



Representación de los Datos en Memoria

LECCIÓN 1

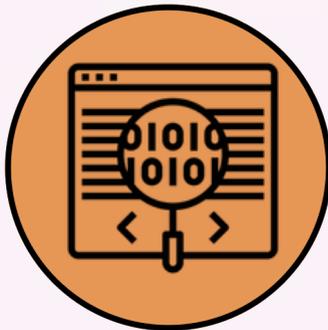


La memoria es la parte de los equipos de cómputo que almacena todos los datos; números, direcciones, procesos, identificadores y referencias necesarias para que toda operación funcione. Esta se compone únicamente de valores binarios representados por números uno y cero; a esta unidad se le conoce como bits. Un bit, por lo tanto, sólo es capaz de almacenar un dígito binario de información que puede tener uno de dos valores: un uno (1) o un cero (0).

Los bits se suelen agrupar en diferentes cantidades, por ejemplo, los grupos de 8 bits se les conoce como byte. Los bytes forman muchas representaciones de datos y sirven para almacenar datos numéricos y datos alfabéticos (letras).



Tanto que los números son infinitos y continuos, esto quiere decir que, si se quiere representar todo tipo de números en memoria, se necesitaría infinitas combinaciones de registros binarios de memoria, lo que es imposible. Por este motivo, es necesario definir tipos de datos, que son representaciones en memoria lo suficientemente aproximadas para representar sistemas numéricos y sistemas de información con la menor pérdida posible. Por este motivo hay que diferenciar cómo se guarda cada dato en memoria, definiendo los tipos de datos.



Usualmente, los tipos de datos son determinados por las capacidades del hardware y del software de un sistema de cómputo. Para evitar problemas de comunicación entre sistemas, todo software se apega a algún estándar, por ejemplo, el IEEE754 que define cómo deben ser las operaciones aritméticas de ciertos datos y su representación.

Esta norma, generalmente, define: El Formato de Representación que, a su vez, define dos formatos de representación: Precisión Simple (32 bits) y Precisión Doble (64 bits).

El formato de precisión simple utiliza 32 bits para representar un número, divididos en un bit de signo, ocho bits de exponente y 23 bits de mantisa. El formato de precisión doble utiliza 64 bits, con un bit de signo, 11 bits de exponente y 52 bits de mantisa. Similar a como se muestra en la figura 1

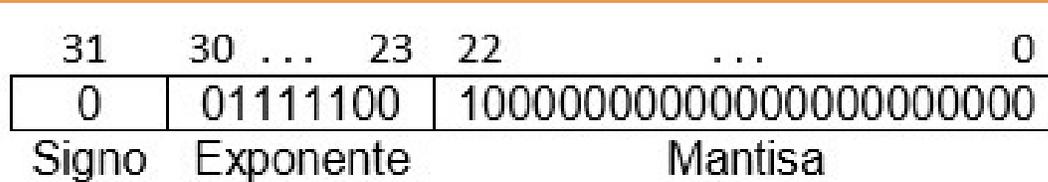


Figura 1: representación de números de coma flotante.

En la Figura 1, se puede observar la colección de unos y ceros que representan un número. La forma de almacenar la información, es utilizando un bit para el signo (positivo / negativo) algunos bits (dependiendo del standard) para el exponente, que representa a qué potencia de 10 está elevado el número (se representa en notación científica) y la mantisa es el número como tal representado en unidades decimales conformadas por el inverso de las potencias de 2, como se muestra en la figura 2:

Figura 2: Representación de decimales con el sistema binario.

<u>Sistema Binario</u>	<u>Sistema Decimal</u>
.1	$\frac{1}{2} = .5$
.01	$\frac{1}{4} = .25$
.001	$\frac{1}{8} = .125$
.0001	$\frac{1}{16} = .0625$
.00001	$\frac{1}{32} = .03125$
.000001	$\frac{1}{64} = .015625$
.0000001	$\frac{1}{128} = .0078125$
.00000001	$\frac{1}{256} = .00390625$

Figura 2: Representación de decimales con el sistema binario.



Es importante tener en cuenta estos detalles de los tipos de datos, porque cambia fundamentalmente las cantidades que pueden ser representadas y sus operaciones dependiendo del tipo de datos. Por ejemplo, en muchos lenguajes de programación, la comparación:

$$0.1 + 0.2 == 0.3$$



Da como resultado falso. Ese comportamiento es precisamente debido a que no todos los números flotantes se pueden representar, por lo que las aproximaciones generan ciertas diferencias en decimales pequeños. Asimismo para caracteres usualmente se emplean bytes, cada uno guardando alguna letra de la tabla ASCII. Los números de 8 bits (bytes) pueden representar combinaciones desde cero (0) hasta el doscientos cincuenta y cinco (255). La tabla ASCII muestra las equivalencias entre tales números binarios (o decimales) y los caracteres que los computadores son capaces de representar.

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

Figura 3: Tabla ASCII