Unidad de trabajo 2: *INFORMÁTICA BÁSICA (primera parte)*

1. Representación interna de datos.

El ordenador, debido a las características de su construcción, emplea únicamente el sistema binario, utilizando una serie de códigos que permiten su perfecto funcionamiento.

Tanto el sistema decimal como el binario están basados en los mismos principios. En ambos, la representación de un número se efectúa mediante cadenas de símbolos, que representan una determinada cantidad dependiendo del propio símbolo y de la posición que ocupa dentro de la cadena.

Por cuestiones técnicas, los circuitos electrónicos que constituyen un ordenador están capacitados para reconocer señales eléctricas de tipo digital; por tanto, es necesario que los métodos de codificación internos tengan su origen en el sistema binario, pudiéndose representar con ellos todo tipo de información y órdenes.

En los circuitos electrónicos, desde el punto de vista lógico, la presencia de tensión en un punto de un circuito suele representarse mediante un 1 y la ausencia de tensión se representa con un 0. En este caso se dice que está utilizando ***lógica positiva.***

Sin embargo, si se asocia el 0 a la presencia de tensión y el 1 a su ausencia, se dice que se utiliza **lógica negativa**.

1.2. Sistemas de numeración.

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

En un sistema de numeración podemos distinguir la base, que es el número de símbolos que utiliza y que se caracteriza por ser el coeficiente que determina cuál es el valor de cada símbolo, dependiendo de su posición. Por tanto, los sistemas de numeración actuales son sistemas posicionales.

Así podremos encontrar los siguientes sistemas que ya conocemos:

El sistema decimal, que utiliza como base el número 10. Esto quiere decir que asocia cada dígito a una potencia de 10 en función de su posición.

Se emplean 10 símbolos para la representación de cantidades:

0 1 2 3 4 5 6 7 8 9

El sistema binario, es un sistema con base 2, es decir, se compone de dos dígitos, el 0 y el 1. Cada dígito de un número representado en este sistema se denomina bit (contracción de binary digit).

Su utilidad es fácilmente comprensible, ya que los circuitos y componentes electrónicos sólo conocen dos estados, ***permitir o no el paso de corriente***. Un interruptor puede estar abierto o cerrado.

El sistema octal, tiene como base los 8 dígitos siguientes:

0 1 2 3 4 5 6 7

El sistema hexadecimal, tiene como base 16 dígitos compuestos por:

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

1.3. Aritmética binaria.

Como es natural podremos operar con los distintos sistemas de numeración, llevando a cabo tanto cambios de base de sistemas como sumas, restas, multiplicaciones y divisiones.

1.3.1. Cambio de base binaria a decimal.

El método empleado consiste en el desarrollo de la serie de potencias de la base. Por ejemplo:

$$\text{Nº} = (\text{dígito}) \times (\text{base})^{i}$$

i -> posición respecto a su dígito, de derecha a izquierda

El número 10100 se resolverá del siguiente modo -> $\mathbf{1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 16 + 4 = 20}$

1.3.2. Cambio de base decimal a binaria.

El método más utilizado es el método de la división/multiplicación sucesiva. Este sistema consiste en dividir la parte entera hasta llegar a un **cociente de 0 ó 1**.

El número 16 se resolverá del siguiente modo -> 16:2=8 Resto ***0***. 8:2=4 Resto ***0***. 4:2=2 Resto ***0***. 2:2=***1*** Resto ***0***. → Cociente = 1. PARAMOS de dividir. Por tanto el resultado es 10000.

Seleccionaremos **primero el cociente** resultante y después los restos, desde el último hasta el primero.

1.3.3. Suma binaria.

La suma es similar al decimal y cuando el resultado excede de los símbolos utilizados, se agrega el exceso a la suma parcial siguiente hacia la izquierda. Las tablas de sumar son:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 1 0 -> Se produce ***acarreo***

1.3.4. Diferencia binaria.

La resta también es similar al decimal, pero si el sustraendo excede al minuendo, se generará un acarreo que se añade al siguiente sustraendo. La tabla de la resta es la siguiente:

0 – 0 = 0
0 – 1 = 1 (negativo/acarreo)
1 – 0 = 1
1 – 1 = 0

1.3.5. Multiplicación binaria.

La multiplicación consiste en repetir el multiplicando tantas veces como indique el multiplicador, por tanto, la tabla de multiplicación será la siguiente:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

1.3.6. Conversión octal/hexadecimal a binario.

El procedimiento más rápido para convertir un número octal a binario consiste en sustituir cada dígito del sistema octal por su correspondiente en binario. Veamos en qué consiste:

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

Y de hexadecimal a binario:

Dígito Hexadecimal	Dígitos Binarios
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A (equivale a 10)	1010
B (equivale a 11)	1011
C (equivale a 12)	1100
D (equivale a 13)	1101
E (equivale a 14)	1110
F (equivale a 15)	1111

¿Pero cómo funciona la conversión en sí?

Si queremos pasar de sistema octal a binario, deberemos coger cada dígito octal y traducirlo a su número binario.

Por ejemplo, el número **$73_{8)}$** equivale al número binario **111 011**
(7) (3)

Para transformar hexadecimal a binario se llevará a cabo la misma operación.

Por ejemplo, el número **$7C_{16)}$** equivale al número binario **0111 1100**

Para hacer la conversión inversa deberemos seguir el mismo proceso, agrupando de tres en tres dígitos si es código octal y de cuatro en cuatro si es hexadecimal.

Ejercicios propuestos:

1) Transformar los siguientes números de **binario** a **decimal**.
   a. 101
   b. 101011
   c. 10100101
   d. 1111011

2) Convertir los siguientes números en sistema decimal a binario.
    a. 4
    b. 27
    c. 45
    d. 220
3) Convertir en sistema octal cada uno de las siguientes cantidades.
    a. 100100
    b. 101101100100
    c. 111111111000111101
    d. 1100110010
4) Convertir en sistema binario las siguientes cantidades:
    a. $3_{8)}$
    b. $40_{8)}$
    c. $1301_{8)}$
    d. $57251_{8)}$
5) Traducir las siguientes cifras hexadecimales en código binario.
    a. F
    b. A2D
    c. ABBA
    d. 19055DE
6) Transformar las siguientes cantidades binarias en sistema hexadecimal.
    a. 1000100000001111
    b. 10110100111100011100
    c. 00100000101100011011110
    d. 011011011011110001000001
7) Resolver las siguientes operaciones aritméticas:
    a. 11010 + 10011
    b. 101011 + 10010
    c. 11010 + 10011 + 101
    d. 10101 + 11010 + 1001
    e. 1000 – 11
    f. 10010 – 101
    g. 11010101 – 1011001
    h. 110100101 – 11101000
    i. 10010 x 111
    j. 11101 x 101
    k. 110100 x 100
    l. 110101 x 110

1.4. Unidades mínimas de información.

El **bit** es la unidad mínima de información, es decir, una cifra binaria (0 ó 1), una posición del circuito abierto (abierto o cerrado) o una posición en una cinta perforada (si hay o no perforación).

Cuando se unen ocho bits forman un conjunto denominado **BYTE**, que procede de la contracción ***binary term*** (término binario) y se conoce como **PALABRA**. En Informática, cada carácter (letra, número o signo de puntuación), suele ocupar un byte y nos sirve para representar acciones computacionales.

Los conjuntos de dígitos binarios reciben un nombre propio en función de su capacidad, estos son:

Ocho bits se conoce como octeto o byte.
1024 bytes forman un kilobyte (Kb).
1024 kilobytes equivalen a un megabyte (Mb).
1024 megabytes constituyen un gigabyte (Gb).
1024 gigabytes son un terabyte (Tb).
1024 terabytes forman un petabyte (Pb).

Estas unidades de medida nos son cotidianas, las vemos en la capacidad de nuestros archivos de música, vídeo, disco duro, tarjetas móviles, entre otros.

Actividades

a) ¿Cuántos bits son 17 bytes?
b) ¿Cuántos bits son 8 Kilobytes?
c) ¿Cuántos bytes son 2 Mb?
d) ¿Cuántos Kb son 5 Tb?
e) ¿Cuántos Mb son 421523 bytes?
f) ¿Qué archivo tendrá más capacidad entre uno de 12,25 Mb, otro de 13458,5 Kb y otro de 0,01 Gb?
g) ¿Cuántos bytes ocuparía tu nombre completo?

1.5. Sistemas de codificación de caracteres.

Cuando una información que originariamente viene representada en un alfabeto es transcrita a un segundo alfabeto, se dice que ha sido codificada. El caso más sencillo de codificación es aquel en el que ambos alfabetos tienen el mismo número de símbolos. Un ejemplo es el sistema de codificación morse.

Las aplicaciones informáticas no sólo manejan datos numéricos, sino también alfanuméricos para la representación de nombres. Por tanto, se necesitará un código capaz de representar números, letras, caracteres especiales, mayúsculas, etc.

El código ASCII (American Standard Code for Information Interchange) es un código binario estándar para la representación de caracteres alfanuméricos. Se trata de un total de 256 caracteres, entre los que destacan:

- ✓ Los 26 códigos que representan las letras mayúsculas A-Z.
- ✓ Los 26 siguientes que simbolizan las letras minúsculas a-z.
- ✓ Los 10 códigos que corresponden a las cifras 0 al 9.
- ✓ Los 32 códigos para los caracteres especiales de teclado.

Por ejemplo, el carácter *M* tiene asignado el código ASCII *77*.

El código ASCII extendido incluye acentos, símbolos y gráficos. Resultará útil tener a mano el código ASCII cuando deseemos teclear una letra o carácter especial y la tecla genera un carácter distinto al deseado. Para lograrlo escribiremos el número indicado en la columna decimal en combinación con la tecla Alt.

El código ASCII – www.elCodigoASCII.com.ar

sigla en inglés de American Standard Code for Information Interchange
(Código Estadounidense Estándar para el Intercambio de Información)

Caracteres de control ASCII

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de línea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

Caracteres ASCII imprimibles

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_	elCodigoASCII.com.ar		

ASCII extendido

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ö	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	