

Sistemas Informáticos

Tema 1. Representación de la Información

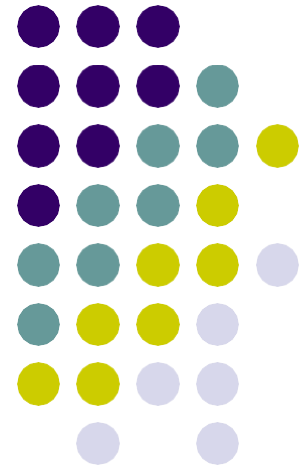




Tabla de contenidos

1. Introducción. El Sistema Informático
 1. Elementos del Sistema Informático
2. El Ordenador
 1. La Información en el Ordenador
3. Sistemas de numeración
 1. El sistema binario
 2. Sistemas de numeración intermedios
4. Codificación alfanumérica.

1. Introducción



Sistema Informático

Cualquier empresa necesita procesar y almacenar **información** para llevar a cabo su actividad.

El **Sistema Informático** es el conjunto de elementos que utiliza la empresa para con este objetivo.

Los elementos del Sistema Informático son **hardware, software, los datos y el personal humano**

Hardware: es la parte física, los dispositivos electrónicos (se puede tocar)

Ordenadores, periféricos y sistema de comunicaciones

Software: es la parte lógica (no se puede tocar)

Sistema Operativo (SO) y otras aplicaciones

Personal Humano: personas que crean, mantienen y utilizan el **Sistema**.

Técnicos, Programadores, Usuarios

Datos: información de la empresa para el desarrollo de su actividad.



1.1. Elementos de un sistema informático

● Hardware

- Materia física del Sistema Informático
- Dispositivos **electrónicos**
- se encuentran distribuidos físicamente en la CPU o torre, los periféricos y sistema de comunicaciones.
- No pueden trabajar por si mismos sino con los demás.
- Proporcionan la capacidad de proceso y la potencia de cálculo del sistema informático, así como las interacción con el mundo exterior.
 - Ej. Monitor, tarjeta gráfica, tarjeta de red ...



1.1. Elementos de un sistema informático

● Software

- programas que se ejecutan en el ordenador
- un programa realiza una función o tarea.
- Los programas tampoco son capaces de trabajar por si mismos sino que se apoya en un software muy especial que es el SO





1.1. Elementos de un sistema informático

● Personal Humano

- personas que participan en la dirección, diseño, desarrollo, implantación y explotación de un sistema informático.
- personal informático, según las funciones que realicen:
 - Director, jefe de proyecto, Técnico de Sistemas, Administrador Base de Datos, Analista funcional u orgánico, Programador, Jefe de explotación, Operador, Grabador y Administrador de sistemas.



2. El Ordenador

- Un ordenador es una máquina electrónica capaz de coger datos de entrada, procesarlos y producir datos de salida.
- La función principal de un ordenador es automatizar tareas
 - Para ello **Ejecuta Programas**: Un ordenador no es inteligente, lo que nos aporta es velocidad.
- Un programa está formado por instrucciones. Una instrucción es una orden que se realiza sobre unos datos.
 - Instrucción: sumar 3 y 5
- **TODAS** las tareas que realizamos con un ordenador , para el ordenador sólo son un conjunto de instrucciones que ejecuta.
 - **Ejemplos**: podemos utilizar un ordenador jugar. Cuando jugamos y movemos un personaje con el ratón o el teclado lo que hace el ordenador es ejecutar la orden mover sobre ese personaje y visualizarlo en el monitor en la posición de destino.

2.1 La Información en el Ordenador



- Tipos de Información:
 - Números
 - Letras
 - Música
 - Imágenes
- Los componentes del ordenador no entienden esta información, solo entienden de corriente eléctrica u otras magnitudes físicas.
 - sólo entiende la presencia o ausencia de corriente eléctrica (**el código binario**)
- Necesitamos sistemas de codificación que conviertan letras y números que nosotros utilizamos en algo entendible por el ordenador.
- El SO con ayuda de los componentes hardware transforman la información en impulsos eléctricos (0,1) o impulsos eléctricos (0,1) en información.



2.1 La Información en el Ordenador

- El **bit** es la unidad mínima de información representado por 0 ó 1.
- Medidas múltiplos de bits:
 - **Nibble o cuarteto** conjunto de 4 bits
 - **Byte u octeto** conjunto de 8 bits.
 - **Kilobyte (KB)** conjunto de 1024 bytes
 - **Megabyte (MB)** conjunto de 1024 KB
 - **Gigabyte (GB)** conjunto de 1024 MB
 - **Terabyte (TB)** conjunto de 1024 GB
 - **Petabyte (PB)** conjunto de 1024 TB
 - **Exabyte (EB)** conjunto de 1024 PB

Ejemplos
Archivo de texto plano unos 20 KB
Archivo .mp3 unos 3MB
Película 1GB
800 películas 1TB

$2^{10} = 1024$ por lo que $1\text{TB} = 2^{40}\text{bytes}$

- Recuerda la B representa bytes y la b quiere decir bit, por ejemplo:
 $30\text{MB} = 30$ megabytes y $30\text{Mbs} = 30$ megabits por segundo



Actividad

- a) ¿Cuántos KB son 2TB?
- b) ¿Cuántos B son 4MB?
- c) ¿Cuántos bits son 2B?
- d) ¿Cuántos MB son 1024KB?
- e) ¿Cuántos GB son 2048MB?
- f) ¿Cuántos TB son 256B?
- g) ¿Cuántos TB son 2048MB?



3. Sistemas de numeración

- Conjunto de símbolos y reglas que se utilizan para representar cantidades o datos numéricos.
 - **Base:** número de símbolos que lo componen.
 - Base 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
 - Base 2: 0, 1
 - El valor de cada símbolo depende de la posición que ocupa en el número:

Teorema fundamental de la numeración

$$\text{NUM} = \sum X_i * B^i$$

donde **X** valor absoluto del dígito
i posición respecto al punto decimal
B base

- Ejemplo: el número $99_{(10)}$ es:
 - $9 * 10^1 + 9 * 10^0 = 9 * 10 + 9 * 1 = 90 + 9 = 99$
- Cada posición del dígito tiene un valor intrínseco que aumenta de derecha a izquierda según potencias sucesivas de la base de sistema empleado



3.1 El sistema binario

- Sistema de numeración en Base 2: Utiliza dos símbolos diferentes 0 y 1 , denominados bits.
- Es el sistema que maneja el ordenador internamente. Ya que se asocia cada uno de estos símbolos a la presencia o ausencia de tensión u otra magnitud.
- Con n bit podemos representar 2^n valores
 - Si $n=1$ podemos representar $2^1=2$ valores el 0 y el 1
 - Si $n=2$ podemos representar $2^2=4$ valores el 00-01-10-11
- Siguiendo el TFN, un número binario estará formado por un conjunto ordenado de bits, cuyo valor aumenta de derecha a izquierda según potencias de dos.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1



3.1.1 Transformaciones entre binario y decimal

- Para transformar un número **binario a decimal**, debemos aplicar el TFN:

$$1001_{(2)} = 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 0 + 0 + 1 = 9_{(10)}$$

- Es decir, sumamos las potencias de dos cuyos dígitos son 1:

$$\begin{array}{cccc|cccc} 1001.001)_2 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ \hline \text{pesos} \rightarrow & 8 & 4 & 2 & 1 & 1/2 & 1/4 & 1/8 \end{array} = 8 + 1 + 1/8 = 9.125)_{10}$$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1



3.1.2 Transformaciones entre decimal y binario

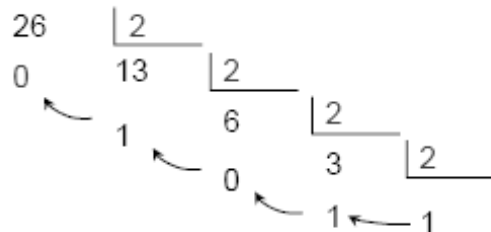
- Para transformar un número **decimal a binario** dividimos la parte entera entre dos, hasta que sea menor que la base, los restos y el último cociente en orden inverso son el número binario:

$$9:2 = 4 \text{ Resto } 1$$

$$4:2 = 2 \text{ Resto } 0$$

$$2:2 = 1 \text{ Resto } 0$$

$$9_{(10)} = 1001_{(2)}$$



$$26_{(10)} = 11010_{(2)}$$

0.1875	0.3750	0.7500	0.5000
$\times 2$	$\times 2$	$\times 2$	$\times 2$
0.3750	0.7500	1.5000	1.0000

$$26.1875_{(10)} = 11010.0011_{(2)}$$



Actividad

Pasa el siguiente número de binario a decimal

$$1010.10100_{(2)} =$$

Pasa el siguiente número de decimal a binario

$$20.125_{(10)} =$$



3.1.4 Operaciones lógicas en binario

- Una operación lógica asigna un valor (CIERTO=1 o FALSO=0) a la combinación de uno o más factores que también toman los valores (CIERTO=1 o FALSO=0) .
- Los resultados de una operación lógica, para cada uno de los valores posibles de las variables, se fijan en una tabla denominada **Tabla de Verdad**, como la del siguiente ejemplo.
- Por ejemplo, imagínate el sistema de control del toldo de una cafetería, que se gobierna mediante una operación lógica. Para que el motor que extiende el toldo se accione deberá tener en cuenta dos factores: ¿es de día? ¿está lloviendo? Si estos dos factores son ciertos, el motor debe ponerse en marcha y extender el toldo.

De día	Llueve	Toldo
Falso	Falso	Falso
Falso	Cierto	Falso
Cierto	Falso	Falso
Cierto	Cierto	Cierto

Falso=0
Cierto=1



3.1.4 Operaciones lógicas en binario

- Si el **detector de metales SI** nota que el cliente lleva objetos metálicos (1) y la puerta **NO** se abre (0); en cambio, si el cliente **NO** lleva objetos metálicos (0), la puerta **SI** se abre (1).

Función NOT

<i>a</i>	<i>s</i>
1	0
0	1

Falso=0

Cierto=1

Función OR

<i>a</i>	<i>b</i>	<i>s</i>
0	0	0
0	1	1
1	0	1
1	1	1

- Una aplicación práctica sencilla de la operación lógica OR, sería **el circuito de señalización instalado en un comercio**, en el que se puede entrar por dos puertas distintas, que avisaría al dependiente al entrar un cliente por cualquiera de las dos puertas del establecimiento. Si un cliente entra por la puerta a (1) O si un cliente entra por la puerta b (1), el timbre suena (1). Si no entra ningún cliente por ninguna de las puertas a (0) ni b (0). El timbre NO suena (0).



3.1.4 Operaciones lógicas en binario

- Una aplicación de la operación lógica AND, sería el sistema de control de los pasajeros en un aeropuerto. Cada pasajero debe pasar por dos controles: ¿Tiene tarjeta de embarque? ¿Tiene pasaporte en regla? Una empleada del aeropuerto comprueba que tiene un billete válido y le da una tarjeta de embarque; a continuación, un agente de policía verifica que su pasaporte está en regla y no está en la lista de personas reclamadas. Un pasajero sólo puede embarcar en el avión si tiene tarjeta de embarque (1) y su pasaporte está en regla (1) . En los demás casos no puede embarcar.

Falso=0
Cierto=1

Función AND

<i>a</i>	<i>b</i>	<i>s</i>
0	0	0
0	1	0
1	0	0
1	1	1



Ejercicios

En el aula virtual Realizamos
Ejercicios1 del Tema1



3.2 Sistemas de numeración intermedios

- Se utilizan por su proximidad de significado al sistema decimal y su facilidad de conversión a binario:

- **Octal**

- Utiliza ocho símbolos diferentes del 0 al 7 (Base 8)
- Tiene correspondencia directa con el sistema binario (cada símbolo en base 8 se puede representar con 3 bits).
- A veces lo utiliza el ordenador, por comodidad, para manejar datos.

- **Hexadecimal**

- Utiliza 16 símbolos, del 0 al 9 y de la A a la F. Estas letras representan los dígitos del 10 al 15 del sistema decimal.
- Tiene correspondencia directa con el sistema binario (cada símbolo en base 16 se puede representar con 4 bits).
- Los usa el ordenador para expresar direcciones de memoria.



3.2 Sistemas de numeración intermedios

- Transformación de binario a códigos intermedios y viceversa
 - **Octal:** hacemos grupos de tres:
 - $\underline{100}101\underline{100}_{(2)} = 454_{(8)}$
 - $551_{(8)} = 101\ 101\ 001_{(2)}$
 - **Hexadecimal:** hacemos grupos de cuatro:
 - $\underline{0001}0010\underline{1100}_{(2)} = 12C_{(16)}$
 - $32A_{(16)} = 0011\ 0010\ 1010_{(2)}$

Ejercicios

En el aula virtual Realizamos
Ejercicios2 del Tema1





4. Codificación alfanumérica

- El ordenador no solo procesa datos numéricos, también procesa datos alfanuméricos (caracteres), estos al ser datos con los que no se realizan operaciones pueden ser creados utilizando tablas establecidas por acuerdo.
- Los sistemas de codificación estándar más importantes son:
 - **ASCII** (*American Estándar Code for Information Interchange*)
 - **LATIN-1**
 - **ISO-8859-1**.
 - **UNICODE** (*Universal Code*)
 - **UTF-8** (*8-bit Unicode Transformation Format*)



4. Codificación alfanumérica. ASCII

- American Standard Code For Information Interchange
- 7 bits para representar cada carácter
- Los 32 primeros son de control, no imprimibles y actualmente obsoletos, como el código 10 que hacía a la impresora saltar a la siguiente línea



4. Codificación alfanumérica

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000 0	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
0001 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0010 2	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0011 3	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0100 4	EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
0101 5	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0110 6	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0111 7	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1000 8	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1001 9	HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121
1010 A	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1011 B	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1100 C	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	 124
1101 D	CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125
1110 E	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1111 F	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127

Tabla de códigos ASCII

HOLA

H = 1001000

O = 1001111

L = 1001100

A = 1000001

HOLA

100100010011111001100 1000001



Actividad

Escribe en hexadecimal la frase **Tengo 18 años** usando ASCII de 7 bits.



4. Codificación alfanumérica. Latin-1

- ASCII sólo sirve para caracteres del alfabeto inglés
 - No permite caracteres con acento ni ñ
- Latin-1 es ASCII extendido de 8 bits
 - Idéntico a ASCII hasta el carácter 127
 - Desde el 128 hasta el 255: vocales acentuada, ñ
 - ...
- Sirve para los caracteres de los alfabetos de Europa Occidental
 - Español, Francés, Portugués ...

Actividad

Busca la tabla de códigos ASCII extendida





Actividad

Pasar la siguiente cadena de bits (escrito en ASCII con 8 bits) a Texto, pasándolo primero a hexadecimal y luego a texto:

```
01000001 00100000 01110000 01100001 01110010 01110100 01101001
01110010 00100000 01100100 01100101 00100000 01101000 01101111
01111001 00101100 01110011 01101111 01111001 00100000 01110101
01101110 00100000 01100010 01110101 01100101 01101110 00100000
01110100 01110010 01100001 01100100 01110101 01100011 01110100
01101111 01110010 00100000 01100100 01100101 00100000 01000001
01010011 01000011 01001001 01001001 00100000 01100001 00100000
01100010 01101001 01101110 01100001 01110010 01101001 01101111
00101110
```



4. Codificación alfanumérica. ISO-8859-1

- Codificación Estándar de la IANA de 1992 de 8 bits
- Para la codificación de texto enviado por Internet
- Está basado en Latin-1 pero no es equivalente
 - Ejemplo: Para indicar que en una página html los caracteres están codificados en formato ISO-8859-1 debe aparecer
 - `<meta http-equiv=«content-type»
content=«text/html»;charset=iso-8859-1»>`

4. Codificación alfanumérica. UNICODE



- **Universal Code**
- Utiliza 16 bits para representar caracteres alfanuméricos
- **UTF-8 (8-bit Unicode Transformation Format)** es una parte del estándar Unicode compatible con ASCII
 - La IETF (Internet Engineering Task Force) define que UTF-8 debe estar soportada por todos los protocolos de Internet
 - Actualmente se tiende a tener todas las aplicaciones y bases de datos con la información codificada en UTF-8
 - Ejemplo: Para indicar que en una página html los caracteres están codificados en formato UTF-8 debe aparecer
 - `<meta http-equiv=«content-type» content=«text/html»;charset=utf-8»>`



Actividad

¿Qué tipo de codificación se usa en las siguientes webs?.

- <http://barrapunto.com>
- <http://meneame.net>
- <http://microsoft.com>
- <http://ubuntu.com>