

Representación de la información: Unidades de información



- La unidad mínima de información, el bit:
 - 1 o 0, Verdadero o Falso, Sí o No.
 - ¿Cómo representamos información más compleja?
- Agrupando bits: el Byte.
- Múltiplos del bit y el Byte:
 - Prefijo del Sistema Internacional (decimal).
 - Prefijo Binario.



El bit

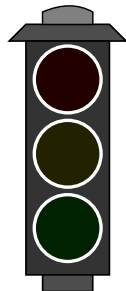
- La mínima unidad de información.
- La respuesta a la pregunta más simple posible.
- Dos valores posibles:
 - Verdadero o Falso.
 - Sí o No.
 - 0 o 1.
 - Voltaje bajo o Voltaje alto.
- Abreviado **b** (minúscula).



El bit

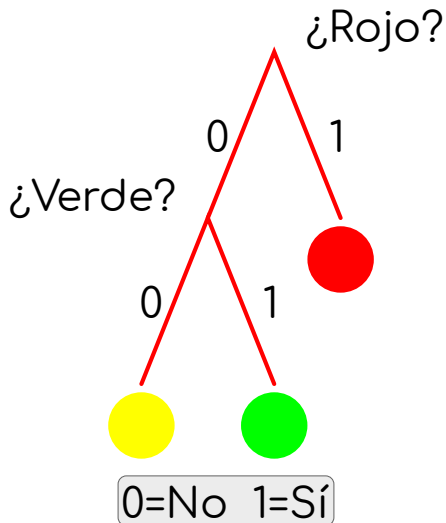
¿Cómo representamos información más compleja?

¿Cómo podríamos saber el estado de un semáforo utilizando solo preguntas *Sí/No*?



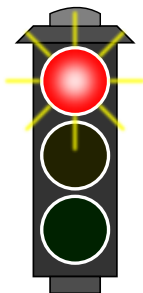
El bit

¿Cómo representamos información más compleja?

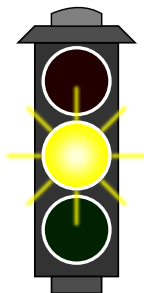


El bit

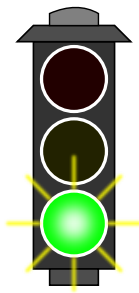
¿Cómo representamos información más compleja?



1



00



01



El bit

¿Cómo representamos información más compleja?

Alternativamente, podemos usar dos bits y asignar la correspondencia de forma arbitraria:

Codificación	Color
00	●
01	●
10	●
11	-



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo:



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b**.



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b**.
- Para representar las letras A-Z del inglés, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan:



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b**.
- Para representar las letras A-Z del inglés, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan: **6b**.



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b**.
- Para representar las letras A-Z del ingles, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan: **6b**.
- Si queremos representar los números del -100 al 100 (201 en total), necesitamos:



El bit

¿Cómo representamos información más compleja?

Para representar N estados distintos se necesitan $\lceil \log_2(N) \rceil = k$ bits.

O dicho de otra manera: *Se necesitan k bits, donde k es el entero más chico tal que $2^k \geq N$.*

Ejemplos:

- Para representar 16 colores distintos hace falta sólo: **4b**.
- Para representar las letras A-Z del ingles, solo en mayúscula, más los dígitos decimales y otros 10 símbolos auxiliares (44 símbolos en total) se necesitan: **6b**.
- Si queremos representar los números del -100 al 100 (201 en total), necesitamos: **8b**.



El bit

Representación de enteros sin signo (positivos y cero)

Si tenemos k bits y queremos representar sólo enteros positivos (y el cero) ¿Qué rango de números podemos representar?



El bit

Representación de enteros sin signo (positivos y cero)

Si tenemos k bits y queremos representar sólo enteros positivos (y el cero) ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.



El bit

Representación de enteros sin signo (positivos y cero)

Si tenemos k bits y queremos representar sólo enteros positivos (y el cero) ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.
- Como queremos incluir el cero (y queremos representar todos los enteros en el rango) el último número no puede ser 2^k , sino $(2^k) - 1$.



El bit

Representación de enteros sin signo (positivos y cero)

Si tenemos k bits y queremos representar sólo enteros positivos (y el cero) ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.
- Como queremos incluir el cero (y queremos representar todos los enteros en el rango) el último número no puede ser 2^k , sino $(2^k) - 1$.
- Por lo tanto el *Rango de Representación* de los enteros sin signo con k bits es $[0, (2^k) - 1]$.



El bit

Representación de enteros sin signo (positivos y cero)

Si tenemos k bits y queremos representar sólo enteros positivos (y el cero) ¿Qué rango de números podemos representar?

- con k bits podemos representar 2^k números distintos.
- Como queremos incluir el cero (y queremos representar todos los enteros en el rango) el último número no puede ser 2^k , sino $(2^k) - 1$.
- Por lo tanto el *Rango de Representación* de los enteros sin signo con k bits es $[0, (2^k) - 1]$.
- Para representar cada número usamos su representación binaria, completando con ceros a la izquierda para utilizar todos los *bits*.



El bit

Representación de enteros sin signo

Ejemplos:

- Con **2b** podemos representar el rango $[0, 3]$:

Dec.	Bin.	2b
0	0	00
1	1	01
2	10	10
3	11	11



Ejemplos:

- Con **2b** podemos representar el rango $[0, 3]$:

Dec.	Bin.	2b
0	0	00
1	1	01
2	10	10
3	11	11

- Con **4b** podemos representar el rango $[0, 15]$.
- Con **16b** podemos representar el rango $[0, 65535]$.
- Con **32b** podemos representar el rango $[0, 4294967295]$.



Ejemplo:

- Con **3b** podemos representar el rango $[0, 7]$:

Dec.	Bin.	3b
0	0	000
1	1	001
2	10	010
3	11	011
4	100	100
5	101	101
6	110	110
7	111	111



Para pensar:

- Si una imagen tiene una altura de $1024px$, un ancho de $720px$, y cada pixel puede tener 1 de 16 000 colores distintos ¿Cuántos bits se necesitan para representar la imagen?
- ¿Cuántos bits necesito para poder representar la edad de una persona?



El Byte

- Conjunto ordenado de **bits**: Actualmente 8**b**.
- Unidad mínima referenciable de la memoria.
- Abreviado **B** (mayúscula).



El Byte

- Conjunto ordenado de **bits**: Actualmente 8**b**.
- Unidad mínima referenciable de la memoria.
- Abreviado **B** (mayúscula).

Con un Byte se puede representar:

- Números enteros del 0 al 255, o del -128 al 127.
- 256 colores distintos.
- Caracteres de letras, números, otros símbolos y secuencias de control.



El Byte

ASCII (1963)

- 95 símbolos imprimibles y 33 caracteres de control: 128 en total.
- Desarrollado para usar sólo 7 bits.
- El bit sobrante se utilizó para crear extensiones. Actualmente se usa en el *UTF-8*.



El Byte

ASCII (1963)

Tabla *ASCII*:

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex								
0	00	NUL	16	10	DLE	32	20	48	30	0	64	40	@	80	50	P	96	60	'	112	70	p	
1	01	SOH	17	11	DC1	33	21	!	49	31	1	65	41	A	81	51	Q	97	61	a	113	71	q
2	02	STX	18	12	DC2	34	22	"	50	32	2	66	42	B	82	52	R	98	62	b	114	72	r
3	03	ETX	19	13	DC3	35	23	#	51	33	3	67	43	C	83	53	S	99	63	c	115	73	s
4	04	EOT	20	14	DC4	36	24	\$	52	34	4	68	44	D	84	54	T	100	64	d	116	74	t
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	69	45	E	85	55	U	101	65	e	117	75	u
6	06	ACK	22	16	SYN	38	26	&	54	36	6	70	46	F	86	56	V	102	66	f	118	76	v
7	07	BEL	23	17	ETB	39	27	'	55	37	7	71	47	G	87	57	W	103	67	g	119	77	w
8	08	BS	24	18	CAN	40	28	(	56	38	8	72	48	H	88	58	X	104	68	h	120	78	x
9	09	HT	25	19	EM	41	29	)	57	39	9	73	49	I	89	59	Y	105	69	i	121	79	y
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	74	4A	J	90	5A	Z	106	6A	j	122	7A	z
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	75	4B	K	91	5B	[	107	6B	k	123	7B	{
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	76	4C	L	92	5C	\	108	6C	l	124	7C	
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	77	4D	M	93	5D	]	109	6D	m	125	7D	}
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	78	4E	N	94	5E	^	110	6E	n	126	7E	~
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	79	4F	O	95	5F	_	111	6F	o	127	7F	DEL



Representando “IC 2023” en *ASCII*:

I	49	0100 1001
C	43	0100 0011
	20	0010 0000
2	32	0011 0010
0	30	0011 0000
2	32	0011 0010
3	33	0011 0011



Múltiplos del bit y el Byte

Prefijo del Sistema Internacional (decimal)

Para el Byte:

KiloByte	$1\,000^1$ Bytes	kB
MegaByte	$1\,000^2$ Bytes	MB
GigaByte	$1\,000^3$ Bytes	GB
TeraByte	$1\,000^4$ Bytes	TB
PetaByte	$1\,000^5$ Bytes	PB
ExaByte	$1\,000^6$ Bytes	EB
ZettaByte	$1\,000^7$ Bytes	ZB
YottaByte	$1\,000^8$ Bytes	YB



Múltiplos del bit y el Byte

Prefijo del Sistema Internacional (decimal)

Para el bit:

Kilobit	$1\,000^1$ bits	kb
Megabit	$1\,000^2$ bits	Mb
Gigabit	$1\,000^3$ bits	Gb
Terabit	$1\,000^4$ bits	Tb
Petabit	$1\,000^5$ bits	Pb
Exabit	$1\,000^6$ bits	Eb
Zettabit	$1\,000^7$ bits	Zb
Yottabit	$1\,000^8$ bits	Yb



Múltiplos del bit y el Byte

Prefijo Binario

KibiByte	$1\,024^1$ Bytes	kiB
MebiByte	$1\,024^2$ Bytes	MiB
GibiByte	$1\,024^3$ Bytes	GiB
TebiByte	$1\,024^4$ Bytes	TiB
PebiByte	$1\,024^5$ Bytes	PiB
ExbiByte	$1\,024^6$ Bytes	EiB
ZebiByte	$1\,024^7$ Bytes	ZiB
YobiByte	$1\,024^8$ Bytes	YiB

$$(1024 = 2^{10})$$



Múltiplos del bit y el Byte

Usos y confusiones

El prefijo binario es estándar desde el 1998, antes de eso se utilizaban los mismos nombres para referirse a cantidades múltiplo de 1 000 y 1 024. A cual de las dos se hacia referencia debía deducirse del contexto. Todavía la costumbre persiste.



Múltiplos del bit y el Byte

Usos y confusiones

El prefijo binario es estándar desde el 1998, antes de eso se utilizaban los mismos nombres para referirse a cantidades múltiplo de 1 000 y 1 024. A cual de las dos se hacia referencia debía deducirse del contexto. Todavía la costumbre persiste.

Usos comunes del prefijo decimal:

- Las velocidades de transmisión suelen medirse en múltiplos decimales del bit.
- Los tamaños de los discos duros se comunican en múltiplos decimales del Byte.



Múltiplos del bit y el Byte

Usos y confusiones

El prefijo binario es estándar desde el 1998, antes de eso se utilizaban los mismos nombres para referirse a cantidades múltiplo de 1 000 y 1 024. A cual de las dos se hacía referencia debía deducirse del contexto. Todavía la costumbre persiste.

Usos comunes del prefijo decimal:

- Las velocidades de transmisión suelen medirse en múltiplos decimales del bit.
- Los tamaños de los discos duros se comunican en múltiplos decimales del Byte.

Usos comunes del prefijo binario:

- El tamaño de las memorias principales (*RAM*).
- El tamaño de los archivos en el disco (aunque algunos sistemas utilicen la nomenclatura incorrecta).



Múltiplos del bit y el Byte

Usos y confusiones

Para pensar:

- Si compro un disco de $2TB$ ¿Cuántos TiB de datos puedo almacenar en el?
- Si mi conexión es de $1Mb/s$ ¿Cuál es la velocidad en MiB/s ?
- Si una imagen tiene una altura de $1024px$, un ancho de $720px$, y cada pixel puede tener 1 de 16 000 colores distintos ¿Cuántos KiB se necesitan para representar la imagen?



- La unidad mínima de información, el bit:
 - 1 o 0, Verdadero o Falso, Sí o No.
 - ¿Cómo representamos información más compleja?
- Agrupando bits: el Byte.
- Múltiplos del bit y el Byte:
 - Prefijo del Sistema Internacional (decimal).
 - Prefijo Binario.



¿Consultas?

