

# Llenguatge Binari i Imatge Digital

*Paula Olid Romero*  
*3r ESDAP*  
*Curs: 2024-2025*

# Relació entre un bit (binary digit) i un byte

Un bit és la unitat mínima d'informació en informàtica, i pot prendre dos valors: 0 o 1. Un byte, en canvi, és una agrupació de 8 bits. Aquesta agrupació permet representar 256 valors diferents (de 0 a 255 en decimal), suficient per representar un caràcter de text en sistemes com l'ASCII.

# Com és possible codificar text i colors amb bits?

Per codificar text, es fa servir un sistema de codificació com l'ASCII, on cada caràcter es tradueix a un valor numèric que després es converteix a binari. Per exemple, la lletra "A" té el valor 65 en ASCII, que en binari és 01000001. Per codificar colors, el model RGB assigna tres valors a cada píxel: un per al canal vermell, un altre per al verd i un per al blau. Cada valor s'expressa en un rang de 0 a 255, requerint 8 bits per canal (24 bits per píxel).

---

# Relació entre un bit (binary digit) i un byte

Un bit és la unitat mínima d'informació en informàtica, i pot prendre dos valors: 0 o 1. Un byte, en canvi, és una agrupació de 8 bits. Aquesta agrupació permet representar 256 valors diferents (de 0 a 255 en decimal), suficient per representar un caràcter de text en sistemes com l'ASCII.

# Què és el pes d'un arxiu? Quantes imatges de 2MB hi caben en un disc dur de 100GB?

El pes d'un arxiu és la quantitat d'espai que ocupa en una unitat de memòria (generalment expressat en bytes, KB, MB, etc.).

1 GB = 1.000 MB Per tant, un disc dur de 100 GB té una capacitat de 100.000 MB. Si cada imatge pesa 2 MB, podem emmagatzemar 50.000imatges

---

# Què pesa més: guardar una paraula en un arxiu de text o en un bitmap (jpg)?

## Per què?

Guardar una paraula en un arxiu de text pesa menys perquè cada caràcter es codifica amb un byte (8 bits) en ASCII. En canvi, una imatge en format bitmap conté molta més informació, ja que cada píxel requereix 24 bits (en RGB). Si la paraula es convertís a una imatge, el nombre de píxels seria significativament superior als bytes necessaris per representar la paraula en text, augmentant el pes total.

Quants caràcters es poden  
representar en total tenint  
en compte que la  
codificació ASCII extended  
fa servir 8 bits?

Amb 8 bits es poden representar  $2^8 = 256$  caràcters en total.

Per codificar només l'alfabet en minúscules (26 lletres sense accents), necessitaríem com a mínim 5 bits, ja que  $2^5 = 32$ , suficient per representar 26 caràcters.

---

Taula de  
codificació  
binària de  
l'abecedari en  
minúscules amb  
el seu valor  
numèric.

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	.
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(	01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41	)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledg	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[	01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93	]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del





# Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

[Create a presentation \(It's free\)](#)