de la información.

Para medir la cantidad de información representada en binario se utilizan múltiplos que a diferencia de otras magnitudes físicas, utilizan el factor multiplicador 1024 en lugar de 1000, debido a que es el múltiplo de 2 más cercano a este último ( $2^{10}=1024$ ).

Tablas de Referencia:

Nombre	Abrev.	Factor binario	Tamaño en el SI
bytes	B	$2^0 = 1$	$10^0 = 1$
kilo	k	$2^{10} = 1024$	$10^3 = 1000$
mega	M	$2^{20} = 1\,048\,576$	$10^6 = 1\,000\,000$
giga	G	$2^{30} = 1\,073\,741\,824$	$10^9 = 1\,000\,000\,000$
tera	T	$2^{40} = 1\,099\,511\,627\,776$	$10^{12} = 1\,000\,000\,000\,000$
peta	P	$2^{50} = 1\,125\,899\,906\,842\,624$	$10^{15} = 1\,000\,000\,000\,000\,000$
exa	E	$2^{60} = 1\,152\,921\,504\,606\,846\,976$	$10^{18} = 1\,000\,000\,000\,000\,000\,000$
zetta	Z	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$	$10^{21} = 1\,000\,000\,000\,000\,000\,000\,000$
yotta	Y	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$	$10^{24} = 1\,000\,000\,000\,000\,000\,000\,000\,000$



Unidades básicas de información (en bytes)				
Prefijos del Sistema Internacional			Prefijo binario	
Múltiplo - (Símbolo)	Estándar <u>SI</u>	<u>Binario</u>	Múltiplo - (Símbolo)	Valor
<a href="#">kilobyte</a> (kB)	$10^3$	$2^{10}$	<b>kibibyte</b> (KiB)	$2^{10}$
<a href="#">megabyte</a> (MB)	$10^6$	$2^{20}$	<a href="#">mebibyte</a> (MiB)	$2^{20}$
<a href="#">gigabyte</a> (GB)	$10^9$	$2^{30}$	<a href="#">gibibyte</a> (GiB)	$2^{30}$
<a href="#">terabyte</a> (TB)	$10^{12}$	$2^{40}$	<a href="#">tebibyte</a> (TiB)	$2^{40}$
<a href="#">petabyte</a> (PB)	$10^{15}$	$2^{50}$	<a href="#">pebibyte</a> (PiB)	$2^{50}$
<a href="#">exabyte</a> (EB)	$10^{18}$	$2^{60}$	<a href="#">exbibyte</a> (EiB)	$2^{60}$
<a href="#">zettabyte</a> (ZB)	$10^{21}$	$2^{70}$	<a href="#">zebibyte</a> (ZiB)	$2^{70}$
<a href="#">yottabyte</a> (YB)	$10^{24}$	$2^{80}$	<a href="#">yobibyte</a> (YiB)	$2^{80}$

1 Byte = 8 bits

Los fabricantes de dispositivos de almacenamiento habitualmente usan los prefijos decimales del **S.I.**, por lo que un disco duro de 30 GB tiene una capacidad aproximada de  $28 * 2^{30}$  bytes, lo que serían 28 GiB

Fuentes:

<http://es.wikipedia.org/wiki/Bit>

<http://www.monografias.com/trabajos16/representacion-informacion/representacion-informacion.shtml>

[http://www.frrg.utn.edu.ar/frrg/apuntes/programacion/sist\\_proc\\_datos/unidades\\_medidas.pdf](http://www.frrg.utn.edu.ar/frrg/apuntes/programacion/sist_proc_datos/unidades_medidas.pdf)

<http://es.wikipedia.org/wiki/Byte>

[http://es.wikipedia.org/wiki/Prefijo\\_binario](http://es.wikipedia.org/wiki/Prefijo_binario)

## 10. Arquitecturas de 4, 8, 16, 32 y 64 bits

Cuando se habla de CPU's o microprocesadores de 4, 8, 16, 32, 64 bits, se refiere al tamaño, en número de bits, que tienen los registros internos del procesador y también a la capacidad de procesamiento de la Unidad aritmético lógica (ALU). Un microprocesador de 4 bits tiene registros de 4 bits y, mientras que un procesador de 8 bits tiene registros y procesa los datos en grupos de 8 bits.

Los **registros** son celdas de memoria de alta capacidad que almacenan temporalmente datos mientras se realiza alguna operación

Los procesadores de 16, 32 y 64 bits tienen registros y ALU de 16, 32 y 64 bits respectivamente, y generalmente pueden procesar los datos, según el tamaño en bits de sus registros

Cuando se habla de procesadores de, digamos 32 bits, nos referimos a su capacidad de procesar datos en hasta 32 bits simultáneamente. La denominación de "microprocesador de 32 bits" se refiere a su capacidad de trabajar normalmente con los datos en el número máximo de bits

Por tanto, podemos decir que el **ancho de palabra**: Indica el número de bits que maneja en paralelo el computador. Cuanto mayor sea más potencia de cálculo tendrá el computador

Fuentes:

<http://es.wikipedia.org/wiki/Bit>

<http://html.rincondelvago.com/arquitectura-de-los-computadores.html>

## 11. Representación de datos alfabéticos y alfanuméricos

La información que un usuario introduce en un ordenador está formada por letras, números y otros signos.

Para almacenar esa información en la máquina se usan estándares de codificación como son:

- ASCII
- UNICODE
- BCD
- EBCDIC

Nos centraremos en las 2 primeros. ASCII y UNICODE

### 11.1. ASCII

ASCII (acrónimo inglés de American Standard Code for Information Interchange — Código Estándar Americano para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales.

Existen dos variantes:

**ASCII:** 1 carácter => 7 bits (128 caracteres diferentes)

**ASCII extendido:** 1 carácter => 8 bits (256 caracteres diferentes) Incluye símbolos de lenguajes extranjeros y símbolos gráficos.  
(Ver tabla ASCII en el Anexo I)

### 11.2. UNICODE

El Estándar Unicode es un estándar de codificación de caracteres que proporciona un número único para cada carácter, sin importar la plataforma, sin importar el programa, sin importar el idioma.

Es el más moderno y extendido ya que está orientado a múltiples plataformas, idiomas y países. Permite la compatibilidad del software y los sitios webs.

Unicode representa cada carácter como un número de 2 bytes (16 bits), de 0 a 65535

Fuente:

<http://unicode.org/standard/translations/spanish.html>

<http://es.wikipedia.org/wiki/Unicode>

[http://www.gulic.org/almacen/diveintopython-5.4-es/xml\\_processing/unicode.html](http://www.gulic.org/almacen/diveintopython-5.4-es/xml_processing/unicode.html)

<http://es.debugmodeon.com/articulo/todo-lo-que-deberias-saber-de-unicode>

**Anexo I: tabla ASCII**

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Fuente:

<http://asciicodes.blogspot.com/>