

OS SISTEMAS DIGITAIS
E A BASE BINÁRIA

Talvez você já tenha assistido a um filme em que um náufrago ou um prisioneiro conta os dias ao agrupar, em quantidades iguais, riscos feitos em uma parede.



O processo de contagem torna-se mais prático e fácil quando agrupamos em partes iguais os elementos a serem contados. O número de elementos de cada agrupamento é a **base** da contagem. Por exemplo:

- agrupando os elementos de 5 em 5, contamos na base 5;
- agrupando os elementos de 2 em 2, contamos na base 2.

Os sistemas de numeração no dia a dia
e na informática

Um sistema de numeração é um conjunto de símbolos e regras usados na representação dos números. No dia a dia, o sistema de numeração mais empregado é o de base 10. Isso significa que, quando contamos, agrupamos as unidades de 10 em 10, de modo que 10 unidades formam uma dezena, 10 dezenas formam uma centena, 10 centenas formam 1 milhar e assim por diante.

Por exemplo, o número 271 pode ser representado pela seguinte decomposição, chamada de **desenvolvimento polinomial relativo à base 10**:

$$271 = 200 + 70 + 1$$

ou seja,

$$271 = 2 \cdot 10^2 + 7 \cdot 10^1 + 1 \cdot 10^0$$

Observe que cada dígito (algarismo) do número 271 tem um valor relativo à sua posição; por exemplo, o dígito 2 representa 2 centenas. Logo, o valor relativo do dígito 2 é 200. Observe também que cada dígito é coeficiente de uma potência natural de 10, começando em 10^0 , da direita para a esquerda.

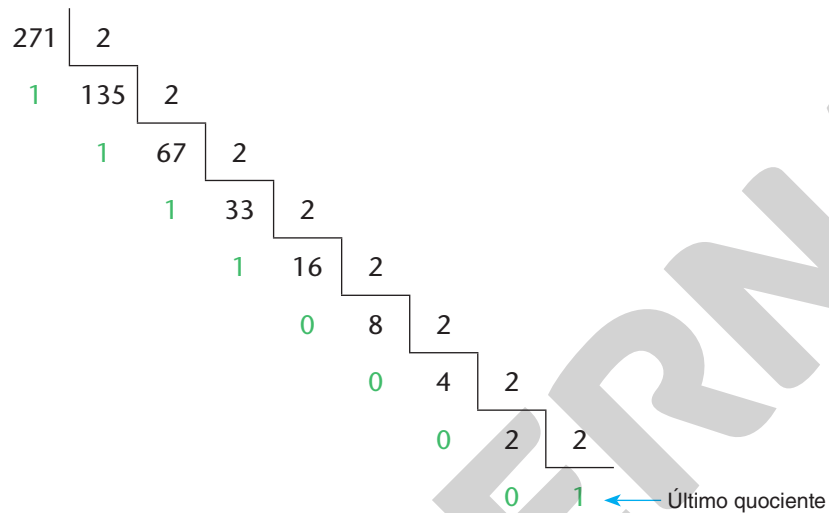
Isso nos parece óbvio, já que trabalhamos com números decimais todos os dias. A base 10 provavelmente se desenvolveu pelo fato de o ser humano ter 10 dedos nas mãos, o que facilita esse tipo de agrupamento.



Embora a base 10 seja a mais usual, nada nos impede de adotar outra base na representação de um número. Por exemplo, se quisermos representar o número 271 na base 2, chamada **base binária**, devemos encontrar os dígitos $a_n, a_{n-1}, a_{n-2}, a_{n-3}, \dots, a_1, a_0$, com $\{n, n-1, n-2, n-3, \dots, 1, 0\} \subset \mathbb{N}$, tais que:

$$271 = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + a_{n-3} \cdot 2^{n-3} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

Para a obtenção desses dígitos, basta efetuar a divisão de 271 por 2, cujo quociente deve também ser dividido por 2; o novo quociente deve ser dividido por 2, e assim por diante, até obter um quociente menor que 2. Veja a representação abaixo:



Os restos das divisões e o último quociente, na ordem inversa em que foram obtidos, isto é, 1, 0, 0, 0, 0, 1, 1, 1 e 1, são os valores de $a_n, a_{n-1}, a_{n-2}, a_{n-3}, \dots, a_1$ e a_0 , respectivamente. Assim:

$$271 = 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow \\ \Rightarrow 271 = 100001111_2$$

Esse número deve ser lido da seguinte maneira: “um, zero, zero, zero, zero, um, um, um, um na base dois”.

Note, portanto, que um número representado na base 2 apresenta apenas dígitos zeros e uns (0 e 1).

Quando a base de contagem é diferente de 10, é necessário indicá-la subscrita, à direita do último algarismo do número representado, como em 100001111_2 .

Quando a base é 10, não é necessário indicá-la; ela fica subentendida, como em 271, que significa 271_{10} .

Inversamente, como converter para a base 10 um número representado na base 2?

Dado um número representado na base 2, por exemplo, 1101010_2 , para representá-lo na base 10, basta escrever seu desenvolvimento polinomial relativo à base 2, isto é:

$$1101010_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

Assim, temos:

$$1101010_2 = 1 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 106$$

Ou seja, o número binário 1101010_2 representa o número decimal 106.

Sugestão

Para um estudo completo da Aritmética binária, sugerimos o trabalho “Sistemas numéricos e a representação interna dos dados no computador”. Disponível em: <<https://www.inf.ufsc.br/~roberto.willrich/Ensino/INE5602/restrito/ii-cap2.PDF>>. Acesso em: 8 jun. 2020.

- 1 Represente na base 2 o número decimal 165.
- 2 Represente na base 10 o número binário 110101_2 .
- 3 Efetue as operações entre números binários, sem convertê-los para a base 10.
 - a) $1011_2 + 101_2$
 - b) $1101_2 - 1011_2$

A base binária e os computadores

Os computadores utilizam a **base binária** de numeração. Todas as informações armazenadas ou processadas são representadas por códigos formados apenas por dígitos 1 ou 0, daí vem o nome **sistema digital**. Por exemplo, no código ASCII (do inglês, *American Standard Code for Information Interchange*, ou Código Padrão Norte-Americano para o Intercâmbio de Informações) a letra **A** é representada pela sequência 01000001, e o símbolo @ é representado pela sequência 01000000.

A escolha do sistema binário, com a tecnologia eletrônica atual, deve-se aos custos de produção e ao fato de que esse sistema oferece a maior velocidade no armazenamento e no processamento de dados, pois o computador deve reconhecer apenas dois impulsos elétricos distintos, representados por 0 ou 1. Se tivesse de reconhecer mais de dois impulsos, o tempo necessário seria, obviamente, maior.

A base binária e as calculadoras eletrônicas

As calculadoras eletrônicas também são computadores, porém limitados a cálculos aritméticos. Mas como elas efetuam os cálculos?

Vamos descrever o processo, resumidamente, usando o cálculo $5 + 12 = 17$ como exemplo.

Ao digitar a tecla correspondente ao número 5, por exemplo, a calculadora armazena na memória o binário de oito dígitos 00000101. Ao digitar a tecla correspondente à soma, um *microchip* identifica que deve ser efetuada a adição do número armazenado com o próximo número que será digitado. Então, ao digitar o segundo número, digamos 12, a calculadora armazena na memória o binário de oito dígitos 00001100. Assim que digitamos a tecla “igual a”, a operação é efetuada, resultando no binário de oito dígitos 00010001, que a calculadora converte para número decimal, exibindo no visor o número 17. De modo análogo são efetuadas as demais operações.



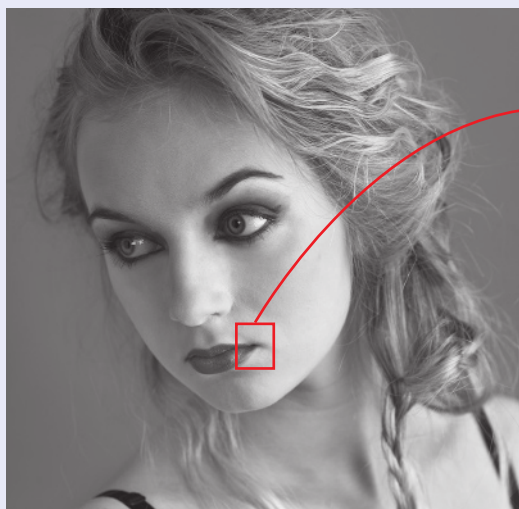
As calculadoras eletrônicas também usam o sistema binário, porém a conversão é rápida, de modo que vemos apenas os números decimais no visor.

Explorando conexões

Visão computacional: como o computador vê uma imagem

Se ampliarmos muito uma imagem vamos ver diversos “quadrinhos coloridos” (como o exemplo abaixo). Esses quadrados, chamados de *pixel*, são a menor divisão de uma imagem. Uma imagem é formada por diversos desses pontos (uma matriz de *pixels*).

A KARNAUSHENKO/SHUTTERSTOCK



A imagem mais à direita é uma ampliação de um detalhe da fotografia ao lado. Nessa ampliação, é possível ver os *pixels*, as menores divisões de uma imagem.

Observação

Os números que representam as cores devem ser naturais. O usuário os digita na base 10, e o computador os interpreta na base 2.

Computadores trabalham com números e não cores, então cada *pixel* é salvo no computador como um valor que corresponde à intensidade de cor presente nele. Normalmente a cor é salva com intervalos de 0 a 255; no caso da imagem acima, 0 é totalmente preto, e 255, totalmente branco. Concluímos que para um computador uma imagem é uma matriz de números. [...]

Cores

[No exemplo acima mostramos] uma imagem em tons de cinza. Então, como funciona com imagens coloridas?

Apesar de existirem diversas maneiras de trabalharmos com cores, nas imagens elas são representadas apenas pela combinação de 3 cores base: vermelho, verde e azul. Esse sistema é chamado de RGB (do inglês *red*, *green* e *blue*) e podemos colocar uma porcentagem de cada tonalidade para formar uma cor. Por exemplo: a cor vermelha é definida com um tom de 100% vermelho, 0% verde e 0% azul; já o amarelo combina igualmente verde e vermelho (100% vermelho, 100% verde e 0% azul). Para termos mais precisão, e consequentemente mais cores, dividimos a tonalidade entre valores de 0 a 255; assim, o vermelho seria (255,0,0) e o amarelo (255,255,0). Abaixo temos alguns exemplos de como o sistema funciona:

Nome da cor	RGB hexadecimal**	RGB decimal	Cor
Amarelo	#FFFF00	255,255,0	
Amarelo esverdeado	#99CC32	153,204,50	
Aquamarine	#70DB93	112,219,147	
Aquamarine médio	#32CD99	50,205,153	
Azul	#0000FF	0,0,255	

Fonte: Let's Code Academy. Disponível em: <<https://letscode-academy.com/blog/visao-computacional-como-o-computador-ve-uma-imagem/>>. Acesso em: 11 ago. 2020.



Sugestão

A rapidez da conversão de números hexadecimais (base 16) em números binários (base 2), e vice-versa, é o motivo pelo qual a base 16 também é usada na representação dos códigos das cores do sistema RGB. Para saber mais sobre isso, sugerimos o texto “Entenda como funciona o código de cores RGB”. Disponível em: <<https://dicasdeprogramacao.com.br/entenda-como-funcionam-os-codigos-de-cores-rgb/>>. Acesso em: 8 jun. 2020.

ATIVIDADE

Ver Manual do Professor –
Orientações específicas.

Não escreva no livro.

- 4 De acordo com as informações do texto, responda aos itens seguintes.
- Observando o diagrama da página anterior, formado pelos círculos coloridos, qual é a cor representada pelo terno ordenado (0,255,255)?
 - A cor de um *pixel* é determinada pelo terno ordenado (51,153,255). Quais percentuais de vermelho, verde e azul compõem esse *pixel*?
 - Quantas cores diferentes pode apresentar cada *pixel*?

O que são *bits* e *bytes*?

Provavelmente você já ouviu falar sobre *bits* e *bytes*. A capacidade da memória RAM (*Random Access Memory*, em português Memória de Acesso Aleatório) e do disco rígido, o tamanho dos arquivos e a capacidade de processamento de um computador são expressos em *bits* e *bytes*.

Por exemplo, um computador pode ter um processador de 64 *bits* com 8 *gigabytes* de memória RAM e 1 *terabyte* de espaço no disco rígido. Mas o que significam essas unidades?

Bit

Como vimos, os computadores operam utilizando o sistema numérico binário, isto é, de base 2. Assim, processam e armazenam informações por meio de sequências formadas apenas por zeros e uns (0 e 1). Cada um desses dígitos, que representa um impulso elétrico reconhecido pelo computador, é chamado de *bit*, abreviação da expressão *binary digit* (dígito binário).

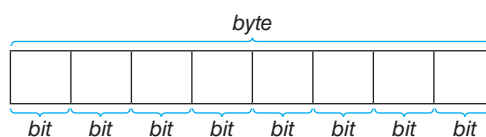


Os computadores usam códigos binários compostos por zeros e uns, sendo que cada dígito corresponde a um *bit*.

Byte

O *byte* é uma unidade de informação digital formada por uma sequência de 8 *bits*. Portanto, um *byte* é representado por um número binário, por exemplo, 10100110. A palavra *byte* é a abreviação de *binary term*.

E quantos *bytes* diferentes podem ser formados? A resposta a essa pergunta pode ser obtida por meio do princípio fundamental da contagem, a partir do seguinte diagrama:



Como cada *bit* pode assumir apenas um dos valores 0 ou 1, temos:

Número de possibilidades → $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^8 = 256$

Concluimos, então, que podem ser formados 256 *bytes* diferentes.

Por que cada *byte* é formado por 8 *bits*?

O *byte* de 8 *bits* é algo que os pesquisadores estabeleceram através de tentativas durante os últimos 50 anos.

Com 8 *bits* em cada *byte* é possível representar 256 números binários, cuja correspondência com os números decimais é mostrada abaixo:

Números decimais	Números binários
0	00000000
1	00000001
2	00000010
.	.
.	.
.	.
166	10100110
.	.
.	.
.	.
255	11111111

Observação

Sinais de controle são comandos não imprimíveis, como a função das teclas *Home*, que leva o cursor para o início da linha, e *End*, que leva o cursor para o final, em um editor de textos, por exemplo.

Como o número de caracteres (letras maiúsculas e minúsculas, sinais de pontuação ou acentuação etc.) e **sinais de controle** (não imprimíveis) enviado para o computador, por meio do teclado, não excede 256, e cada *byte* pode representar um único caractere ou sinal de controle, concluiu-se que 8 *bits* em cada *byte* seriam suficientes.

Múltiplos do *byte*

Indicamos os múltiplos do *byte* acrescentando prefixos, conforme a tabela:

Múltiplos do <i>byte</i>					
Prefixo binário (IEC)			Prefixo do SI		
Nome	Símbolo	Múltiplo	Nome	Símbolo	Múltiplo
<i>byte</i>	B	2^0	<i>byte</i>	B	10^0
<i>kibibyte</i>	kiB	2^{10}	<i>kilobyte</i>	kB	10^3
<i>mebibyte</i>	MiB	2^{20}	<i>megabyte</i>	MB	10^6
<i>gibibyte</i>	GiB	2^{30}	<i>gigabyte</i>	GB	10^9
<i>tebibyte</i>	TiB	2^{40}	<i>terabyte</i>	TB	10^{12}
<i>pebibyte</i>	PiB	2^{50}	<i>petabyte</i>	PB	10^{15}
<i>exbibyte</i>	EiB	2^{60}	<i>exabyte</i>	EB	10^{18}
<i>zebibyte</i>	ZiB	2^{70}	<i>zettabyte</i>	ZB	10^{21}
<i>yobibyte</i>	YiB	2^{80}	<i>yottabyte</i>	YB	10^{24}

Um esclarecimento necessário

Até algum tempo atrás, o *quilo*byte, o *mega*byte, o *giga*byte, o *tera*byte etc. eram representados por potências de 2 (2^{10} , 2^{20} , 2^{30} , 2^{40} etc.). No entanto, os fabricantes de mídias digitais passaram a expressar essas unidades em potência de 10 (10^3 , 10^6 , 10^9 , 10^{12} etc.), cujos valores são aproximados das potências de 2 correspondentes, por exemplo $2^{10} \approx 10^3$.

Para padronizar a nomenclatura, a IEC (*International Electrotechnical Commission*, ou Comissão Eletrotécnica Internacional) estabeleceu, por volta de 1998, que as unidades *quilo*byte (kB), *mega*byte (MB), *giga*byte (GB) etc. devem seguir a definição do Sistema Internacional de Unidades (SI) sendo representadas por potências de 10 (10^3 , 10^6 , 10^9 etc.), pois nesse sistema os prefixos quilo, mega, giga etc. significam 1.000, 1.000.000, 1.000.000.000 etc., respectivamente.

Para as unidades de armazenamento digital representadas por potências de 2, a IEC criou os termos *kibi*byte, *mebi*byte, *gibi*byte etc., representadas pelas potências 2^{10} , 2^{20} , 2^{30} etc., respectivamente. Os símbolos que representam essas novas unidades derivam de kB, MB, GB etc., com a letra i entre as duas letras, ou seja, kiB, MiB, GiB, TiB etc.

ATIVIDADES

Ver Manual do Professor –
Orientações específicas.

Não escreva no livro.

- 5 (IMT-SP) Fotografias coloridas são armazenadas em formato digital como matrizes de pontos (*pixels*), cada qual composto por três *bytes*. Sequências de 24 fotografias estáticas por segundo podem constituir um filme. Quantos *bytes* serão necessários para armazenar, em um computador, um filme de 30 segundos de duração, se cada fotografia apresentar resolução de 200 por 200 pontos?



INDEPENDENT BIRDS/SHUTTERSTOCK

- 6 No exercício anterior, quantos *megabytes* serão necessários para armazenar o filme no computador?

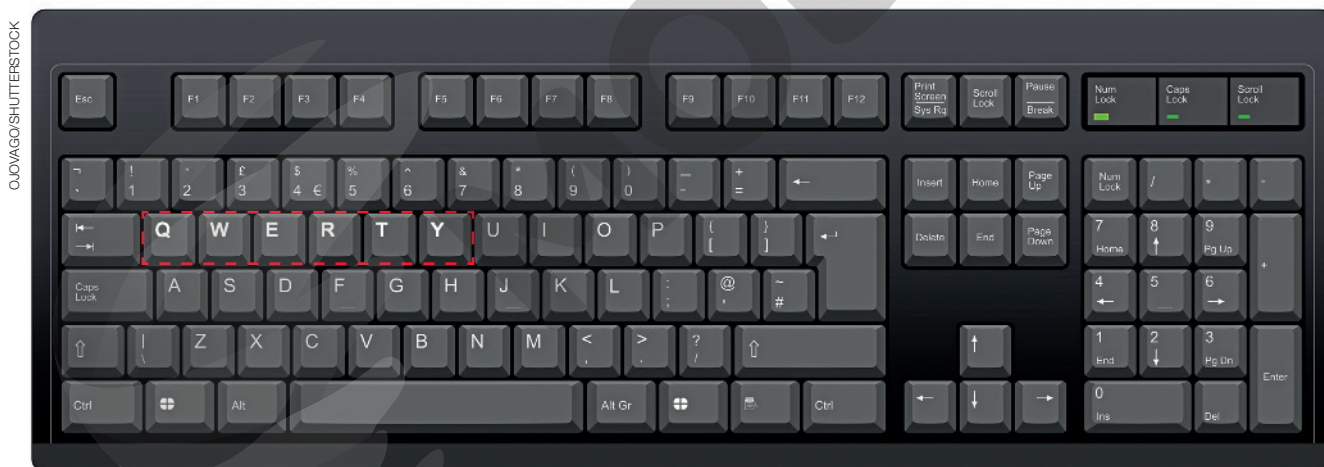
Bytes e o padrão ASCII

Os *bytes* podem representar caracteres individuais (letras não acentuadas, pontuação, acentos etc.) ou operações de controle (não imprimíveis) em um documento de texto. No **sistema padrão de caracteres ASCII** cada um desses *bytes* é representado por um número binário correspondente a um dos números decimais de 0 a 127 e está associado a um caractere individual ou a um código de controle de texto. Dentre os demais valores binários, correspondentes aos decimais de 128 a 255, estão os representantes de elementos especiais, como caracteres acentuados de diversas línguas, como o francês.



Variante francesa do teclado AZERTY, nome que vem da sequência das seis primeiras letras à direita da tecla *Tab* (A Z E R T Y).

No Brasil, o teclado adotado é o QWERTY, que obedece às orientações da Associação Brasileira de Normas Técnicas (ABNT). O nome QWERTY vem da sequência das seis primeiras letras à direita da tecla *Tab*, como destacado abaixo.



Teclado QWERTY, padrão adotado no Brasil.

Os 128 primeiros códigos padrão da tabela ASCII são apresentados a seguir. Os computadores armazenam documentos de texto utilizando esses códigos.

Tabela ASCII padrão

Os 32 primeiros códigos (00000000 a 00011111, em binário) e o 127º (01111111, em binário) são códigos de controle como ir para o começo do texto (STX) e entrar (CR), os demais são códigos de escrita.

TABELA ASCII PADRÃO

Decimal	Binário	Referência	Decimal	Binário	Referência	Decimal	Binário	Referência
0	00000000	Null – NUL	43	00101011	+	86	01010110	V
1	00000001	Start of Heading – SOH	44	00101100	,	87	01010111	W
2	00000010	Start of Text – STX	45	00101101	–	88	01011000	X
3	00000011	End of Text – ETX	46	00101110	.	89	01011001	Y
4	00000100	End of Transmission – EOT	47	00101111	/	90	01011010	Z
5	00000101	Enquiry – ENQ	48	00110000	0	91	01011011	[
6	00000110	Acknowledge – ACK	49	00110001	1	92	01011100	\
7	00000111	Bell, rings terminal bell – BEL	50	00110010	2	93	01011101]
8	00001000	BackSpace – BS	51	00110011	3	94	01011110	^
9	00001001	Horizontal Tab – HT	52	00110100	4	95	01011111	_
10	00001010	Line Feed – LF	53	00110101	5	96	01100000	`
11	00001011	Vertical Tab – VT	54	00110110	6	97	01100001	a
12	00001100	Form Feed – FF	55	00110111	7	98	01100010	b
13	00001101	Enter – CR	56	00111000	8	99	01100011	c
14	00001110	Shift-Out – SO	57	00111001	9	100	01100100	d
15	00001111	Shift-In – SI	58	00111010	:	101	01100101	e
16	00010000	Data Link Escape – DLE	59	00111011	;	102	01100110	f
17	00010001	Device Control 1 – D1	60	00111100	<	103	01100111	g
18	00010010	Device Control 2 – D2	61	00111101	=	104	01101000	h
19	00010011	Device Control 3 – D3	62	00111110	>	105	01101001	i
20	00010100	Device Control 4 – D4	63	00111111	?	106	01101010	j
21	00010101	Negative Acknowledge – NAK	64	01000000	@	107	01101011	k
22	00010110	Synchronous idle – SYN	65	01000001	A	108	01101100	l
23	00010111	End Transmission Block – ETB	66	01000010	B	109	01101101	m
24	00011000	Cancel line – CAN	67	01000011	C	110	01101110	n
25	00011001	End of Medium – EM	68	01000100	D	111	01101111	o
26	00011010	Substitute – SUB	69	01000101	E	112	01110000	p
27	00011011	Escape – ESC	70	01000110	F	113	01110001	q
28	00011100	File Separator – FS	71	01000111	G	114	01110010	r
29	00011101	Group Separator – GS	72	01001000	H	115	01110011	s
30	00011110	Record Separator – RS	73	01001001	I	116	01110100	t
31	00011111	Unit Separator – US	74	01001010	J	117	01110101	u
32	00100000	Space – SPC	75	01001011	K	118	01110110	v
33	00100001	!	76	01001100	L	119	01110111	w
34	00100010	"	77	01001101	M	120	01111000	x
35	00100011	#	78	01001110	N	121	01111001	y
36	00100100	\$	79	01001111	O	122	01111010	z
37	00100101	%	80	01010000	P	123	01111011	{
38	00100110	&	81	01010001	Q	124	01111100	
39	00100111	'	82	01010010	R	125	01111101	}
40	00101000	(83	01010011	S	126	01111110	~
41	00101001)	84	01010100	T	127	01111111	Delete
42	00101010	*	85	01010101	U			

Fonte: Tec Ciência (UFBA). Disponível em: <<http://teccienciapiloto.ufba.br/numeros-binarios/curiosidades/ascii-completa.png>>. Acesso em: 12 jun. 2020.

Observe, por exemplo, que o número binário 01010010 representa a letra **R** (maiúscula), e 01110010 representa a letra **r** (minúscula).

Sugestões

Apresentamos a seguir duas sugestões de vídeos sobre a base binária:

O *hit dos bits*. Disponível em: <<https://m3.ime.unicamp.br/recursos/1116>>. Acesso em: 8 jun. 2020.

O *mágico das Arábias*. Disponível em: <<https://m3.ime.unicamp.br/recursos/1131>>. Acesso em: 8 jun. 2020.

Professor, se possível, reproduza os vídeos em sala e promova uma discussão com toda a turma.

- 7 Observe a Tabela ASCII estendida, apresentada a seguir, que complementa a reproduzida anteriormente, e responda aos itens seguintes.

Nota: Dependendo do conjunto de caracteres adotado pelo sistema operacional de seu computador, os códigos ASCII estendidos podem não coincidir com os desta tabela. O conjunto de caracteres mais usual é conhecido como ISO 8859-1 ou ISO Latin 1, cujos caracteres são utilizados na maioria dos idiomas ocidentais, incluindo a língua portuguesa.

TABELA ASCII ESTENDIDA					
Decimal	Referência	Decimal	Referência	Decimal	Referência
128	Ç	171	½	214	í
129	ü	172	¼	215	î
130	é	173	¡	216	ï
131	â	174	«	217	
132	ä	175	»	218	
133	à	176		219	
134	á	177		220	
135	ç	178		221	
136	ê	179		222	
137	ë	180		223	
138	è	181	Á	224	
139	ï	182	Â	225	
140	î	183	Ã	226	
141	ì	184	©	227	
142	Ä	185		228	
143	Å	186		229	
144	É	187		230	
145	æ	188		231	
146	Æ	189		232	
147	ó	190	¥	233	
148	ö	191		234	¡
149	ò	192		235	¢
150	û	193		236	£
151	ù	194		237	¤
152	ÿ	195		238	¥
153	Ö	196		239	¦
154	Ü	197		240	§
155	ø	198		241	¨
156	£	199		242	©
157	Ø	200		243	ª
158	×	201		244	«
159	ƒ	202		245	¬
160	á	203		246	­
161	à	204		247	®
162	ó	205		248	¯
163	ú	206		249	°
164	ñ	207		250	±
165	Ñ	208		251	²
166	ª	209		252	³
167	º	210		253	´
168	¿	211		254	
169	®	212		255	
170	¬	213			

Fonte: Tec Ciência (UFBA). Disponível em: <<http://teccienciapiloto.ufba.br/numeros-binarios/curiosidades/ascii-completa.png>>. Acesso em: 12 jun. 2020.

- a) O caractere ä corresponde ao número decimal 132. Qual é o *byte* que representa esse caractere?
- b) Qual é o caractere representado pelo *byte* 11000001?

Explorando conexões

O que é CGI e computação gráfica?

CGI é uma sigla em inglês para o termo *Computer Graphic Imagery*, ou seja, imagens geradas por computador, a famosa computação gráfica. O termo se refere a todas as imagens geradas através de computadores feitas em três dimensões, com a profundidade de campo sendo possível graças apenas à computação. Parece uma conversa técnica, certo? Nem tanto. O CGI nada mais é que praticamente todo tipo de efeito ou animações que vemos hoje em dia, seja em filmes, *videogames* ou mesmo na televisão. [...]

Um pouquinho de história...

[...] Em 1962, o Dr. Ivan Sutherland, um dos pioneiros da Internet e realidade aumentada, lançou a tese “Sketchpad – A Man-Machine Graphical Communication System”, que se tornou bastante influente e despertou a curiosidade de empresas dos ramos automobilístico e aeroespacial. No final da década, o uso de computadores nestas empresas fez surgir a tecnologia **CAD**, ramo da computação gráfica voltada ao auxílio na criação de desenhos e estruturas, que se tornou um padrão até os dias de hoje. [...]

Observação

A sigla CAD (*Computer Aided Design*) é o nome genérico de *softwares* utilizados pela Engenharia, Geologia, Geografia, Arquitetura e *design* para facilitar a elaboração de projetos e desenhos técnicos.

Tecnologia + Arte = Entretenimento

Posteriormente, o grande avanço no poder de processamento gráfico dos computadores e a criação de mais e mais ferramentas que facilitam o uso de CGI permitiram que ela fosse usada em muitas mais áreas de aplicação. Um dos maiores exemplos é o uso na história do cinema. Hoje, mesmo uma cena simples pode ter bons recursos de CGI envolvidos em sua criação, pois além de tornar desnecessárias as gravações em locações externas, permite a elaboração de cenas que de outro modo seriam impossíveis.

Um dos primeiros filmes a usar essa técnica foi *Star Wars – Uma Nova Esperança*, de 1977. Os filmes da épica saga de George Lucas acabariam por ser referência no uso de CGI para criação de efeitos especiais, em grande parte por causa de sua empresa, a Industrial Light and Magic, uma subdivisão de sua produtora para efeitos especiais e responsável pelo desenvolvimento de novas câmeras e modos de filmagem. [...]

Em 1982, os estúdios Disney inovaram e marcaram a história da CGI cinematográfica com o filme *TRON*. O filme, que conta a história do *hacker* Kevin Flynn lutando num mundo inteiramente digital, foi o primeiro filme a usar a CGI em larga escala. [...]

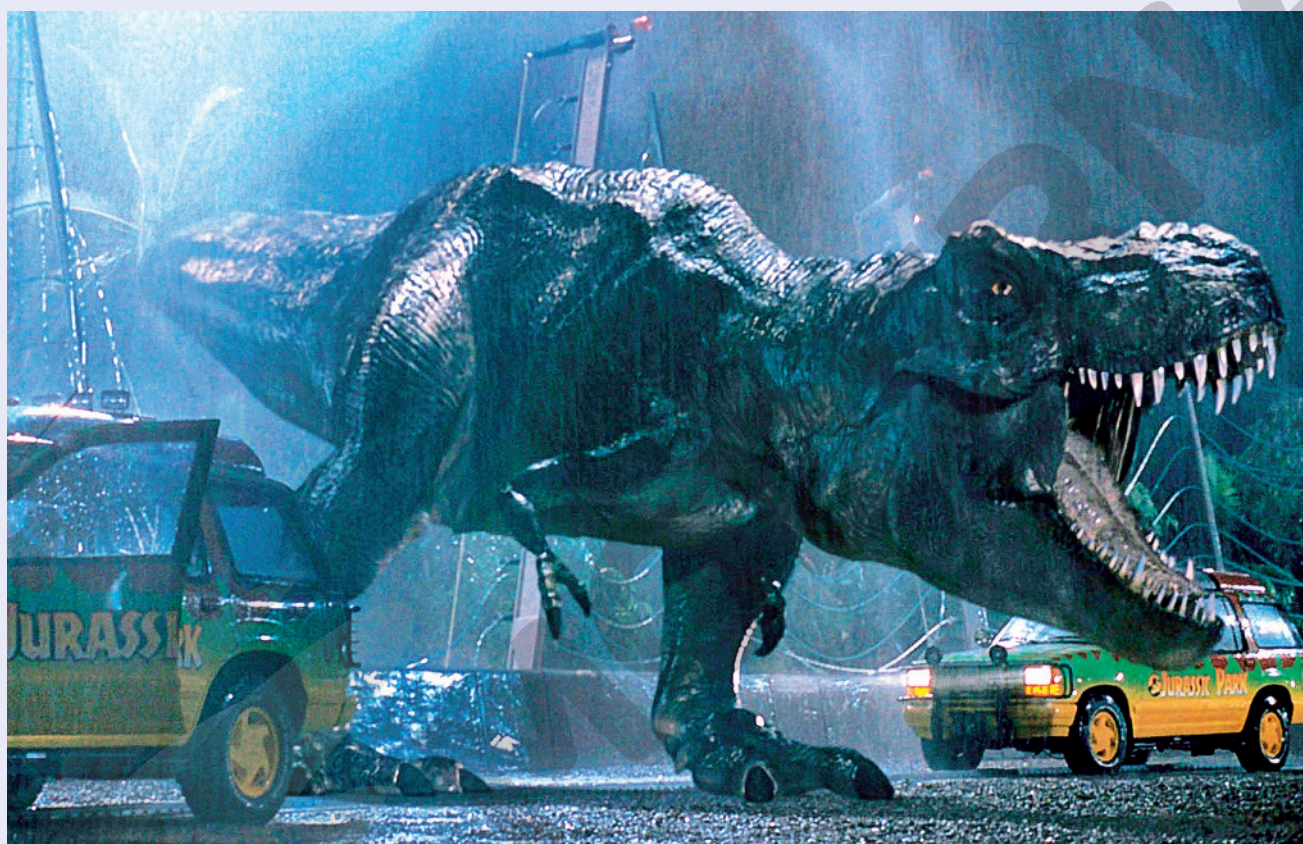
O próximo passo da revolução viria em 1985 com o lançamento de *O Enigma da Pirâmide*. O filme, de Barry Levinson e produção de Steven Spielberg, levou às telas o primeiro personagem criado e animado digitalmente: o Cavaleiro de Vitral. [...]

O domínio da técnica e uma nova indústria

Claro que Hollywood acabou se tornando a principal investidora do segmento. A própria ILM, que havia feito *Star Wars*, nos brindou com verdadeiros ícones do CGI logo no começo da década de 90, fosse o robô T-1000 de *Exterminador do Futuro II*, que combinava cenas reais com o androide de metal líquido, até os dinossauros de *Jurassic Park*, que novamente quebrou barreiras ao não só criar um personagem imaginário em cena, mas recriar um ser vivo, de ossos, músculos e movimentos, camada por camada, totalmente feito pelo computador. Esse processo de modelagem acabou se tornando padrão para os efeitos cada vez mais realistas que víamos no cinema, desenvolvidos agora à semelhança da vida real.

[...]

Fonte: O que é CGI e computação gráfica? Canaltech. Disponível em: <<https://canaltech.com.br/software/O-que-e-CGI-e-computacao-grafica/>>. Acesso em: 11 ago. 2020.



UNIVERSAL PICTURES/PHOTO 12/GLOW IMAGES

Reprodução proibida. Art. 184 do Código Penal e Lei 9.610 de 19 de fevereiro de 1998.

Cena do filme *Jurassic Park*. (Direção de Steven Spielberg. Estados Unidos: Amblin Entertainment, 1993). A criação envolveu um processo de modelagem feito no computador para obter os efeitos realistas.

ATIVIDADE COMPLEMENTAR

Ver Manual do Professor –
Orientações específicas.

Não escreva no livro.

8 Alguns dos fundamentos matemáticos da computação gráfica são as transformações geométricas. Façam uma pesquisa na internet ou em bibliotecas atendendo aos dois itens seguintes:

- Escrevam um breve texto sobre transformações geométricas, com ilustrações, feitas à mão ou no computador, que complementem a pesquisa.
- Escrevam um breve texto sobre aplicações das transformações geométricas na computação gráfica, ilustrado com exemplos.