

## Código binario

El código binario permite entender y diseñar sistemas electrónicos digitales. Su sistema de numeración es de vital importancia para la electrónica. También existen distintas configuraciones como BDC, ASCII, etc. Los códigos binarios son un arreglo de dos caracteres, 1 y 0. Entonces, con sólo estos dos dígitos se pueden representar a todo el conjunto de números. También se pueden realizar operaciones lógicas y aritméticas. Incluso se pueden representar números imaginarios y reales. Dicha numeración se puede representar o medir en bits de información. Por ejemplo, 8 bits de información permiten representar hasta  $2^8 = 256$  combinaciones, desde 0 hasta 255. Cuando se crea un número con signo, el bit 7 se usa para representar al signo, 0 en el último bit representaría el + y 1 el menos. Por lo tanto un número con signo de 8 bits, puede representar valores desde -127 hasta +127.

### ¿Qué es?

Un código binario y su sistema de numeración tienen dos características: la posición y el valor que se le asigna a esa posición. Recordarán por ejemplo, cuando aprendíamos los números decimales, que había unidades, decenas, centenas, millares, etc. Entonces de la misma forma, los códigos binarios ó también llamados sistemas de numeración binaria, presentan un arreglo similar.

Por ejemplo, las posiciones se generan de derecha a izquierda comenzando por el cero. Cada posición representa el exponente de una base 2. Por ejemplo, tome el siguiente Número binario:

### Importancia en la electrónica del código binario

Todos los sistemas electrónicos digitales funcionan bajo el principio del código binario. Existen dos posibles estados llamados: verdadero-falso, alto-bajo ó 1-0. Estos dos estados son representaciones abstractas de regiones de voltajes específicos. Por ejemplo, para los circuitos integrados que operan a 5VDC, un 0 representa un voltaje entre -0.5V y 1.2V. Mientras que el 1 es representado por valores entre 2.2 y 5.5V.

Las compuertas lógicas utilizan el código binario para poder representar valores en sus entradas y proporcionar resultados. Las compuertas lógicas son los bloques con los que están contruidos los flip-flops, sumadores, multiplexores, multiplicadores, decodificadores, registros (memorias), registros de desplazamiento (módulos de comunicación serial), etc. A su vez, estos bloques más complejos forman sistemas como procesadores, memorias, periféricos y unidades de control que son el principio de funcionamiento de los microprocesadores y los microcontroladores.

Para la electrónica, en especial en los sistemas embebidos como los microcontroladores o los microprocesadores, se utiliza el código binario para codificar información digital. Por ejemplo, dicha información se puede transportar mediante pines de entrada/salida de estos dispositivos digitales. Algunos ejemplos de puertos de comunicación serial son: puerto serial, I2C, SPI, CAN, USB o JTAG. Entonces, conocer cómo funciona y se codifican/decodifican los números binarios en bits/bytes, etc., es importante para entender los protocolos de comunicación digitales.

Variables y su relación con el código binario

También se utiliza en la codificación de las variables en cualquier lenguaje de programación. Como por ejemplo, C/C++, el lenguaje Arduino, el lenguaje MBED, etc. Por ejemplo, los tipos básicos de variables en C serían:

- char – 8 bit's, con signo, valores desde -127 hasta +128.
- unsigned char – 8 bts, sin signo, y con valores de 0 a 255.
- int (dependiendo del procesador, 16, 32, 64 bits).
- unsigned int (también desde 16 hasta 64 bits).
- etc

### Agrupaciones de los números binarios

Los códigos binarios se agrupan para poder ser manejados de una mejor manera. Por ejemplo, la cantidad mínima de información que se puede representar en un sistema binario es el bit. Pero también existe la agrupación de 4 bits, se llama nibble. Si agrupamos 8 bit o 2 nibble, ya se llama byte y así sucesivamente: Bit: Un bit, es la cantidad mínima de información que se puede representar en un sistema digital. Sólo puede valer 0 o 1. Verdadero o Falso.

**Nibble:** Un nibble es la agrupación de 4 bits. Este puede representar a todas las combinaciones con 4 bits, que son 16 combinaciones posibles.

**Byte:** Un byte es la agrupación binaria más común y ocurre cuando se agrupan 8 bits. Un byte es muy importante debido a que al inicio, los procesadores eran de 8 bits (hubo incluso procesadores de 4 bits). Por lo tanto, la información o los resultados se guardaban en memorias de 8 bits para ser más eficientes. En un byte de información se pueden guardar variables del tipo char. El código ascii, es una codificación binaria que puede ser guardada en chars.

Para las variables del tipo int, se requieren de entre 2 y 4 bytes dependiendo de la arquitectura de los procesadores.

**Kbyte:** Un kbyte es la agrupación de 1000 bytes. Aunque algunos autores incluso mencionan que son 1024 bytes. Dependiendo de la agrupación, ambas consideraciones pueden ser igualmente válidas.

**Mbyte**

Un Mbyte es la agrupación de 1 millón u 8 millones de bits.

**KBit, MBit y Gbit:** El Kbit es distinto al kbyte, en el sentido que representa sólo 1000 bits. Es una métrica utilizada en los protocolos de comunicación digital que son seriales. Dado que los datos se transportan uno a la vez, es más recomendable agruparlos en bits, en lugar de bytes. Esta medida se utiliza mucho por ejemplo para medir la velocidad del internet.

Generalmente pensamos que los kbits son 1000 bytes, pero esto es incorrecto. Cuando nuestros planes de internet dicen que tenemos 4GB, pensamos en 4 mil millones de bytes, pero en realidad son 4 mil millones de bits.

#### *Código 8421 para código binario*

El código 8421 permite formar números binarios de forma muy sencilla. Sólo se requiere recordar a estos cuatro números decimales. Con estos se pueden formar todos los números de 0 a 15. Por ejemplo, si queremos formar el número 7 en binario, sería:  $4+2+1$ , por lo tanto el número que representa al 8 sería 0 y el numero quedaría como:

0111

Ejemplo-2. Formar el número 12 en binario. El doce es la combinación o suma de  $8+4$ . Entonces el 1 que representa al 8 y al 4 se colocan y el resto se pone en ceros.

#### *Código BCD*

La codificación BCD es muy utilizada para poder representar a los números de un reloj de tiempo real (en la electrónica de sistemas embebidos). Es una codificación similar a la 8421 en agrupaciones de 4 bits. Por ejemplo.

El siguiente numero binario, en la codificación normal, representa al 55: 0011 0111. Por el contrario, en la codificación BCD se representa al 5 en cuatro bits en 8421 y queda como: 0101, por lo tanto, en BCD quedaría como: 0101 0101.

Notas	Binario 8421	Hexadecimal	Decimal
0+0+0+0	0000	0	0
0+0+0+1	0001	1	1
0+0+2+0	0010	2	2
0+0+2+1	0011	3	3
0+4+0+0	0100	4	4
0+4+0+1	0101	5	5
0+4+2+0	0110	6	6
0+4+2+1	0111	7	7
8+0+0+0	1000	8	8
8+0+0+1	1001	9	9
8+0+2+0	1010	A	10
8+0+2+1	1011	B	11
8+4+0+0	1100	C	12
8+4+0+1	1101	D	13
8+4+2+0	1110	E	14
8+4+2+1	1111	F	15

### Código ASCII

El código ASCII es una forma de representar caracteres de un sistema de escritura. Para las regiones donde utilizamos un abecedario de caracteres occidentales, se llama UTF-8. Consiste de 8 bits que representan a los caracteres.

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

Fue creado en 1963 por el Comité Estadounidense de Estándares o "ASA", este organismo cambio su nombre en 1969 por "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 en aquel momento en telegrafía por la compañía Bell. En un primer momento solo incluía letras mayúsculas y números, 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 caracteres 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 inglés.

En 1981, la empresa IBM desarrolló una extensión de 8 bits del código ASCII, llamada "página de código 437", en esta versión se reemplazaron algunos caracteres de control obsoletos, por caracteres gráficos. Además se incorporaron 128 caracteres nuevos, con 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 sumaron los caracteres que van del ASCII 128 al 255.

IBM incluyó soporte a esta página de código en el hardware de su modelo 5150, conocido como "IBM-PC", considerada la primera computadora personal. El sistema operativo de este modelo, el "MS-DOS" también utilizaba el código ASCII extendido.

Casi todos los sistemas informáticos de la actualidad utilizan el código ASCII para representar caracteres, símbolos, signos y textos (147).

#### Como utilizar el código ASCII:

Sin saberlo lo utilizas todo el tiempo, cada vez que utilizas algún sistema informático; pero si lo que necesitas es obtener algunos de los caracteres no incluidos en tu teclado debes hacer lo siguiente:

Como escribir con el teclado, o tipear: Letra ÑE mayúscula - letra N con tilde - ENIE

WINDOWS: en computadoras con sistema operativo como Windows 8, Win 7, Vista, Windows Xp, etc.  
Para obtener la letra, caracter, signo o símbolo "Ñ": (Letra EÑE mayúscula - letra N con tilde - ENIE) en ordenadores con sistema operativo Windows:

- 1) Presiona la tecla "Alt" en tu teclado, y no la sueltes.
- 2) Sin dejar de presionar "Alt", presiona en el teclado numérico el número "165", que es el número de la letra o símbolo "Ñ" en el código ASCII.
- 3) Luego deja de presionar la tecla "Alt" y... ¡Ya está listo! (148).

Caracteres ASCII de control			Caracteres ASCII imprimibles			ASCII extendido (Página de código 437)										
00	NULL	(carácter nulo)	32	espacio	64	@	96	ı	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(inicio encabezado)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
02	STX	(inicio texto)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ö
03	ETX	(fin de texto)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	Û
04	EOT	(fin transmisión)	36	\$	68	D	100	d	132	ä	164	ñ	196	Ł	228	ö
05	ENQ	(consulta)	37	%	69	E	101	e	133	à	165	ñ	197	ł	229	Û
06	ACK	(reconocimiento)	38	&	70	F	102	f	134	å	166	²	198	Ł	230	µ
07	BEL	(timbre)	39	'	71	G	103	g	135	ç	167	°	199	Ł	231	þ
08	BS	(retroceso)	40	(	72	H	104	h	136	ê	168	¿	200	Ł	232	p
09	HT	(tab horizontal)	41	)	73	I	105	i	137	ë	169	©	201	Ł	233	Ů
10	LF	(nueva línea)	42	*	74	J	106	j	138	è	170	¬	202	Ł	234	Ù
11	VT	(tab vertical)	43	+	75	K	107	k	139	ï	171	½	203	Ł	235	Ú
12	FF	(nueva página)	44	,	76	L	108	l	140	î	172	¾	204	Ł	236	ý
13	CR	(retorno de carro)	45	-	77	M	109	m	141	ï	173	¸	205	Ł	237	Ÿ
14	SO	(desplaza afuera)	46	.	78	N	110	n	142	Ā	174	«	206	Ł	238	ı
15	SI	(desplaza adentro)	47	/	79	O	111	o	143	Ă	175	»	207	Ł	239	ˆ
16	DLE	(esc.vínculo datos)	48	0	80	P	112	p	144	Ē	176	¼	208	Ł	240	≡
17	DC1	(control disp. 1)	49	1	81	Q	113	q	145	æ	177	½	209	Ł	241	±
18	DC2	(control disp. 2)	50	2	82	R	114	r	146	Æ	178	¾	210	Ł	242	≡
19	DC3	(control disp. 3)	51	3	83	S	115	s	147	ø	179	¾	211	Ł	243	¼
20	DC4	(control disp. 4)	52	4	84	T	116	t	148	ö	180	¾	212	Ł	244	ŧ
21	NAK	(conf. negativa)	53	5	85	U	117	u	149	õ	181	À	213	Ł	245	§
22	SYN	(inactividad sinc)	54	6	86	V	118	v	150	ü	182	Á	214	Ł	246	÷
23	ETB	(fin bloque trans)	55	7	87	W	119	w	151	ù	183	Â	215	Ł	247	ˆ
24	CAN	(cancelar)	56	8	88	X	120	x	152	ý	184	Ã	216	Ł	248	ˆ
25	EM	(fin del medio)	57	9	89	Y	121	y	153	ÿ	185	Ä	217	Ł	249	ˆ
26	SUB	(sustitución)	58	:	90	Z	122	z	154	Û	186	Å	218	Ł	250	ˆ
27	ESC	(escape)	59	;	91	[	123	{	155	ø	187	Ä	219	Ł	251	ˆ
28	FS	(sep. archivos)	60	<	92	\	124		156	£	188	Å	220	Ł	252	ˆ
29	GS	(sep. grupos)	61	=	93	]	125	}	157	Ø	189	Æ	221	Ł	253	ˆ
30	RS	(sep. registros)	62	>	94	^	126	~	158	×	190	¥	222	Ł	254	ˆ
31	US	(sep. unidades)	63	?	95	_			159	f	191	ſ	223	Ł	255	nbsp
127	DEL	(suprimir)														

de uso frecuente (idioma español)	vocales con acento (acento agudo español)	vocales con diéresis	símbolos matemáticos	símbolos comerciales	comillas, llaves, paréntesis
ñ alt + 164	á alt + 160	ä alt + 132	½ alt + 171	\$ alt + 36	" alt + 34
Ñ alt + 165	é alt + 130	ë alt + 137	¼ alt + 172	£ alt + 156	' alt + 39
@ alt + 64	í alt + 161	ï alt + 139	¾ alt + 243	¥ alt + 190	( alt + 40
¿ alt + 168	ó alt + 162	ö alt + 148	² alt + 251	¢ alt + 189	) alt + 41
? alt + 63	ú alt + 163	u alt + 129	³ alt + 252	¤ alt + 207	[ alt + 91
! alt + 173	Á alt + 181	Ä alt + 142	⁴ alt + 253	© alt + 169	] alt + 93
! alt + 33	É alt + 144	È alt + 211	f alt + 159	® alt + 184	{ alt + 123
: alt + 58	Í alt + 214	İ alt + 216	± alt + 241	ª alt + 166	} alt + 125
/ alt + 47	Ö alt + 224	Ô alt + 153	× alt + 158	º alt + 167	« alt + 174
\ alt + 92	Ü alt + 233	Ů alt + 154	÷ alt + 246	° alt + 248	» alt + 175

Nota: para mayor información de cómo se trabaja con conversiones a base 2 o binario revisen en video que les puse en las fuentes.

Fuentes:

<https://elcodigoascii.com.ar/>

<https://hetpro-store.com/TUTORIALES/codigo-binario/>

<https://www.youtube.com/watch?v=lqFaPj6BYi4>

