

# INTRODUÇÃO AO PROCESSAMENTO DE DADOS

*Tiago Oliveira*  
*Instituto de Matemática e Estatística*  
*Departamento de Ciência da Computação*  
*Universidade Federal da Bahia*



# Unidades de Medidas Computacionais

## QUANTIFICAÇÃO DE DADOS



# Dados e Informações

Como nós representamos?

- Texto: Nome ou Endereço?
  - Letras de Alfabeto + Símbolos e Pontuação;
- Número: Valor ou Quantidade?
  - Numeração Decimal;
- Pixel: Cor?
  - Conjunto de cores visíveis;
- Som: Frequência?
  - 20Hz ~ 20.000Hz;

# Dados e Informações

Como o computador representa o menor dado?

- Bit.
- O que é um Bit?
  - É o menor dado ou informação existente no computador (binary digit, ou dígito binário).
- Como representar um Bit?
  - O Bit pode possuir apenas 1 entre 2 possíveis estados, algumas formas:
- 0 e 1;
- Verdadeiro e Falso;
- Ligado e Desligado;

# Bits

No sistema decimal, trabalhamos com números de 0 a 9;

- Para representar números maiores que 9, passamos a agrupar dígitos decimais: 10, 11, 12

- No sistema binário trabalhamos com números de 0 a 1;

- Para representar números maiores que 1, passamos a agrupar dígitos binários: 10, 11, 100

# Byte

Por ser uma máquina e para obter velocidades maiores, um computador costuma agrupar bits de 8 em 8 bits. Cada agrupamento de 8 bits é chamado de byte.

Os bytes são as unidades de medidas mais utilizadas na computação, servem para referenciar tamanho de arquivos ou espaço em memória.

# Conjuntos de Bytes

Em nosso dia-a-dia utilizamos diversas abreviações, principalmente com números, para encurtar a pronúncia ou escrita:

- 1000 gramas: 1 Kg
- 100000 metros: 100 Km
- Na computação também possuímos tal abreviação, mas esta é feita de modo particular.

# Conjuntos de Bytes

Como a base numérica humana é decimal, costumamos dividir os números de 10 em 10.

– Ex:  $1\text{Km} = 1\text{m} \times 10^3$ , logo = 1000m

• Como a base numérica computacional é binária, na computação costumamos dividir os números na base 2.

– Ex:  $1\text{KB} = 1\text{B} \times 2^{10}$ , logo = 1024 B



# Conjuntos de Bytes

## Medidas (Múltiplos)

A base “2” surge devido ao sistema binário

8 bits	$(2^0)$	1 byte (B)
1.024 bytes	$(2^{10})$	1 kilobyte (KB)
1.024 kilobytes	$(2^{20})$	1 megabyte (MB)
1.024 megabytes	$(2^{30})$	1 gigabyte (GB)
1.024 gigabytes	$(2^{40})$	1 terabyte (TB)
1.024 terabytes	$(2^{50})$	1 petabyte (PB)
1.024 petabytes	$(2^{60})$	1 exabyte (EB)
1.024 exabytes	$(2^{70})$	1 zettabyte (ZB)
1.024 zettabytes	$(2^{80})$	1 yottabyte (YB)

# Conjuntos de Bytes

Vendedores de Discos Rígidos e alguns outros dispositivos de armazenamento tratam cada 1.000 bytes como 1KB, ao invés de 1024.

- O tamanho da letra “B” diferencia entre bit e Byte, lembre que a razão entre eles é 8.
  - Dispositivos de comunicação geralmente informam velocidade em bits, e não bytes.

# REPRESENTAÇÃO DE DADOS



# Bits

Uma solução: o uso de dispositivos eletrônicos baseados na tecnologia dos semicondutores, como os transistores.

O transistor: é um dispositivo usado para controlar o fluxo de corrente. Ele tem duas características importantes:

1- é capaz de amplificar um sinal elétrico.

2- é capaz de chavear (comutar) entre ligado e desligado (ou fechado e aberto), deixando corrente passar através dele ou bloqueando-a.

# Representação de Dados

É possível utilizar os bytes para representar qualquer tipo de dado;

- Para isso, geralmente existe algum meio de transformar um byte na representação adequada;
- Essa transformação pode ser feita através de tabelas ou equações matemáticas.

# Representando Texto

A representação mais comum em um computador, depois dos números, é a do texto;

- Os textos são estabelecidos como a união de diversos caracteres;
- Caracteres são, em geral, traduzidos por:
  - Tabela ASCII; ou
  - Padrão Unicode.

# A Tabela ASCII

É um modelo antigo e um pouco defasado, mas ainda utilizado em alguns sistemas;

Sua defasagem está na ausência de representação para letras de escritas em algumas outras línguas, principalmente orientais;

Cada letra, pontuação ou símbolo, é representado por um conjunto de 8 bits, ou seja, 1 Byte.

# A Tabela ASCII

<b>Binário</b>	<b>Decimal</b>	<b>Glifo</b>
0100 0001	65	A
0100 0010	66	B
0110 0001	97	a
0110 0010	98	b
0011 0001	49	1
0011 0010	50	2



# Padrão Unicode

Permite ao computador representar texto em qualquer sistema de escrita existente;

- Mais de 107 mil caracteres comportados;
- É composto por um conjunto de diagramas de códigos e metodologias de codificação;
- É desenvolvido e mantido por um consórcio sem fins lucrativos chamado Unicode Consortium que é mantido por diversas universidades e grandes empresas;

# Padrão Unicode

Este padrão surgiu principalmente com base na necessidade de troca mundial de informações;

- Impulsionado pela globalização e pela internet;
- Também pode ser representado por tabelas mais complexas: [Tabela](#)

# Padrão Unicode

Faça esse experimento:

Abra o Bloco de Notas e insira a frase: INTRODUÇÃO AO PROCESSAMENTO DE DADOS!

Salve o arquivo no disco com o nome de MAT115.txt

Utilize o Explorer e veja o tamanho do arquivo.

Você irá descobrir que o arquivo ocupa um espaço de \_\_\_\_ bytes, \_\_ byte(s) para cada caractere.

Adicione seu nome ao final da sentença e salve novamente, o tamanho do arquivo irá subir para o número referente de \_\_\_\_ bytes.

# CONVERSÃO ENTRE BASES NUMÉRICAS

# Conversão entre Bases

Binária:

- 1010 (2)

- Decimal:

- 10 (10)

- Octal:

- 12 (8)

- Hexadecimal:

- A (16)

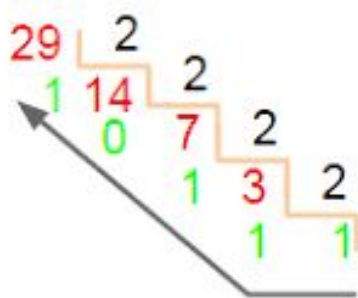
# Conversão entre Bases

Como só existem dois números no sistema binário ( 0 e 1) temos a seguinte correspondência:

0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

# Conversão entre Bases

A conversão de números do sistema decimal para outro sistema de numeração processa-se através de operações de divisão.



29 Decimal = 11101 Binário

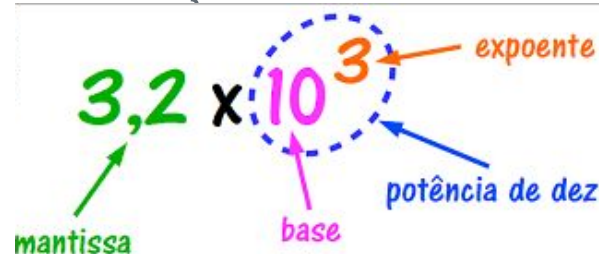
# Conversão entre Bases

A conversão de números do sistema binário para decimal é feito através de multiplicações.

0	0	1	0	0	1	1	0
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
$0*128$	$0*64$	$1*32$	$0*16$	$0*8$	$1*4$	$1*2$	$0*1$

$$38_{(10)} = 00100110_{(2)}$$

Notação científica





# Números binários negativos

Os computadores lidam com números positivos e números negativos, sendo necessário encontrar uma representação para números com sinal **negativo**. Existe uma grande variedade de opções, das quais nesta seção serão apresentadas apenas três para representar valores negativos:

- sinal e amplitude/magnitude (S+M)

- complemento de 1

- complemento de 2

# Sinal magnitude

Como o próprio nome indica, a representação sinal e magnitude utiliza um bit para representar o sinal, o bit mais à esquerda: 0 para indicar um valor positivo, 1 para indicar um valor negativo.

$$+10_{10} = \mathbf{0}1010_2 \quad -10_{10} = \mathbf{1}1010_2$$

# Complemento de 1

Na representação em complemento de 1 inverte-se todos os bits de um número para representar o seu complementar: assim, se converte um valor positivo para um negativo, e vice-versa. Quando o bit mais à esquerda é 0, esse valor é positivo; se for 1, então é negativo.

$$100_{10} = 01100100_2 \text{ (com 8 bits)}$$

# Complemento de 1

O problema desta representação é que existem 2 padrões de bits para o 0, havendo assim desperdício de representação:

$$0_{10} = 00000000_2 = 11111111_2$$

# Complemento de 2

A solução encontrada consiste em representar os números em complemento de 2. Para determinar o negativo de um número, inverte-se todos os seus bits e soma-se uma unidade.

$$101_{10} = 01100101_2 \text{ (com 8 bits)}$$

$$10011010_2$$

$$10011010_2 + 1 = 10011011_2 = -101_{10}$$

# Complemento de 2

A representação em complemento para 2 tem as seguintes características:

- O bit da esquerda indica o sinal;

- Possui processo para converter um número de positivo para negativo e de negativo para positivo;

- O 0 tem uma representação única: todos os bits a 0;

- A gama de valores que é possível representar com  $n$  bits é  $-2^{n-1} \dots 2^{n-1}-1$ .

# Complemento de 2

Qual o número representado por  $11100100_2$  (com 8 bits)? Como o bit da esquerda é 1 este número é negativo. Invertendo todos os bits -> Somando uma unidade:

$$00011011_2$$

$$00011011_2 + 1 = 00011100_2 = 28_{10}$$

$$11100100_2 = -28_{10}$$

# Conversão entre Bases

- (a) Converta  $99_{(10)}$  em binário
- (b) Converta  $325_{(10)}$  em binário
- (c) Converta  $7858_{(10)}$  em binário
- (d) Converta  $28591_{(10)}$  em binário
- (e) Converta  $101010_{(2)}$  em decimal
- (f) Converta  $11001100_{(2)}$  em decimal
- (g) Converta  $111011010001_{(2)}$  em decimal
- (h) Converta  $1000000000000000_{(2)}$  em decimal
- (i) Converta sua idade para binário.



# OPERAÇÕES LÓGICAS

# Operações Lógicas com Binários

Operações lógicas para realizar com números binários;

- Estas operações estão relacionadas à lógica proposicional;
- Por definição, assuma:
  - Dígitos binários 0 = Falso;
  - Dígitos binários 1 = Verdadeiro;

# Operações Lógicas com Binários

As Principais Operações lógicas são:

Português	Inglês
Não	NOT
E	AND
OU	OR
OU exclusivo	XOR

# Operações Lógicas com Binários

Numa operação matemática, recebemos dois valores e geramos um terceiro valor. Exemplo:

$$2 + 3 = 5$$

- Não é muito diferente na operação lógica, contudo, não iremos fazer operações básicas aritméticas, mas sim lógicas. Exemplo:

$$10101 \text{ ou } 01010 = 11111$$

# Operações Lógicas com Binários

Operador lógico **NÃO**.

Inverte o valor lógico de um único dígito:

- Verdadeiro passa a ser Falso ( 1 -> 0)
- Falso passa a ser Verdadeiro ( 0 -> 1)

# Operações Lógicas com Binários

Operador lógico **E**.

Escrita:  $a \text{ E } b$

• Tabela Verdade:

<b>a</b>	<b>b</b>	<b><math>a \text{ E } b</math></b>
1	1	1
1	0	0
0	1	0
0	0	0

# Operações Lógicas com Binários

$$1. 1101_{(2)} \quad E \quad 101_{(2)}$$

$$2. 111_{(2)} \quad E \quad 7_{(10)}$$

$$3. 50_{(10)} \quad E \quad 100_{(10)}$$

$$4. 32_{(10)} \quad E \quad 63_{(10)}$$

$$5. 7_{(10)} \quad E \quad 15_{(10)}$$

# Operações Lógicas com Binários

Operador lógico **OU**.

Escrita:  $a \text{ OU } b$

• Tabela Verdade:

<b>a</b>	<b>b</b>	<b>a OU b</b>
1	1	1
1	0	1
0	1	1
0	0	0



# Operações Lógicas com Binários

- |                 |    |              |
|-----------------|----|--------------|
| 1. $101_{(2)}$  | OU | $101_{(2)}$  |
| 2. $111_{(2)}$  | OU | $8_{(10)}$   |
| 3. $40_{(10)}$  | OU | $101_{(10)}$ |
| 4. $31_{(10)}$  | OU | $33_{(10)}$  |
| 5. $333_{(10)}$ | OU | $555_{(10)}$ |

# Operações Lógicas com Binários

Operador lógico **XOU**.

Escrita:  $a \text{ XOU } b$

• Tabela Verdade:

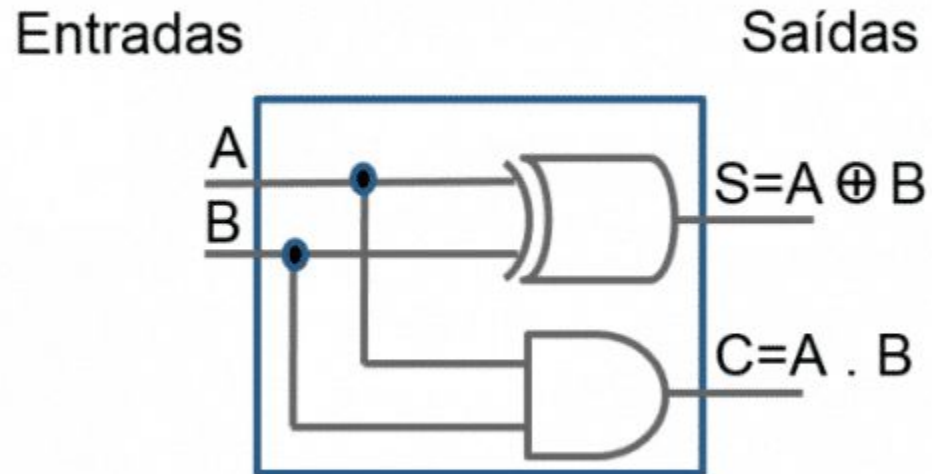
<b>a</b>	<b>b</b>	<b>a XOU b</b>
1	1	0
1	0	1
0	1	1
0	0	0

# Operações Lógicas com Binários

1. $11101_{(2)}$	XOU	$101_{(2)}$
2. $111_{(2)}$	XOU	$9_{(10)}$
3. $60_{(10)}$	XOU	$121_{(10)}$
4. $41_{(10)}$	XOU	$23_{(10)}$
5. $22_{(10)}$	XOU	$44_{(10)}$

# Meio Somador

## Meio Somador de 1-bit



# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# Bibliografia

ALVES, William Pereira. "**Informática fundamental: introdução ao processamento de dados.**" Editora Érica, 1a. Edição (2010).

Capron, Harriet L., and J. A. **Johnson**. **Introdução à informática**. Vol. 8. No. 0. Pearson Prentice Hall, 2006.

POLLONI, ENRICO GIULIO FRANCO, and RICARDO DANIEL FEDELI. **Introdução à ciência da computação**. Cengage Learning Editores, 2010.

BROOKSHEAR, J. G., **Ciência da Computação, Uma Visão Abrangente**. 5a ed. Bookman Companhia Editora, 2000.

MARILYN M.; ROBERTA B. & PFAFFENBERGER, B., **Nosso Futuro e o Computador**. 3a ed. Bookman, 2000.

NORTON, PETER, **Introdução à Informática**, Editora Makron Books, 1997.

O'BRIEN, J. A., **Sistemas de Informação e as decisões gerenciais na era da Internet**. Ed. Saraiva, 2001.

WHITE, R., **Como Funciona o Computador**, 8a ed. Editora QUARK, 1998.

Apostilas. Páginas na Web. Vídeos na Web

Contatos:  
[tiagocompuesc@gmail.com](mailto:tiagocompuesc@gmail.com)

