

# Representación de Datos Alfanuméricos

Elementos de Informática

# Representación de Datos

Las limitaciones de la electrónica convencional imponen una sola forma práctica de representación: la lógica de dos estados "0" y "1".

La "representación binaria de datos" significa representar datos por medio de estos ceros y unos, denominados "dígitos binarios" o "bits".

El bit es la unidad mínima de información. El problema a resolver es como los bits pueden representar los datos de una manera útil, y como recuperar posteriormente esos datos.

# Representación de Datos

Si vamos a representar datos de forma normal y que resulte comprensible, necesitaremos grupos de bits que identifiquen al menos.

- los números del 0 al 9 (10 combinaciones de bits)
- las letras del alfabeto (26 combinaciones de bits)
- otros símbolos (\$ # & \* @ % ^ < =)

# Representación de Datos

Cantidad de bits = x	$2^x$	
1	2	0 y 1
2	4	00, 01, 10, 11
3	8	000, 001, 010, 011, 100, 101, 110, 111
4	16	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, ....
5	32	.....
6	64	.....

Utilizando grupos de 7 bits se obtienen  $2^7 = 128$  combinaciones posibles de bits que nos permiten representar 10 números, letras y símbolos.



# Codificación y Decodificación

Los datos que ingresan a una computadora deben ser convertidos a una "representación interna" para que el computador pueda procesarlos. Después de que este proceso es llevado a cabo, los resultados obtenidos son convertidos nuevamente a una "representación externa" para que el usuario reciba datos provistos de significado.

Todo este proceso implica una ***codificación*** y posterior ***decodificación*** de la información.

# Codificación

**Codificar** es aplicar un conjunto de reglas inequívocas que especifiquen la forma en que pueden representarse los datos, de forma tal que sea posible la decodificación ulterior.

La elección de un sistema de codificación adecuado es importante, ya que de esto dependerá la fiabilidad, universalidad y velocidad del sistema.

# Códigos

Existen en la actualidad una infinidad de códigos, algunos que son propios de una máquina o marca específica y otros que son universalmente aceptados.

Algunos ejemplos de estos códigos internacionales son : BCD, EBCDIC, ASCII , etc.

# Códigos

Normalmente, en el almacenamiento de una computadora, se pueden almacenar dos clases de datos:

- ✓ Alfanuméricos (letras, caracteres y nros.)
- ✓ Datos numéricos.

La mayoría de las computadoras se manejan con códigos de 8 bits para la representación de un carácter.



# Representación de Datos Alfanuméricos

En informática hay dos códigos de uso general para representación de caracteres:

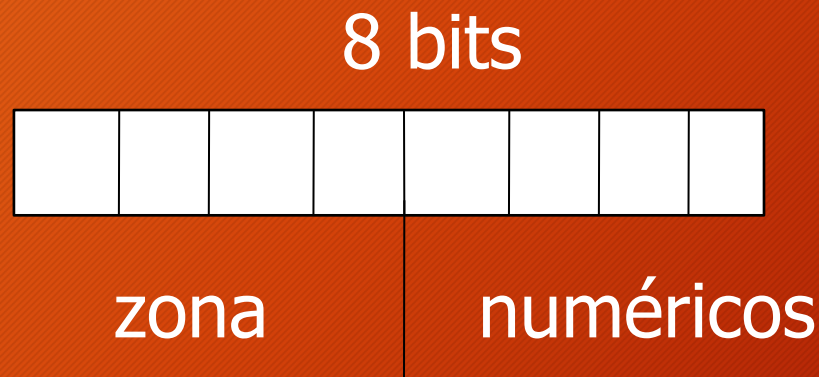
ASCII: American Standard Code for Information Interchange, de uso universal en microprocesadores.

EBCDIC: Extended Binary Coded Decimal Interchange Code; que usan principalmente algunos modelos de "IBM".

# Esquema de Representación

Esquema de representación de un byte en ambos códigos:

La representación de cada carácter se divide en dos partes, una de zona a la izquierda y una numérica a la derecha.



# ASCII - Un Poco de Historia

El código ASCII proviene de las siglas en inglés para American Standard Code for Information Interchange, es decir Código Americano (estadounidense) Estándar para el intercambio de Información. (se pronuncia Aski).

Fue creado en 1963 por el Comité Estadounidense de Estándares o "ASA", organismo que desde 1969 se llama "Instituto Estadounidense de Estándares Nacionales" o "ANSI" como se lo conoce desde entonces.

Este código nació a partir de reordenar y expandir el conjunto de símbolos y caracteres ya utilizados por ese entonces en telegrafía por la compañía Bell.



# ASCII - Un Poco de Historia

En un primer momento solo incluía las letras mayúsculas, pero en 1967 se agregaron las letras minúsculas y algunos caracteres de control, formando así lo que se conoce como US-ASCII, es decir los códigos del 0 al 127.

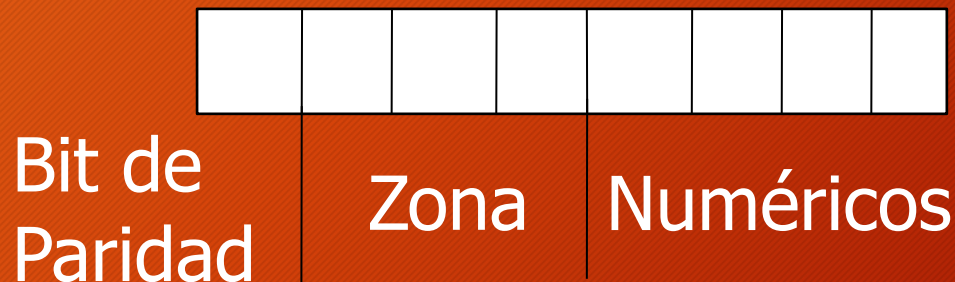
Así con este conjunto de solo 128 caracteres fue publicado en 1967 como estándar, conteniendo todos lo necesario para escribir en idioma ingles.



# ASCII - 7

Este código utiliza 7 bits que proporciona  $2^7 = 128$  combinaciones diferentes.

El octavo bit o bit mas significativo, es decir el de la izquierda, es el bit de paridad; utilizado para controlar si la transmisión del byte tiene errores.



# Bit de Paridad

El bit de paridad se utiliza como un método de detección de errores.

La paridad puede ser par o impar. Esto implica que la cantidad de 1's en el paquete de datos, para nuestro caso el byte, debe tener una cantidad par o impar de unos, según la paridad elegida.

Generalmente se utiliza paridad par. Ejemplos:

1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

Bit de Paridad

0	1	0	1	1	1	1	1
---	---	---	---	---	---	---	---

Bit de Paridad

# ASCII - 7

- Cada dígito (0 al 9) tiene 011 como la parte de zona del byte y su representación binaria como la parte numérica.

Número decimal	Zona	Numérica
0	011	0000
1	011	0001
2	011	0010
3	011	0011
4	011	0100
5	011	0101
6	011	0110
7	011	0111
8	011	1000
9	011	1001

# ASCII - 7

Los caracteres alfabéticos (mayúsculas) están distribuidos en dos grupos según si sus bytes tienen 100 o 101 como la parte de zona.

Las tablas de códigos nos muestran las equivalencias entre los caracteres y su representación binaria, decimal, hexadecimal u octal.

Por lo tanto debemos plantear una manera de poder encontrar representaciones equivalentes a partir de la representación mostrada en la tabla.



# Tabla del Código ASCII 7 bits

Bits menos significativos

Zona Bits más significativos

*bits de zona.*

	000	001	010	011	100	101	110	111
	0	1	2	3	4	5	6	7
0000	NUL	DLE	SP	0	@	P		p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL
	15	31	47	63	79	95	111	127

Representación Decimal

Tabla del código ASCII 7 bits

# ASCII 7

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
0 0 NUL	16 10 DLE	32 20 (space)	48 30 0
1 1 SOH	17 11 DC1	33 21 !	49 31 1
2 2 STX	18 12 DC2	34 22 "	50 32 2
3 3 ETX	19 13 DC3	35 23 #	51 33 3
4 4 EOT	20 14 DC4	36 24 \$	52 34 4
5 5 ENQ	21 15 NAK	37 25 %	53 35 5
6 6 ACK	22 16 SYN	38 26 &	54 36 6
7 7 BEL	23 17 ETB	39 27 '	55 37 7
8 8 BS	24 18 CAN	40 28 (	56 38 8
9 9 TAB	25 19 EM	41 29 )	57 39 9
10 A LF	26 1A SUB	42 2A *	58 3A :
11 B VT	27 1B ESC	43 2B +	59 3B ;
12 C FF	28 1C FS	44 2C ,	60 3C <
13 D CR	29 1D GS	45 2D -	61 3D =
14 E SO	30 1E RS	46 2E .	62 3E >
15 F SI	31 1F US	47 2F /	63 3F ?

ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo	ASCII Hex Simbolo
64 40 @	80 50 P	96 60 `	112 70 p
65 41 A	81 51 Q	97 61 a	113 71 q
66 42 B	82 52 R	98 62 b	114 72 r
67 43 C	83 53 S	99 63 c	115 73 s
68 44 D	84 54 T	100 64 d	116 74 t
69 45 E	85 55 U	101 65 e	117 75 u
70 46 F	86 56 V	102 66 f	118 76 v
71 47 G	87 57 W	103 67 g	119 77 w
72 48 H	88 58 X	104 68 h	120 78 x
73 49 I	89 59 Y	105 69 i	121 79 y
74 4A J	90 5A Z	106 6A j	122 7A z
75 4B K	91 5B [	107 6B k	123 7B {
76 4C L	92 5C \	108 6C l	124 7C
77 4D M	93 5D ]	109 6D m	125 7D }
78 4E N	94 5E ^	110 6E n	126 7E ~
79 4F O	95 5F _	111 6F o	127 7F

bits de zona.

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000 0	NUL 0	DLE 16	SP 32	@ 48	P 64	Q 80		p 112
0001 1	SOH 1	DC1 17	! 33	A 49	B 65	R 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 del código ASCII 7 bits

bits de zona.

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
000 0	NUL 0	DLE 16	SP 32	@ 48	P 64	Q 80		p 112
001 1	SOH 1	DC1 17	! 33	A 49	B 65	R 81	a 97	q 113
010 2	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
011 3	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
100 4	EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
101 5	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
110 6	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
111 7	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
000 0	2 <sup>6</sup> 64	2 <sup>5</sup> 32	2 <sup>4</sup> 16	2 <sup>3</sup> 8	2 <sup>2</sup> 4	2 <sup>1</sup> 2	2 <sup>0</sup> 1	
001 1	64	0	16	0	0	0	0	80
010 2	64	0	0	0	0	0	1	65
011 3	64	0	16	8	0	2	0	90

Tabla del código ASCII 7 bits

$3 - 2 = 1$ 
$$5-4=1$$

bits de zona.

Tabla del código ASCII 7 bits

ASCII Hex	Simbolo	ASCII Hex	Simbolo	ASCII Hex	Simbolo	ASCII Hex	Simbolo
16 10	DLE	32 20	(space)	48 30	0		
17 11	DC1	33 21	!	49 31	1		
18 12	DC2	34 22	"	50 32	2		
19 13	DC3	35 23	#	51 33	3		
20 14	DC4	36 24	\$	52 34	4		
21 15	NAK	37 25	%	53 35	5		
22 16	SYN	38 26	&	54 36	6		
23 17	ETB	39 27	'	55 37	7		
24 18	CAN	40 28	(	56 38	8		
25 19	EM	41 29	)	57 39	9		
26 1A	SUB	42 2A	*	58 3A	:		
27 1B	ESC	43 2B	+	59 3B	;		
28 1C	FS	44 2C	,	60 3C	<		
29 1D	GS	45 2D	-	61 3D	=		
30 1E	BS	46 2E	.	62 3E	>		
1	0	0	0	0	0	1	
ASCII Hex	Simbolo	ASCII Hex	Simbolo	ASCII Hex	Simbolo	ASCII Hex	Simbolo
1	1	0	1	1	0	1	
B 82 52	R	98 62	D	114 72	r		
C 83 53	S	99 63	d	115 73	s		
D 84 54	T	100 64	D	116 74	t		
1	1	0	1	1	1	1	
H 88 58	X	104 68	h	120 78	x		
I 89 59	Y	105 69	i	121 79	y		
1	1	1	0	0	1	0	
M 93 5D	M	109 6D	m	125 7D	}		
N 94 5E	^	110 6E	n	126 7E	~		
O 95 5F	_	111 6F	o	127 7F			



# Ejercicio

Completa con el bit de paridad, teniendo en cuenta la paridad par en los siguientes bytes.

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

0	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

# ASCII 8

Existe también el código ASCII - 8; este código utiliza los 8 bits para almacenar un carácter y el computador almacena un bit extra o noveno bit llamado bit de paridad.

En 1981, la empresa IBM desarrolló una extensión de 8 bits del código ASCII, en esta versión se reemplazaron algunos caracteres de control obsoletos.

# ASCII 8

Se incorporaron 128 caracteres nuevos, tales como símbolos, signos, gráficos adicionales y letras latinas, necesarias para la escrituras de textos en otros idiomas, como por ejemplo el español. Así fue como se agregaron los caracteres que van del ASCII 128 al 255.

# ASCII 8

## 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 sinc)
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 (Página de código 437)

128	Ç	160	á	192	Ł	224	Ó
129	ü	161	í	193	ł	225	ô
130	é	162	ó	194	Ł	226	Ô
131	â	163	ú	195	ł	227	Ò
132	ä	164	ñ	196	—	228	ö
133	à	165	Ñ	197	†	229	Õ
134	á	166	ª	198	ā	230	μ
135	ç	167	º	199	Ā	231	þ
136	ê	168	¿	200	Ĺ	232	þ
137	ë	169	®	201	Œ	233	Ú
138	è	170	¬	202	Œ	234	Û
139	ï	171	½	203	Œ	235	Ü
140	î	172	¼	204	Œ	236	ý
141	ì	173	ı	205	=	237	Ÿ
142	Ā	174	«	206	†	238	—
143	Ă	175	»	207	‡	239	·
144	É	176		208	ø	240	≡
145	æ	177		209	Ð	241	±
146	Æ	178		210	Ê	242	≡
147	ô	179		211	Ë	243	¼
148	ö	180	—	212	È	244	¶
149	ò	181	Ā	213	İ	245	§
150	û	182	Ă	214	Í	246	÷
151	ù	183	Ā	215	Î	247	°
152	ÿ	184	©	216	Ï	248	°
153	Œ	185	Œ	217	Œ	249	·
154	Ů	186	Œ	218	Œ	250	·
155	ø	187	Œ	219	Œ	251	·
156	£	188	Œ	220	Œ	252	·
157	Ø	189	¢	221	Œ	253	·
158	×	190	¥	222	Œ	254	■
159	f	191	Œ	223	Œ	255	nbsp



# ASCII-8

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
10000000	128	Ç	10100000	160	à	11000000	192	+	11100000	224	Ô
10000001	129	ü	10100001	161	í	11000001	193	-	11100001	225	ß
10000010	130	é	10100010	162	ó	11000010	194	-	11100010	226	Ô
10000011	131	â	10100011	163	ú	11000011	195	+	11100011	227	Ò
10000100	132	ä	10100100	164	ñ	11000100	196	-	11100100	228	ø
10000101	133	å	10100101	165	Ñ	11000101	197	+	11100101	229	Ö
10000110	134	â	10100110	166	ª	11000110	198	ä	11100110	230	µ
10000111	135	ç	10100111	167	•	11000111	199	Ä	11100111	231	þ
10001000	136	ê	10101000	168	¿	11001000	200	+	11101000	232	Ð
10001001	137	ë	10101001	169	®	11001001	201	+	11101001	233	Ù
10001010	138	è	10101010	170	¬	11001010	202	-	11101010	234	Û
10001011	139	ï	10101011	171	½	11001011	203	-	11101011	235	Ü
10001100	140	ì	10101100	172	¼	11001100	204	-	11101100	236	ý
10001101	141	í	10101101	173	½	11001101	205	-	11101101	237	ÿ
10001110	142	Ä	10101110	174	«	11001110	206	+	11101110	238	-
10001111	143	Å	10101111	175	»	11001111	207	□	11101111	239	-
10010000	144	È	10110000	176	-	11010000	208	ø	11110000	240	-
10010001	145	æ	10110001	177	-	11010001	209	Ð	11110001	241	±
10010010	146	Æ	10110010	178	-	11010010	210	Ê	11110010	242	-
10010011	147	ø	10110011	179	-	11010011	211	Ë	11110011	243	¼
10010100	148	ø	10110100	180	-	11010100	212	E	11110100	244	¶
10010101	149	ó	10110101	181	À	11010101	213	í	11110101	245	§
10010110	150	ô	10110110	182	Á	11010110	214	Î	11110110	246	+
10010111	151	ù	10110111	183	Â	11010111	215	Ï	11110111	247	-
10011000	152	ý	10111000	184	©	11011000	216	Ï	11111000	248	ó
10011001	153	Ö	10111001	185	-	11011001	217	+	11111001	249	-
10011010	154	Û	10111010	186	-	11011010	218	+	11111010	250	-
10011011	155	ü	10111011	187	+	11011011	219	-	11111011	251	ı
10011100	156	£	10111100	188	+	11011100	220	-	11111100	252	ı
10011101	157	Ø	10111101	189	¢	11011101	221	-	11111101	253	ı
10011110	158	×	10111110	190	¥	11011110	222	İ	11111110	254	-
10011111	159	f	10111111	191	+	11011111	223	-	11111111	255	-

# Ejercicio

Convierte a hexadecimal el siguiente mensaje y luego tradúcelo al lenguaje español:

01011001 01001111 01001101  
01000101 01010001 01010101  
01000101 01000100 01001111  
01000101 01001110 01000011  
01000001 01010011 01000001

Carácter	ASCII	Carácter	ASCII
A	0100 0001	W	0101 0111
B	0100 0010	X	0101 1000
C	0100 0011	Y	0101 1001
D	0100 0100	Z	0101 1010
E	0100 0101	0	0011 0000
F	0100 0110	1	0011 0001
G	0100 0111	2	0011 0010
H	0100 1000	3	0011 0011
I	0100 1001	4	0011 0100
J	0100 1010	5	0011 0101
K	0100 1011	6	0011 0110
L	0100 1100	7	0011 0111
M	0100 1101	8	0011 1000
N	0100 1110	9	0011 1001
O	0100 1111	+	0010 1011
P	0101 0000	-	0010 1101
Q	0101 0001	*	0010 1010
R	0101 0010	:	0011 1010
S	0101 0011	=	0011 1101
T	0101 0100	<	0011 1100
U	0101 0101	;	0011 1011
V	0101 0110		

# EBCDIC

El código ampliado decimal codificado en binario es un código de 8 bits utilizado principalmente por los equipos "IBM" y compatibles.

Se forma:

- Cada **dígito numérico** tiene **1111** como la parte de zona del byte y su representación binaria, como la parte numérica.
- Los **caracteres alfabéticos** (mayúsculas) están distribuidos en tres grupos, ya sea que los bytes tengan 1100; 1101 o 1110 como bits de zona.



# EBCDIC

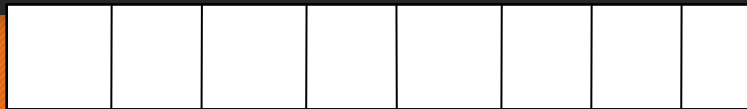
Ejemplos:

<b>Carácter</b>	<b>zona</b>	<b>numérica</b>
A	1100	0001
B	1100	0010
1	1111	0001
2	1111	0010

La codificación se comprende observando el siguiente diagrama.



# EBCDIC



8 bits

Zona

Numérico

0001 a 1001 = A a I  
0001 a 1001 = J a R  
0010 a 1001 = S a Z  
0000 a 1001 = 0 a 9

00 = A a I  
01 = J a R  
10 = S a Z  
11 = NUMÉRICO

11 = MAYÚSCULAS Y NUMÉRICO  
10 = MINÚSCULAS  
01 = CARACTERES ESPECIALES  
00 = SIN ASIGNACIÓN

# Tabla EBCDIC

A8	A7	A6	A5	A4	A3	A2	A1	LSB	MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
PF perfo hors fonction LC en code RES restauration IL neutre BYP ignorer SM changement de mode PN perfo en fonction RS arrêt lecteur UC hors code	0	0	0	0	0	0	0	0	NUL	DLE				SP	&	-					{	}	\		0
	0	0	0	1	1	0	0	1	1	SOH	DC1					/		a	j	~		A	J		1
	0	0	1	0	2	0	0	1	0	STX	DC2	FS	SYN					b	k	s		B	K	S	2
	0	0	1	1	3	0	0	1	1	ETX	DC3							c	l	t		C	L	T	3
	0	1	0	0	4	0	0	1	0	PF	RES	BYP	PN					d	m	u		D	M	U	4
	0	1	0	1	5	0	0	1	1	HT	NL	LF	RS					e	n	v		E	N	V	5
	0	1	1	0	6	0	0	1	0	LC	BS	ETB	UC					f	o	w		F	O	W	6
	0	1	1	1	7	0	0	1	1	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
	1	0	0	0	8	0	0	0	0		CAN							h	q	y		H	Q	Y	8
	1	0	0	1	9	0	0	1	1		EM							i	r	z		I	R	Z	9
	1	0	1	0	A	0	0	1	0			SM		Ø	!	,	:								
	1	0	1	1	B	0	0	1	1	VT				.	\$	%	#								
	1	1	0	0	C	0	0	0	0	FF	IFS		DC4	<	*	_	@								
	1	1	0	1	D	0	0	1	0	CR	IGS	ENQ	NAK	(	)	<	.								
	1	1	1	0	E	0	0	0	0	SO	IRS	ACK		+	;	?	=								
	1	1	1	1	F	0	0	1	1	SI	IUS	BEL	SUB		^		..								

# Ejercicio

Escribe el día de la semana que es hoy utilizando el código EBCDIC