UD 1 REPRESENTACIÓN DE LA INFORMACIÓN.

0.- CONTENIDOS

- Informática e información.
- Sistemas de numeración.
- Sistema Binario, operaciones lógicas y aritméticas en binario
- Sistemas Octal y Hexadecimal.
- Conversiones entre sistemas.
- Representación interna de la información.
- Unidades de medidas de la información.
- Representación de datos alfabéticos: códigos BCD, ASCII, EBCDIC y UNICODE

1.- OBJETIVOS.

- Operar de la misma forma que lo hacen los ordenadores internamente.
- Comprender el funcionamiento interno de los ordenadores.
- Interpretar y diferenciar los distintos códigos que utilizan los ordenadores.

2.- INFORMÁTICA E INFORMACIÓN.

La **informática** es la ciencia tecnológica que estudia el tratamiento ***automático y racional*** de la información, con el fin de obtener de ella la máxima utilidad. La informática utiliza los ordenadores para el tratamiento y el proceso de la información. Pero ¿qué es la información?

La **información** es sinónimo de conocimiento, de noticia, de datos, etc. podemos pensar que hay información cuando existe una comunicación, pero no es así. En un proceso de comunicación, la información que se adquiere depende mucho del receptor. Podemos tener receptores que no conocían la información, a estos les llegará toda la información, pero a los receptores que ya conocían la información lo que les llega es menor y si nos encontramos con receptores que no conocen el idioma, a estos no les llegará nada de información.

La **comunicación** es el proceso mediante el cual una entidad transmite información a otra con el objeto de ponerla en su conocimiento. En este proceso intervienen los siguientes elementos:

- ***Emisor***, fuente o transmisor: entidad que emite o genera la información.
- ***Receptor***: entidad que recibe la información.
- ***Mensaje***: información que el emisor transmite al receptor.

- ***Medio o Canal***: vía por la que se transmite el mensaje.
- ***Código***: conjunto de signos, reglas y normas (lenguaje) que se emplean para construir el mensaje.

El emisor y el receptor pueden intercambiar sus papeles o incluso realizar ambos papeles de forma simultánea.

En el proceso de comunicación interviene un factor adicional denominado **ruido**. Se considera ruido cualquier interferencia que contamine tanto el mensaje como el canal o el código.

Podemos entender la transmisión de la información entre el ser humano y el ordenador como una comunicación en la que el emisor es una persona y el receptor, el ordenador, o viceversa, y el medio o canal son los periféricos de entrada y salida del ordenador, que son los dispositivos que se conectan al ordenador y que van a permitir introducir datos para que el ordenador los procese y trasforme en forma de información.

Para que exista información es necesario que el que envía los datos y el que los recibe se entiendan, es decir, que utilicen el mismo código; de lo contrario, necesitarán un traductor de un código al otro.

En la transmisión de la información entre el ser humano y el ordenador puede hacerse de muchas formas:

- Mediante caracteres alfanuméricos: los caracteres que introducimos mediante el teclado.
- Mediante sonidos, como los introducidos a través de un micrófono o el que sale por los altavoces.
- Mediante vídeos, como imágenes obtenidas a través de una cámara de vídeo.
- Mediante gráficos e imágenes, introducidas por medio de un escáner, fotografías descargadas de una cámara fotográfica, etc.
- En general, cualquier tipo de datos enviados por un periférico del ordenador capaz de tomar datos de cualquier tipo y enviarlos al ordenador, o a la inversa.

En cada caso el canal es diferente y para proceder a la comunicación de los datos es necesario cambiar la forma en que estos se representan. Podría haber hasta tres formas de representación: la del emisor, la del canal y la del receptor.

Por tanto, los datos deben ser traducidos o codificados. La traducción es necesaria cuando los códigos utilizados por el emisor el canal y el receptor son diferentes.

2.1.- Simbología y codificación.

A lo largo de la historia el hombre ha utilizado una cantidad de simbología para comunicarse. Tenemos los símbolos egipcios, los mayas, etc.

Al conjunto de símbolos y reglas establecidas para la transformación de información es a lo que se llama **codificación**. Por tanto, **codificar** será la transformación de unos datos a una representación predefinida y preestablecida. El abecedario es un sistema de codificación que se desarrolló para ser usado en un medio plano como el papel y poder transmitir información a otras personas, quienes la descodifican y la convierten en pensamiento o ideas. Otro ejemplo de codificación es el Morse para el telégrafo.

Código es el conjunto de condiciones y convenios que permiten transformarla información de una representación concreta a otra. De tal forma que un código está compuesto por:

- Un conjunto de reglas y convenios de transformación del alfabeto fuente.
- Un nuevo alfabeto que sustituirá al original.

La representación interna de la información en los ordenadores ha de darse en forma de impulsos eléctricos; esto se efectúa empleando señales biestables con dos posibles estados, activado-desactivado, encendido-apagado, abierto-cerrado, es decir, hay impulso o no lo hay. Por eso, tendremos que codificar la información utilizando un código con dos únicos símbolos que representen los dos estados: 1 para indicar que hay impulso y el 0 para indicar que no lo hay; todo el lenguaje se transcribirá a combinaciones de ceros y unos para que el ordenador lo pueda interpretar.

Este código es el **código binario**, que está basado en el sistema de numeración binario, cuyos símbolos son el 0 y el 1.

Éstos suelen representarse con señales electromagnéticas, tales como la tensión eléctrica o la luz. Una señal es una magnitud (algo que puede medirse) física que varía con el tiempo.

Esta representación es muy adecuada porque permite una implementación e integración correcta con la tecnología del silicio y un tratamiento matemático potente al digitalizar la información. La digitalización facilita la comunicación al poder disponerse de equipos electrónicos que realizan las tareas propias de la comunicación de forma sencilla y eficiente.

3.- SISTEMAS DE NUMERACIÓN.

Un **sistema de numeración** es el conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen dicha representación.

Un sistema de numeración se distingue por su ***base***, que es el número de símbolos que utiliza, y se caracteriza por ser el coeficiente que determina cual es el valor de cada símbolo dependiendo de su posición.

El sistema de numeración que utilizamos normalmente es el sistema decimal, de base 10. El ***sistema decimal*** utiliza diez dígitos o símbolos: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.

Otras bases usadas en TIC son:

- Base 10 o Sistema Decimal: {0,1,…,9}
- Base 2 o Sistema Binario: {0,1}
- Base 8 o Sistema Octal: {0,1…,7}
- Base 16 o Sistema Hexadecimal: {0,…, 9, A, B, C, D, E, F}

Dependiendo de la posición que ocupe un dígito dentro de la cifra, representará las unidades, decenas, centenas, millares, etc. Por ello se dice que el sistema decimal es un sistema ponderado o sistema de numeración posicional.

Por ejemplo, el número 5545 se puede expresar como la suma de las potencias de la base 10.

$$(5*10^3) + (5*10^2) + (4*10^1) + (5*10^0)$$

3.1.- Teorema fundamental de la numeración.

Este teorema relaciona una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal; es decir, el valor decimal de una cantidad expresada en otro sistema de numeración que utiliza otra base. Viene dado por la fórmula:

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

Dónde:

- i = posición respecto a la como. Para los dígitos de la derecha la i es negativa, empezando en -1; para los de la izquierda es positiva, empezando en 0.
- d = número de dígitos a la derecha de la coma.
- n = número de dígitos a la izquierda de la coma -1.
- dígito = cada uno de los que componen el número.
- base = base del sistema de numeración.

El número en decimal será el sumatorio de multiplicar cada digito por la base elevada a su posición. *i* indica la posición del dígito respecto a la coma; si el número tiene decimales, *i* se iniciará en un valor negativo.

El teorema aplicado a la inversa, servirá para obtener la representación de una cantidad decimal en cualquier otra base por medio de divisiones sucesivas por dicha base.

3.2.- El sistema binario.

El sistema de numeración binario utiliza solo dos dígitos (0 y 1) para representar cantidades, por lo que su base es 2. Cada dígito de un número representado por este sistema se denomina **bit** (**bi**nary digi**t**).

Los bits tienen distinto valor dependiendo de la posición que ocupan; por eso este sistema también es posicional. Estos valores vienen determinados por una potencia de base 2 a la que llamaremos peso. Así, por ejemplo, el número binario 1011,01 expresado en decimal quedaría:

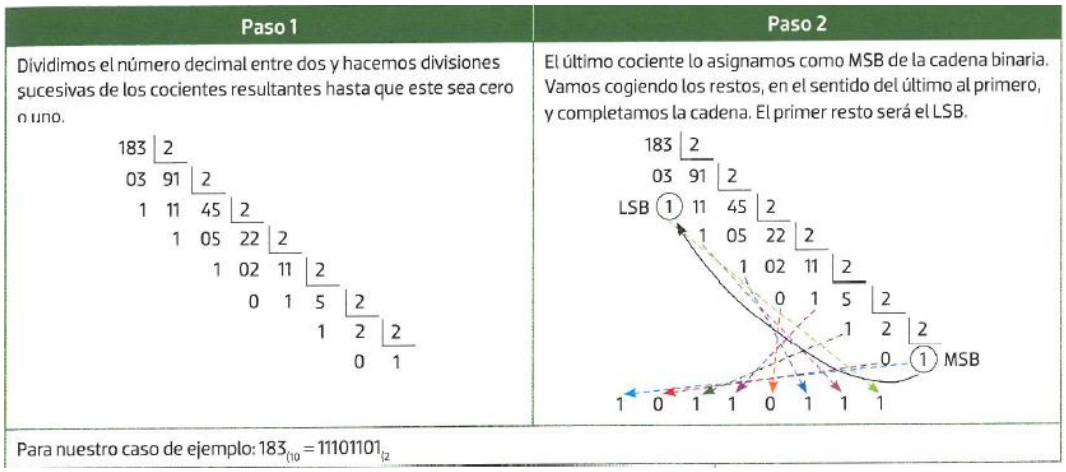
$$(1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) + (0 \cdot 2^{-1}) + (1 \cdot 2^{-2}) \rightarrow 11{,}25$$

Podemos observar la tabla, donde se muestran los pesos en potencia de 2 asociados según a posición del dígito. Para convertir a decimal, basta con colocar los dígitos en las columnas correspondientes y sumar los pesos donde hay un 1, hasta obtener la cantidad.:

Pesos asociados											**Nºdecimal**
2^6	2^5	2^4	2^3	2^2	2^1	2^0	,	2^{-1}	2^{-2}	2^{-3}	
64	**32**	**16**	**8**	**4**	**2**	**1**	**-**	**0,5**	**0,25**	**0,125**	
		1	1	0	0	1	,	0	1	0	33,25
1	0	0	0	0	1	1	,	1	1		67,75
		1	0	1	0	1	,	1	0	1	21,625

A. Conversión de un número decimal a binario.

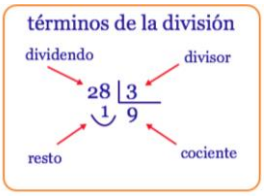
Para representar un número en sistema binario podemos utilizar los dígitos 0 y 1. La forma más simple de convertir a binario es dividir sucesivamente el número decimal, y los cocientes que se van obteniendo por 2 hasta que el cociente sea menor de 2. La unión del último cociente y todos los restos obtenidos, escritos en orden inverso, será el número expresado en binario.



Siglas:

- LSB: **bit menos significativo** o **L**east **S**ignificant **B**it es el bit que, de acuerdo a su posición, tiene el menor valor.
- MSB: **bit más significativo** o **M**ost **S**ignificant **B**it es el bit que, de acuerdo a su posición, tiene el mayor valor.

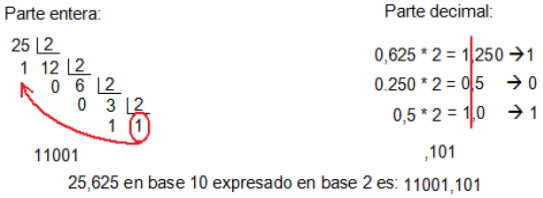
Recordatorio de terminología:



B. Conversión de una fracción decimal a binario.

La forma más sencilla consiste en multiplicar sucesivamente la parte fraccionaria por 2 *(por ser la base del sistema de numeración binario)* hasta que dé 0 como resultado. La parte entera de cada multiplicación formará los bits del número binario.

Puede ocurrir que la parte fraccionaria no salga 0, entonces dependerá con que error se quiera obtener el resultado, si el error queremos que sea menor que 2^{7}, sacaremos siete dígitos decimales.



C. Conversión de una fracción binaria a decimal.

Para esta conversión se utiliza el TFN. El resultado es la suma de los productos de los resultados de multiplicar cada dígito por la base elevado a la posición que ocupa pero en negativo.

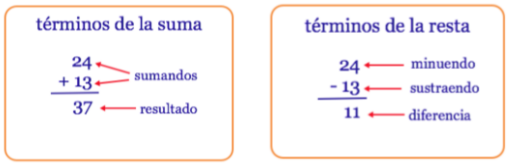
D. Operaciones aritméticas: Suma y resta en binario.

La suma binaria es parecida a la decimal, con la diferencia de que se manejan solo dos dígitos, el 0 y el 1. Si el resultado de la suma excede de 1, se agrega un acarreo a la suma parcial al siguiente término.

De igual forma la resta binaria es similar. Si el sustraendo es mayor que el minuendo (0-1), se agrega un acarreo a la resta parcial al siguiente sustraendo.

Suma binaria
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 y acarreo 1, que se suma al siguiente sumando
Resta binaria
0 - 0 = 0
0 - 1 = 1 y acarreo 1, que se suma al siguiente sustraendo
1 - 0 = 1
1 - 1 = 0

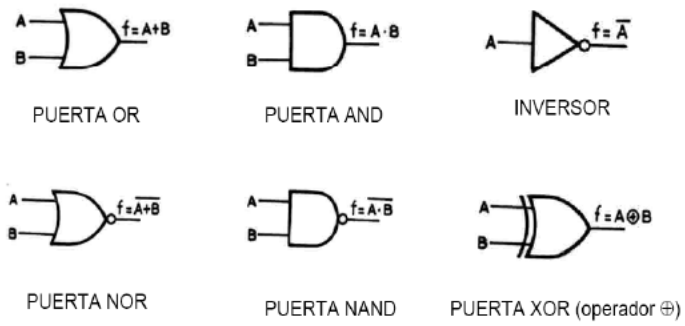
Recordatorio de terminología:



E. Operaciones lógicas: AND / OR / NOT.

Una conjunción lógica, una multiplicación binaria, en electrónica se implementa con un circuito llamado puerta lógica tipo AND. La suma binaria se implementa con una puerta lógica tipo OR. En la siguiente figura se presentan los tipos de puertas lógicas más comunes. Hay muchas otras, pero éstas son las fundamentales, que sirven para ilustrar los fundamentos de la electrónica digital.

Las puertas lógicas representan la base sobre la que se asientan los circuitos digitales, que van incrementando su complejidad hasta llegar a los microprocesadores.



La tabla de verdad indica el resultado de la operación. Su utilidad es la siguiente: para tratar la información, se usan enunciados lógicos. Un enunciado lógico es una proposición que puede tomar dos valores, verdadero o falso, en lógica binaria se usa por convenio 1 para verdadero y 0 para falso.

Sea la siguiente afirmación: Si hace frío y hay nubes, lloverá. ¿Lloverá? Sólo si se verifican las dos condiciones a la vez, es decir si el enunciado “hace frío” toma el valor 1 y a la vez el enunciado “hay nubes” toma también el valor 1. En cualquier otro caso no lloverá. Este enunciado puede tratarse con una operación AND en la que la variable X sea el enunciado “hace frío” y la variable Y “hay nubes”. Si se enuncia la proposición: Si hace frío o hay nubes, lloverá. La lógica cambia. Bastaría con que se verificase sólo una de las condiciones, que haga frío o nubes para que llueva en caso contrario no lloverá. Se representaría con una operación tipo OR.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	NOT X
0	1
1	0

3.3.- El sistema octal.

Los primeros sistemas informáticos utilizaban solo el sistema binario para interpretar y transformar los datos, con lo que las labores de programación eran bastante tediosas; se ocurrió el uso de sistemas intermedios que permitan una fácil traducción hacia y desde el sistema binario. Estos sistemas son el octal y el hexadecimal.

El sistema octal tiene base de numeración 8, es decir, utiliza 8 símbolos para representar las cantidades. Estos símbolos son 0, 1, 2, 3, 4, 5, 6, 7.

Para convertir de decimal a octal y viceversa se procede como en el sistema binario.

- Conversión de decimal a octal:
  - por medio de divisiones sucesivas
  - usando el paso intermedio de pasar a binario y luego a octal (pto. 3.5.)
- Conversión de un número octal a decimal:
  - empleando el TFN, donde la base ahora es 8.

3.4.- El sistema hexadecimal.

El sistema hexadecimal tiene como base de numeración 16, es decir, utilizan dieciséis símbolos para representar las cantidades. Estos símbolos son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

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

Para convertir de decimal a hexadecimal y viceversa se procede como en casos anteriores.

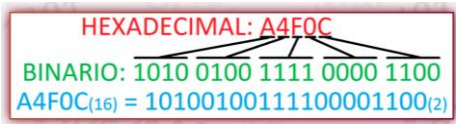
- Conversión de decimal a hexadecimal:
  - por medio de divisiones sucesivas
  - usando el paso intermedio de pasar a binario y luego a hexadecimal (pto. 3.5.)
- Conversión de un número hexadecimal a decimal:
  - empleando el TFN, donde la base ahora es 16.

3.5.- Conversiones entre sistemas.

De la misma forma que convertimos del sistema decimal al binario, octal y hexadecimal, y viceversa. También podemos convertir del binario al octal y hexadecimal y del hexadecimal al octal, etc.

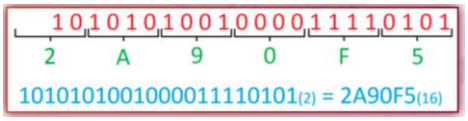
A. Conversión hexadecimal-binario.

Se sustituye cada dígito hexadecimal (0, 1, 2, 3…, D, E, F) por su representación binaria utilizando 4 dígitos; así el 0 se representará por 0000, el 1 por 0001, etc. se utilizan cuatro dígitos porque el valor más alto de este código, el 15, que se representa con la F, necesita cuatro dígitos: 1111.



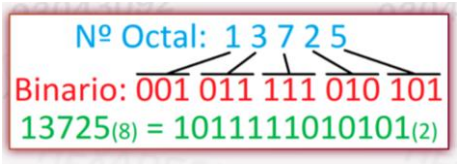
B. Conversión binario-hexadecimal.

Se agrupan los dígitos binarios de cuatro en cuatro a partir del punto decimal hacia la izquierda y hacia la derecha, y se sustituye cada grupo de cuatro por su valor correspondiente en hexadecimal.



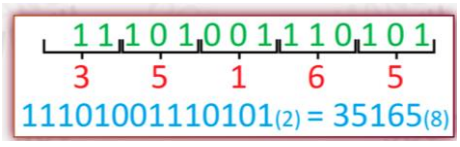
C. Conversión octal-binario.

Se procede como en la conversión hexadecimal-binario, se sustituye cada dígito octal por su representación binaria utilizando tres dígitos binarios.



D. Conversión binario-octal.

Se agrupan los dígitos de tres en tres a partir del punto decimal hacia la izquierda y hacia la derecha, sustituyendo cada grupo de tres por su equivalente en octal.



E. Conversión hexadecimal-octal y octal-hexadecimal.

Estas conversiones se realizan mediante un paso intermedio. Se pasan primero a base 2 y de ahí se pasará al sistema que se desee como ya hemos visto.

Tabla equivalencias entre distintos sistemas de numeración:

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
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Tabla resumen métodos de conversión entre distintos sistemas de numeración:

	Binario	**octal**	**decimal**	**hexadecimal**
binario	_____	Agrupar de a 3 bits	Formula poilinomica(pot. de 2)	Agrupar de a 4 bits
octal	Escribir c/digito en binario (3 bits)	_____	Formula polinómicas (pot. de 8)	Pasar por binario
decimal	Entera % 2 Fraccionaria x 2	Entera % 8 Fraccionaria x 8	_____	Entera % 16 Fraccionaria x 16
hexadecimal	Escribir c/digito en binario (4 bits)	Pasar por binario	Formula polinómicas (pot. de 16)	_____

4.- REPRESENTACIÓN INTERNA DE LA INFORMACIÓN.

El **bit** es la unidad mínima de información; con él podemos representar dos valores cualesquiera, como verdadero, falso, on, off, blanco o negro, etc. basta con asignar uno de estos valores al estado “apagado” (0) y otro al estado “encendido” (1).

Cuando se almacena información no se trabaja a nivel de bit, sino que se trabaja a nivel de carácter (letra, número o signo de puntuación), que ocupa lo que se denomina un **byte**, que a su vez está compuesto por 8 bits. El ordenador trabaja con agrupaciones de bits, ya que es más fácil de trabajar con ellos y trabajan con múltiplos de 2, que es la base del sistema binario. Los tamaños más comunes son:

- **Octeto, carácter o byte**: son 8 bits, tamaño típico de la información, con él se puede codificar el alfabeto completo (ASCII estándar).
- **Palabra**: tamaño de la información manejada en paralelo por los componentes del sistema, como la memoria, los registros o los buses. Son comunes las palabras de 8, 16, 32, 64, 128, 256 y 512 bits (1 byte, 2, 4, 8, 16, 32 o 64 bytes). A mayor tamaño de palabra, mayor precisión y potencia de cálculo del ordenador. Está dependerá de la arquitectura del ordenador. Este término está en desuso por su ambigüedad.

Cuando decimos que un archivo ocupa 5000 bytes, queremos decir que contiene el equivalente a 5000 letras o caracteres.

Lo normal es utilizar múltiplos del byte. En informática se utilizan las potencias de 2 (2^3, 2^{10}, 2^{20}, 2^{30},…), que son sus múltiplos

En Informática se han utilizado tradicionalmente los prefijos kilo (K), mega (M), tera (T) con un significado distinto al habitual en otros campos técnicos. Por ejemplo, kilo significa 1000 unidades pero kilo en Informática suele referirse a 1024 (210). Para resolver esta confusión, la Comisión Electrotécnica Internacional definió en 1998 las unidades kibi (Ki), mebi (Mi), gibi (Gi), tebi (Ti), pebi (Pi) y exbi (Ei), que se definen como potencias de 2.

Sistema Internacional (Decimal)			Sistema ISO/IEC 80000-13 (Binario)		
Múltiplo	Abreviatura	Equivalencia en bytes	Múltiplo	Abreviatura	Equivalencia en bytes
Kilobyte	KB	10^3 Bytes = 1000 Bytes	Kibibyte	KiB	2^{10} Bytes = 1024 Bytes
Megabyte	MB	10^6 Bytes = 1000000 Bytes	Mebibyte	MiB	2^{20} Bytes = 1 048 576 Bytes
Gigabyte	GB	10^9 Bytes = 1000000000 Bytes	Gibibyte	GiB	2^{30} Bytes = 1073741824Bytes
Terabyte	TB	10^{12} Bytes = 1000 Gigabytes	Tebibyte	TiB	2^{40} Bytes = 1024 Gibibytes
Petabyte	PB	10^{15} Bytes = 1000 Terabytes	Pebibyte	PiB	2^{50} Bytes = 1024 Tebibytes
Exabyte	EB	10^{18} Bytes = 1000 Petabytes	Exbibyte	EiB	2^{60} Bytes = 1024 Pebibytes
Zettabyte	ZB	10^{21} Bytes = 1000 Exabytes	Zebibyte	ZiB	2^{70} Bytes = 1024 Exbibytes
Yottabyte	YB	10^{24} Bytes = 1000 Zettabytes	Yobibyte	YiB	2^{80} Bytes = 1024 Zebibytes

1 kilobyte = 1000 B = 10^3 bytes. -> Usado para expresar Velocidad
1 kibibyte = 1024 B = 2^{10} bytes. -> Usado para expresar Capacidad

4.1.- Codificación.

La información se compone de datos. Para procesar los datos, se necesita correspondencia entre los símbolos y las señales que los representan. Ejecutar esta relación se denomina codificación. La codificación es una relación biunívoca entre el alfabeto fuente y el destino. La función inversa, a partir de las señales obtener la información es decodificar.

La información en general, se representa según su naturaleza. Esto da lugar a distintos tipos de codificación o formato. Según su naturaleza se puede identificar distintos tipos de información. Por ejemplo:

- Texto: Es un conjunto de símbolos que definen la expresión escrita humana. Se representa con símbolos o caracteres alfanuméricos. La agrupación en bloque de estos caracteres se trata como un fichero de texto. Un código típico de representación alfanumérica es el código ASCII, estándar estadounidense.
- Imagen: Es información que representa el objeto de la visión humana. Se representan como datos, que variarán en función de sus propiedades de calidad, resolución, etc. Ejemplos de codificación serían los formatos jpg o gif.
- Audio: Es información que sirve para representar sonidos. Formatos son cda, wma, mp3, etc.
- Video: Información que representa imágenes en movimiento. La impresión de movimiento para el ojo humano equivale a procesar un mínimo de 25 imágenes en un segundo. Formatos son avi o dvd.

En general se puede decir que la información se trata como un conjunto de datos. Los datos pueden clasificarse en numéricos, alfabéticos y alfanuméricos. En las TIC se usan distintos códigos para codificarlos, según su tipo.

Los códigos de E/S permiten traducir la información o los datos que nosotros podemos entender a una representación que la máquina puede interpretar y procesar. Los datos llegan y salen del ordenador a través de los periféricos de entrada y salida, respectivamente. Cada fabricante de componentes de E/S podría asignar una combinación diferente al mismo símbolo de origen; sin embargo, esto no sería nada positivo en un mercado abierto como el informático. Por eso se tiende a la estandarización de códigos, que ha llevado a la universalización de unos pocos códigos de E/S, como BCD, EBCDIC, ASCII y Unicode. La mayoría de estos códigos representan cada carácter por medio de un byte. Sin duda el más importante de todos es el ASCII.

4.2.- Representación de datos alfabéticos y alfanuméricos. Caracteres.

Los datos alfabéticos representan los caracteres del alfabeto. Se engloban en la representación alfanumérica que representa además números y caracteres como signos de puntuación, en

general para ser tratados como texto. En general, los códigos usan 8 bits para realizar esta representación.

Existen tablas distintas de caracteres según el idioma utilizado. Si se precisa usar varios idiomas pueden usarse códigos de dos bytes como UNICODE que permite representar 65536 símbolos distintos.

A. ASCII.

El código Estadounidense Estándar para el intercambio de información o **ASCII** (*American Standard Code for Information Interchange*), es la recomendación X3.4-1977 del Instituto Estadounidense de Normas Nacionales (ANSI). Utiliza grupos de 7 bits por carácter, permitiendo 2^7 →128 caracteres diferentes, lo que es suficiente para el alfabeto en letras mayúsculas y minúsculas y los símbolos más corrientes, además de algunas combinaciones reservadas para su uso interno. El código **ASCII extendido** emplea 8 bits por carácter, lo que añade 128 caracteres posibles: 2^8 →256. Este juego de códigos más amplio permite que se agreguen los símbolos de lenguajes extranjeros y varios símbolos gráficos.

ASCII también se conoce como la ISO 8856-1 y se utiliza en los sistemas operativos MS-DOS, Windows y UNIX.

Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sínc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

Caracteres ASCII imprimibles

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

ASCII extendido

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	█	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

B. Unicode.

El Unicode Standard es una norma de codificación universal de caracteres que se emplean en los ordenadores Windows NT y en los navegadores de Internet Explorer y Netscape a

partir de la versión 4. Su uso se está extendiendo. Utiliza 16 bits, lo que permite codificar todos los caracteres de cualquier lenguaje, hasta 65536.

La versión 3 de Unicode tiene 9194 caracteres de los utilizados en los lenguajes más importantes del mundo. El objetivo de Unicode es representar cada elemento usado en la escritura de cualquier idioma del planeta. Los idiomas actuales más importantes del mundo pueden escribirse con Unicode, incluyendo su puntuación, símbolos especiales, símbolos matemáticos y técnicos, formas geométricas, caracteres gráficos y modelos braille.

Unicode proporciona un número único para cada carácter, sin importar la plataforma, sin importar el programa, sin importar el idioma. Líderes de la industria tales como Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, un, Sybase, Unisys y muchos otros han adoptado la norma Unicode. Unicode es un requisito para los estándares modernos tales como XML, Java, ECMA Script (JavaScript), LDAP, CORBA 3.0, WML, Etc., y es la manera oficial de aplicar la norma ISO/EIC 10646. Es compatible con numerosos sistemas operativos, con todos los exploradores actuales y con muchos otros productos. La aparición de la norma Unicode y la disponibilidad de herramientas que la respaldan se encuentran entre las más recientes e importantes tendencias en tecnología de software.

La incorporación de Unicode en sitios web y en aplicaciones de cliente-servidor o de múltiples niveles permite disminuir ostensiblemente los costos del uso de juegos de caracteres heredados. Unicode permite que un producto de software o sitio web específico se oriente a múltiples plataformas, idiomas y países, sin necesidad de rediseñarlo. Además, permite que los datos se trasladen a través de gran cantidad de sistemas distintos sin sufrir daños.

Básicamente, los ordenadores solo trabajan con números. Almacenando letras y otros caracteres mediante la asignación de un número a cada uno. Antes de que se inventara el Unicode, existían cientos de sistemas de codificación distintos para asignar estos números. Ninguna codificación específica podía contener caracteres suficientes; por ejemplo, la Unión Europea, por sí sola necesita varios sistemas de codificación distintos para cubrir todos sus idiomas. Incluso para un solo idioma como el inglés no había un único sistema de codificación que se adecuara a todas las letras, signos de puntuación y símbolos técnicos comunes.

C. EBCDIC.

El EBCDIC, Extended BDC Interchange Code (o código BCD extendido de caracteres decimales codificados en binario para el intercambio de información), es un sistema de codificación que tiene como objetivo la representación de caracteres alfanuméricos. Es el utilizado por la empresa IBM para sus ordenadores de la serie IBM PC miniordenadores y mainframes).

En este sistema de codificación, cada carácter tiene 8 bits. Al tener 8, podemos representar 2^8 → 256 caracteres. Será posible almacenar letras mayúsculas, minúsculas, caracteres especiales, caracteres de control para dispositivos de E/S y para comunicaciones.

Código EBCDIC

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	00	NUL	32	20	DS	64	40	SP	96	60	-
1	01	SOH	33	21	SOS	65	41	RSP	97	61	/
2	02	STX	34	22	FS	66	42	â	98	62	Â
3	03	ETX	35	23		67	43	ä	99	63	Ä
4	04	PF	36	24	BYP	68	44	à	100	64	À
5	05	HT	37	25	LF	69	45	á	101	65	Á
6	06	LC	38	26	ETB	70	46	ã	102	66	Ã
7	07	DEL	39	27	ESC	71	47	å	103	67	Å
8	08	GE	40	28		72	48	ç	104	68	Ç
9	09	RLF	41	29		73	49	ñ	105	69	Ñ
10	0A	SMM	42	2A	SM	74	4A	[	106	6A	¦
11	0B	VT	43	2B	CU2	75	4B	.	107	6B	,
12	0C	FF	44	2C		76	4C	<	108	6C	%
13	0D	CR	45	2D	ENQ	77	4D	(	109	6D	_
14	0E	SO	46	2E	ACK	78	4E	+	110	6E	>
15	0F	SI	47	2F	BEL	79	4F	!	111	6F	?
16	10	DLE	48	30		80	50	&	112	70	ø
17	11	DC1	49	31		81	51	é	113	71	É
18	12	DC2	50	32	SYN	82	52	ê	114	72	Ê
19	13	TM	51	33		83	53	e	115	73	È
20	14	RES	52	34	PN	84	54	è	116	74	È
21	15	NL	53	35	RS	85	55	í	117	75	Í
22	16	BS	54	36	UC	86	56	î	118	76	Í
23	17	IL	55	37	EOT	87	57	ı	119	77	Ì
24	18	CAN	56	38		88	58	ì	120	78	Ì
25	19	EM	57	39		89	59	ß	121	79	`
26	1A	CC	58	3A		90	5A	]	122	7A	:
27	1B	CU1	59	3B	CU3	91	5B	$	123	7B	#
28	1C	IFS	60	3C	DC4	92	5C	*	124	7C	@
29	1D	IGS	61	3D	NAK	93	5D	)	125	7D	'
30	1E	IRS	62	3E		94	5E	;	126	7E	=
31	1F	IUS	63	3F	SUB	95	5F	°	127	7F	"

COM II- I. Zamora Uni II - Conf3: Cod. fte y Fmteo 18

4.3.- Representación de datos numéricos. Números.

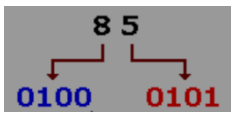
Representar números tiene el problema de ser un conjunto infinito y, por tanto, con una memoria limitada no se puede representar cualquier número, hay que acotar. Utilizando n bits se pueden representar 2^n números distintos. El rango de representación es el intervalo entre el mayor y el menor número representable.

A. BCD

BCD, Binary Coded Decimal (Decimal Codificado en Binario), en realidad no es un código de E/S, sino una forma de codificar los símbolos numéricos del 0 al 9 que se emplean en varios códigos de E/S, entre los que figuran EBCDIC y ASCII.

BCD divide cada octeto en dos mitades o cuartetos, cada uno de los cuales almacena en binario una cifra. Con este código es muy fácil convertir del binario al sistema decimal.

Con el BCD solo se utilizan 10 de las 16 posibles combinaciones que se pueden formar con números de 4 bits, por lo que el sistema pierde capacidad de representación, aunque se facilita la compresión de los números. Esto es porque el BCD solo se usa para representar **cifras**, no números en su totalidad. Esto quiere decir que **para números de más de una cifra hacen falta dos números BCD**.



AMPLIACIÓN: CODIFICACIÓN DE DATOS NUMÉRICOS.

Para representar números debe tenerse en cuenta el tipo de número a representar (enteros, real…), el rango de representación, la cantidad de números del rango o la memoria necesaria, entre otros.

Enteros

Para representar un número entero, se distingue si es positivo o negativo. Si sólo se representan enteros positivos, con n bits se pueden representar de 0 a 2n -1. Para codificar enteros negativos, hay que tener en cuenta que el intervalo de representación de enteros positivos debe ser igual al de los negativos y que el signo y el 0 sean fácilmente detectables y el código resultante permita realizar las operaciones habituales con una implementación hardware sencilla.

Para ello, se utilizan tres técnicas: signo y magnitud, complemento A1 (CA1) y complemento A2 (CA2).

- La representación de signo y magnitud consiste en reservar un bit para el signo y el resto para la magnitud. Típicamente se usa el 1 para indicar que el número es negativo. Presenta el problema de representar el 0 de dos formas y la suma no es tan sencilla. Esta representación no suele implementarse.
- La representación con CA1 para un número positivo usa su CA1 y el complementario del CA1 si el número es negativo. También representa dualmente el 0.
- La representación con CA2 con n bits para un número positivo utiliza el CA1 del número y para un número negativo se usa el CA2, que es el CA1+1. Es más costoso de implementar, pero elimina la representación dual del 0. Es la representación más extendida para los enteros negativos.

Existen también otras técnicas de representación de enteros, no posicionales, es decir, se crea un nuevo código para representar cada cifra. Los más representativos son:

- BCD Natural. La codificación del valor del dígito se expresa con 4 bits. Por tanto, desaprovecha 6 símbolos.
- Exceso a 3: Código BCD natural que suma 3 a cada dígito.
- Aiken. Código BCD en que los pesos dentro de cada grupo de 4 bits son 2, 4, 2, 1 en vez de 8, 4, 2, 1. Es autocomplementario.
- Gray. Consiste en diferenciar símbolos consecutivos en un único bit de diferencia.
- JOHNSON 5 bits. Consiste en incrementar el número de unos desde la derecha y disminuirlos paulatinamente por la izquierda según avanza la representación. No es ponderado.

Decimal	BINARIO	BCD natural	BCD exceso 3	BCD Aiken	GRAY	JOHNSON 5 bits
0	0000	0000	0011	0000	0000	00000
1	0001	0001	0100	0001	0001	00001
2	0010	0010	0101	0010	0011	00011
3	0011	0011	0110	0011	0010	00111
4	0100	0100	0111	0100	0110	01111
5	0101	0101	1000	1011	0111	11111
6	0110	0110	1001	1100	0101	11110
7	0111	0111	1010	1101	0100	11100
8	1000	1000	1011	1110	1100	11000
9	1001	1001	1100	1111	1101	10000

Reales

La representación de reales puede seguir diversas convenciones. Las más comunes son punto fijo, donde se usan una parte de los bits como parte entera y otra parte como decimal; punto flotante, que representa el número en forma exponencial, con mantisa, base y exponente. El exponente usa n bits y la mantisa m; IEEE 754, que es un estándar para representación en punto flotante de con 32 y 64 bits.

La representación de 32 bits (simple precisión) usa el primer bit para signo (S), los 8 siguientes representan el exponente (E) en exceso 127 (C=E+127) y los 23 restantes para la mantisa (M), normalizada como 0,xxx. La representación de 64 bits (doble precisión) es similar: usa el primer bit de signo (S), 11 para el exponente (E) y 52 de mantisa (M).

4.4.- Representación de colores

Para definir los colores en las páginas web se utilizan tres pares de números hexadecimales que representan la combinación de los tres colores primarios: rojo, verde y azul (paleta de colores RGB Red-Green-Blue).

La sintaxis para codificar un color en HTML es la siguiente: color = «#RRGGBB».

Donde RR, GG y BB representan un número hexadecimal para cada color entre 00 y FF (0 a 255 en decimal) en el orden rojo, verde y azul.

Así por ejemplo el color rojo se representa por «#FF0000», es decir «255» rojo, «0» verde, y «0» azul. El verde sería «#00FF00», y el azul «#0000FF».

El blanco se representa por «#FFFFFF» y el negro por «#000000».

ACTIVIDADES:

1.- Expresa la cantidad según el teorema fundamental de la numeración.

a) 234,765
b) 347,21
c) 800,102

2.- Representa en el sistema decimal los siguientes números de distintas bases.

a) $123,45_{(6}$
b) $4300,012_{(5}$
c) $1101,0011_{(2}$

3.- Convierte a binario:

a) $178,2_{(8}$
b) $29,3125_{(10}$
c) $A,B2_{(16}$

4.- Convierte a hexadecimal:

a) $110010,1101_{(2}$
b) $56,375_{(10}$
c) $156,22_{(8}$

5.- Convierte a octal:

a) $1101110,01001_{(2}$
b) $29,3125_{(10}$
c) $9A,53F2_{(16}$

6.- Realiza las siguientes sumas en binario:

a) 11111111 +1
b) 1011,101 + 101,110
c) 11001,11 + 10,1

7.- Consulta las tablas de los códigos ASCII y EBCDIC, representa el nombre del instituto, cada carácter es un byte. Ponlo en hexadecimal y en binario. Haz lo mismo con tu nombre.

8.- Expresa en bytes las siguientes cantidades:

a) 25 YB
b) 15 ZB
c) 20 PB