

Representación Interna de la información



Representación Interna de la información

- **Sistemas de numeración: Código binario**
- **Representación de enteros**
- **Representación de caracteres**
- **Representación de reales**
- **Representación de tipos complejos**



Sistemas de numeración

- **Decimal, (Base 10)**

- 10 Símbolos (0,1,2,3,4,5,6,7,8,9)
- $234 = 2 * 10^2 + 3 * 10^1 + 4 * 10^0$

- **Binario (Base 2)**

- 2 Símbolos (0, 1)
- $11011 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = \mathbf{27}$

- **Octal (Base 8)**

- 8 Símbolos (0, 1,2,3,4,5,6,7)
- $231 = 2 * 8^2 + 3 * 8^1 + 1 * 8^0 = \mathbf{153}$

- **Hexadecimal (Base 16)**

- 16 Símbolos (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
- $A1C = A_{(10)} * 16^2 + 1 * 16^1 + C_{(12)} * 16^0 = \mathbf{2588}$



El código binario

- **Ver el video:**

<https://youtu.be/f9b0wwhTmeU>

Pasar de binario a decimal → Multiplicar por la potencia de 2: 1,2,4,8,16,32,....

- $10_{(2)} = 1 \times 2 + 0 \times 1 = 2$
- $100_{(2)} = 1 \times 4 + 0 \times 2 + 0 \times 1 = 4$
- $101_{(2)} = 1 \times 4 + 0 \times 2 + 1 \times 1 = 5$
- $1010_{(2)} = 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 10$
- $10110_{(2)} = 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 22$



El código binario

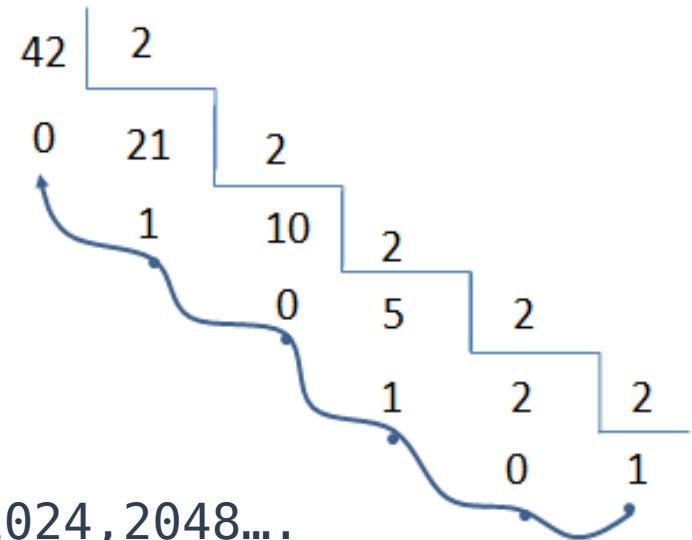
- **Pasar de decimal a binario:** Método de restas sucesivas, hasta que el resto sea 0 o 1.

- $42_{(10)} \rightarrow 101010_{(2)}$

- Otro método: Suma de las potencias

de dos: 1, 2, 4, 8, 16, 32, 128, 256, 512, 1024, 2048...

$$42 = 32 + 8 + 2 = 100000 + 1000 + 10 = 101010$$



Representación de enteros

- El ordenador representa los enteros en grupos de bytes 1,2,3,4 (8,16,32,64 bits) → Tiene un valor máximo, no existe el infinito.
- Ej .- El n.º 12 en dos byte 00000000 | 00001100 |

1º byte2 byte

¿Cual es el entero más grande con dos byte?

|11111111|11111111| → 65535

Número más grandes → más byte

Ojo: *La mayor parte de los lenguajes de programación no controlan el **desbordamientos**: Efecto 2000 y efecto 2038*



Representación de enteros

- Suma de números binarios:

```
  00001110   14
+ 00000101   +5
-----
  00010011   19
```

$0+0 = 0$	$0+1 = 1$
$1+0 = 1$	$1+1 = 10$ (0 y me llevo 1)

- los números negativos

– Signo y magnitud **0**0000010 → 2

10000010 → -2

- Complemento a 2 (Invertir y sumar 1)

2 → 00000010 -2 → 11111101 +1 → 11111110



Representación de números reales

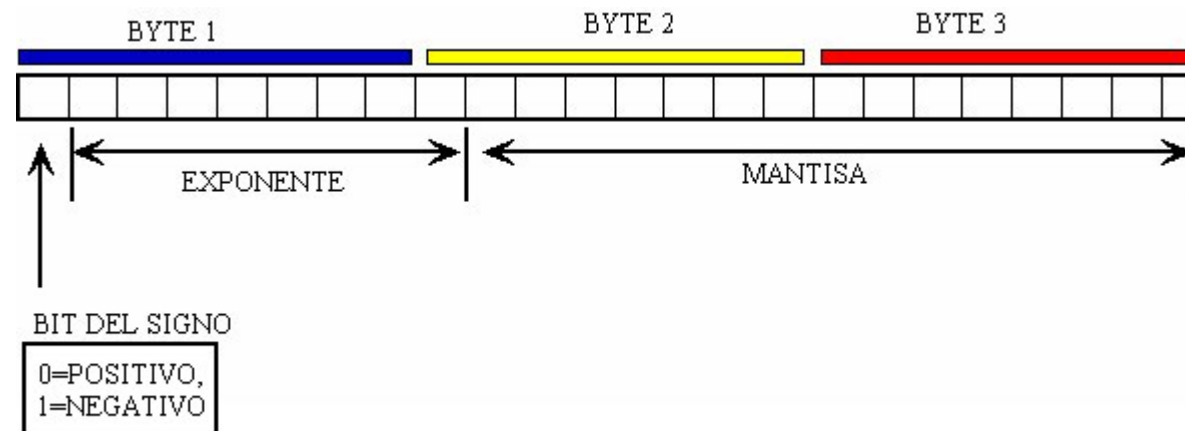
- **Número real: parte entera y parte decimal 3,1416**

La precisión (número de dígitos significativos) está limitada por la representación interna del ordenador normalmente utilizas 32 bit float / 64 bit double.

Pueden ser número muy grandes o muy pequeños. Se representan en mantisa y exponente (notación normalizada)

0,000002123 $\rightarrow 2,123 \times 10^{-6}$

8939844400000 $\rightarrow 8,9398444 \times 10^{12}$



Representación de caracteres

El código ASCII ha sido la representación habitual de los caracteres. Cada carácter le corresponde un número

Consultar tabla ASCII

ASCII (8 bits) un byte $2^8 = 256$ caracteres posibles

Tipos

- Alfabéticos : A,d,F,s,d..
- Numéricos : 0,1,2,3....
- Especiales : +,* (,/,,?,-
- Control : NULL,CR,CAN,SP

Problema: caracteres castellanos (á,Ü,ñ..), lenguajes no latinos (chino, árabe, hindú..) solución: Unicode → UTF-8 Caracteres de 1,2, 3 o 4 bytes.



Representación de caracteres

TABLA ASCII

DEC	HEX	OCT	CHAR	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH
0	0	000	NUL	32	20	040		64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051)	73	49	111	I	105	69	151	i
10	A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	80	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Código hexadecimal

- El código hexadecimal (base 16) es muy utilizado pues permite la rápida conversión en binario y ocupando menos espacio al visualizarse.

16 valores: **1,2,3,4,5,6,7,8,9,A,B,C,D,E,F**

- Ejemplos:**

0010 0011 $\rightarrow 23_{(16)} \rightarrow 35_{(10)}$

1000 1100 $\rightarrow 8C_{(16)} \rightarrow 140_{(10)}$

Ej.- Codificación de colores en
HTML5 (RGB)

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



Representación de datos complejos

Todo se codifica con números que se almacenan en binario:

Digitalizar → codificar en binario

Imágenes: tabla de colores RGB: los colores son números

Sonidos: Se recogen valores de la señal y se codifican en binario

Video: Secuencias de imágenes

