CRP 292

Introdução à Informática

Prof. João Batista Ribeiro

joao421batista@gmail.com

Slides baseados no material da Prof.ª Larissa F. Rodrigues

1 Sistemas, códigos numéricos e aritmética binária

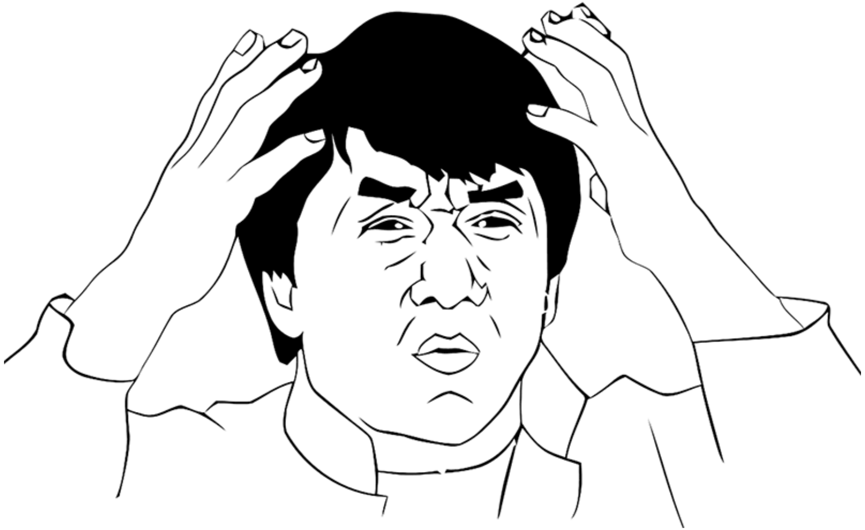
Conteúdo

- Introdução
- Sistemas Numéricos
- A Informação e sua Representação
- Conversões entre Sistemas Numéricos
- Aritmética com Números Binários
    - Soma
    - Subtração
    - Multiplicação
    - Divisão

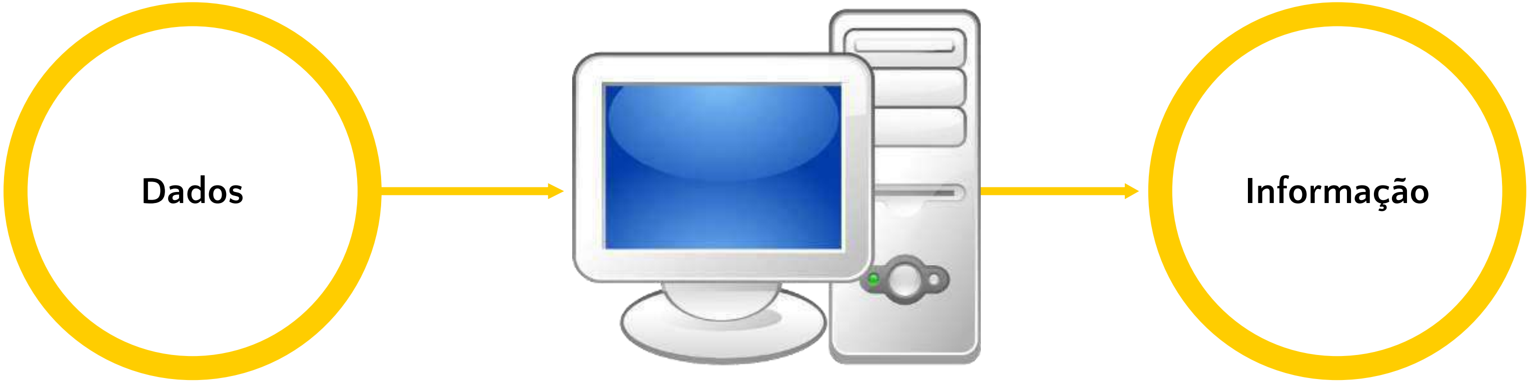
Introdução

Existem **10 tipos** de pessoas no mundo:

As que entendem binário e as que não entendem.



Introdução



Processamento

Sistemas Numéricos

- Os sistemas de numeração têm por objetivo estabelecer símbolos e convenções para representar quantidades, de forma a registrar a informação quantitativa e poder processá-la. A representação de quantidades faz-se com os números.

- O homem utilizou diversos sistemas numéricos antes de adotar o sistema decimal. “Resquícios” de bases numéricas ancestrais persistem até hoje, como a base 60, utilizada na contagem do tempo e na trigonometria.

Sistemas Numéricos

- Os sistemas de numeração são definidos pela base que eles utilizam, isto é, o número de dígitos que o sistema utiliza.
  - Exemplo: Sistema Decimal

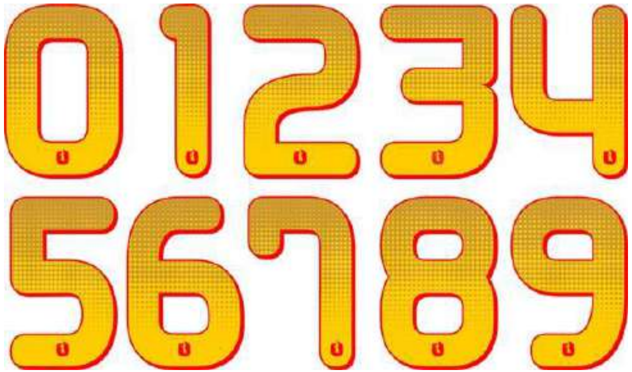
0 1 2 3 4 5 6 7 8 9

$$(765) = 700 + 60 + 5 = 7 \;.\; 10^{2} + 6 \;.\; 10^{1} + 5 \;.\; 10^{0}$$

Sistemas Numéricos

- Os sistemas de numeração são definidos pela base que eles utilizam, isto é, o número de dígitos que o sistema utiliza.
  - Exemplo: Sistema Decimal

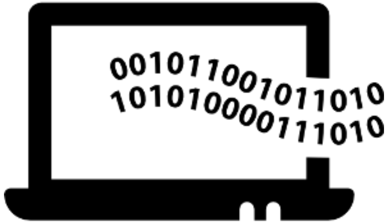
O sistema decimal utiliza 10 dígitos e possuí base 10



Sistemas Numéricos

- Como os computadores representam as informações usando dois estados, eles são adequados para números binários.

    - Desligado: 0
    - Ligado: 1



O sistema binário utiliza 2 dígitos e possuí base 2

Sistemas Numéricos

Sistema	Base	Algarismos
Binário	2	0, 1
Ternário	3	0, 1, 2
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Duodecimal	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A Informação e sua Representação

- Número binário no computador: BIT – Binary Digit
- É a menor unidade computacional
- Um bit pode representar apenas 2 símbolos (0 e 1)

A representação de toda e qualquer informação em um computador é, em seu nível mais elementar, constituído por conjuntos de bits

A Informação e sua Representação

- Um número de *n* bits pode representar 2^n valores distintos

BITS	Símbolos
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

A Informação e sua Representação

- BYTE (Binary Term)
  - Grupo ordenado de 8 bits
  - Unidade de armazenamento e transferência
  - Representa um caractere
  - Todas as letras, números e outros caracteres são codificados e decodificados por meio dos bytes

1 byte = 8 bits = 1 caractere (letra, número ou símbolo)

A Informação e sua Representação

Prefixo	Simbolo	Tamanho	
quilo	k	10^3	1000^1
mega	M	10^6	1000^2
giga	G	10^9	1000^3
tera	T	10^{12}	1000^4
peta	P	10^{15}	1000^5
exa	E	10^{18}	1000^6
zeta	Z	10^{21}	1000^7
iota	Y	10^{24}	1000^8

Unidade	Tamanho
Bit (b)	1 ou 0
Byte (B)	8 bits
Kilobyte (kB)	1000 B ou 10^3 bytes
Megabyte (MB)	1000 KB ou 10^6 bytes
Gigabyte (GB)	1000 MB ou 10^9 bytes
Terabyte (TB)	1000 GB ou 10^{12} bytes
Petabyte (PB)	1000 TB ou 10^{15} bytes
Exabyte (EB)	1000 PB ou 10^{18} bytes
Zettabyte (ZB)	1000 EB ou 10^{21} bytes
Yottabyte (YB)	1000 ZB ou 10^{24} bytes

A Informação e sua Representação

Prefixo	Simbolo	Tamanho	
quibi	Ki	2^{10}	1024^1
mebi	Mi	2^{20}	1024^2
gibi	Gi	2^{30}	1024^3
tebi	Ti	2^{40}	1024^4
pebi	Pi	2^{50}	1024^5
exbi	Ei	2^{60}	1024^6
zebi	Zi	2^{70}	1024^7
iobi	Yi	2^{80}	1024^8

Unidade	Tamanho
Bit (b)	1 ou 0
Byte (B)	8 bits
Kibibyte (KiB)	1024 B ou 2^{10} bytes
Mebibyte (MiB)	1024 KB ou 2^{20} bytes
Gibibyte (GiB)	1024 MB ou 2^{30} bytes
Tebibyte (TiB)	1024 GB ou 2^{40} bytes
Pebibyte (PiB)	1024 TB ou 2^{50} bytes
Exbibyte (EiB)	1024 PB ou 2^{60} bytes
Zebibyte (ZiB)	1024 EB ou 2^{70} bytes
Yobibyte (YiB)	1024 ZB ou 2^{80} bytes

A Informação e sua Representação

Unidade	Tamanho
Bit (b)	1 ou 0
Byte (B)	8 bits ou 1 carácter
Kilobyte (kB)	1000 ou 2^{10} bytes
Megabyte (MB)	1000 KB ou 2^{20} bytes
Gigabyte (GB)	1000 MB ou 2^{30} bytes
Terabyte (TB)	1000 GB ou 2^{40} bytes
Petabyte (PB)	1000 TB ou 2^{50} bytes
Exabyte (EB)	1000 PB ou 2^{60} bytes
Zettabyte (ZB)	1000 EB ou 2^{70} bytes
Yottabyte (YB)	1000 ZB ou 2^{80} bytes

Taxa de Transferência

1 Mbps vs 1 MBps

ps => por segundo

1.000 b por segundo

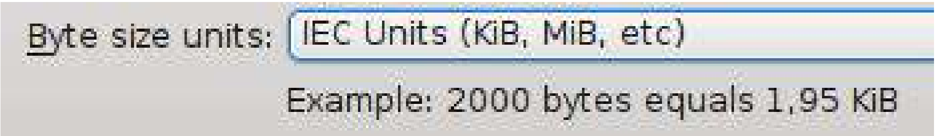
vs

1.000 B por segundo

b = 1 bit

B = 8 bits

A Informação e sua Representação



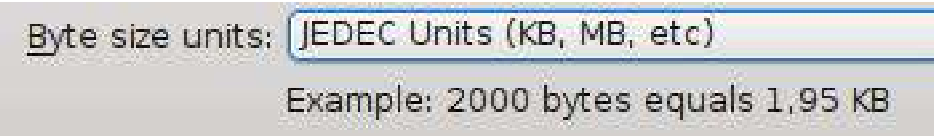


Tabela ASCII (alguns itens)

American Standard Code for Information Interchange
"Código Padrão Americano para o Intercâmbio de Informação"

Bin	Oct	Dec	Hex	Sinal
0011 0000	060	48	30	0
0011 0001	061	49	31	1
0011 0010	062	50	32	2
0011 0011	063	51	33	3
0011 0100	064	52	34	4
0011 0101	065	53	35	5
0011 0110	066	54	36	6

Bin	Oct	Dec	Hex	Sinal
0101 0000	120	80	50	P
0101 0001	121	81	51	Q
0101 0010	122	82	52	R
0101 0011	123	83	53	S
0101 0100	124	84	54	T
0101 0101	125	85	55	U
0101 0110	126	86	56	V

Binário	Caractere
0100 0001	A
0100 0010	B
0110 0001	a
0110 0010	b
0011 1100	<
0011 1101	=
0001 1011	ESC
0111 1111	DEL

https://pt.wikipedia.org/wiki/ASCII

A Informação e sua Representação

MEGABYTE

Lembra daquele **disquete** (ou disco flexível) que costumávamos usar para guardar dados? O de maior capacidade podia armazenar até **5,76 MB**: daria para salvar só **5 fotos digitais** ou ouvir um arquivo de música em mp3 com aproximadamente 5 minutos de duração.

*dados de Igor Bessera da Seagate

A Informação e sua Representação

GIGABYTE

Usar **pendrives** para guardar arquivos e levá-los onde você quiser já é algo bem comum. Num dispositivo de **1 GB**, daria para gravar 320 **fotos digitais** (.jpg), mas com resolução bem maior do que no exemplo anterior. Se fosse guardar só músicas, você gastaria 16 horas para ouvir toda a lista (dá para ir de avião de São Paulo a Moscou durante esse tempo).

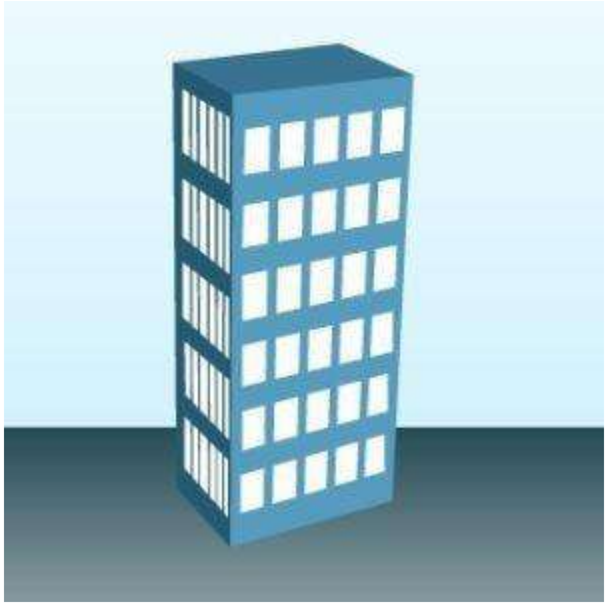
*dados de Igor Bessera da Seagate

A Informação e sua Representação

TERABYTE

Para aqueles que precisam de mais espaço, já existem **HDs (discos rígidos) externos**, que bem como pendrives tem a facilidade de serem portáteis. Um HD externo de **1 TB** pode armazenar cerca de 40 filmes em alta definição ou 500 jogos. Já em fotos digitais em alta resolução, seriam **320 mil** e em música digital, 16,6 mil horas (666 dias ou quase 1 ano e meses)

*dados de Igor Bessera da Seagate



PETABYTE

Para armazenar **1 PB** em dados, seria necessário um **datacenter** (local projetado especialmente para guardar dados de empresas) que ocuparia uma área total de **1.000 m²**, com **4.000 máquinas** (entre servidores e estações de trabalho)

*dados de Márcio Silva, pesquisador do Laboratório de Arquitetura e Redes do Departamento de Ciência da Computação da Poli-USP

A Informação e sua Representação

EXABYTE

Para armazenar **1 EB** em dados, seriam necessários **71 datacenters** que, juntos, ocupariam **9 campos de futebol**.
Se cada homem, mulher e criança do planeta guardasse consigo 1 pacote de arquivos de 2,5 GB (entre fotos, músicas, documentos, vídeos e outros), conseguiriam alcançar 1 EB – considerando que a população mundial de 6,9 bilhões de pessoas.

*dados de Márcio Silva, pesquisador do Laboratório de Arquitetura e Redes do Departamento de Ciência da Computação da Poli-USP

A Informação e sua Representação

ZETTABYTE

Para guardar **1 ZB** em volume de dados, seriam necessários **73 mil datacenters** que, juntos, ocupariam toda a área da **cidade de São Paulo** ou **9 mil campos de futebol**. Essa é a demanda aproximada de armazenamento no mundo, até o final deste ano.

*dados de Márcio Silva, pesquisador do Laboratório de Arquitetura e Redes do Departamento de Ciência da Computação da Poli-USP

A Informação e sua Representação

YOTABYTE

Por fim, temos **1 YB**, uma quantidade gigantesca de dados: para você ter uma ideia, seriam necessários 75 milhões de datacenters, que ocupariam toda área do Estado de São Paulo.

*dados de Márcio Silva, pesquisador do Laboratório de Arquitetura e Redes do Departamento de Ciência da Computação da Poli-USP

Conversões entre Sistemas Numéricos

Decimal para Binário

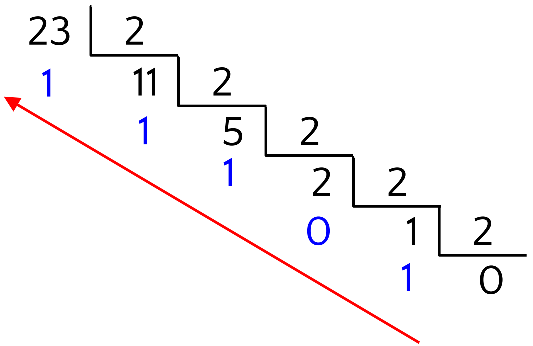
- Divida o número por 2 até que o quociente seja 0
- O número binário correspondente será formado pelos restos das divisões, sendo o resto da última divisão o dígito binário mais à esquerda (bit mais significativo)
- Exemplo: Converter o número 23 para binário

Conversões entre Sistemas Numéricos

Decimal para Binário

- Divida o número por 2 até que o quociente seja 0
- O número binário correspondente será formado pelos restos das divisões, sendo o resto da última divisão o dígito binário mais à esquerda (bit mais significativo)

- Exemplo: Converter o número 23 para binário



Resultado:
$(23)_{10} = (10111)_2$

Conversões entre Sistemas Numéricos

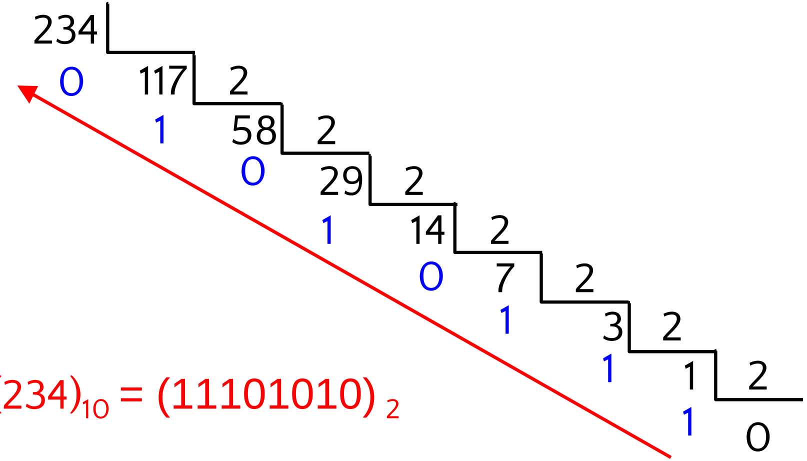
Decimal Fracionário para Binário

- Separar a parte inteira da fracionária
- Converter a parte inteira pelo método de divisões sucessivas por 2
- Converter a parte fracionária por multiplicações sucessivas por 2, até conseguir uma precisão satisfatória.
- O número fracionário convertido será composto por algarismos inteiros resultantes tomados nas multiplicações

Conversões entre Sistemas Numéricos

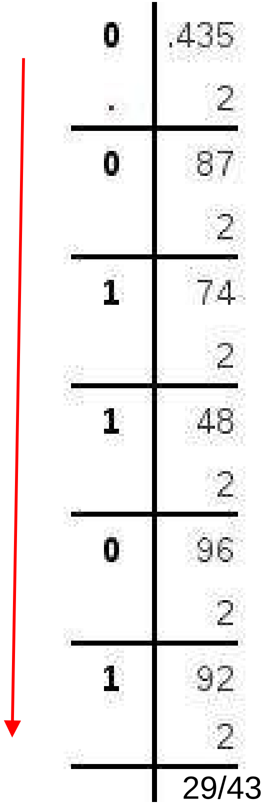
Decimal Fracionário para Binário

- Exemplo: Converter para binário (considerando 5 dígitos na parte decimal) o número 234,435



$(234)_{10} = (11101010)_2$

Resultado:
$(234.435)_{10} = (11101010.01101)_2$



Conversões entre Sistemas Numéricos

- **Binário para Decimal**

$(10111)_2 =$

$$\begin{array}{ccccc} 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{array} = 1x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 1x2^0$$

$$16 + 0 + 4 + 2 + 1$$

$$23$$

Conversões entre Sistemas Numéricos

- **Número Binário para Decimal**

Lei de Formação ampliada (polinômio)

$$a_n * b^n + a_{n-1} * b^{n-1} + a_{n-2} * b^{n-2} + \ldots + a_0 * b^0 + a_{-1} * b^{-1} + a_{-2} * b^{-2} + \cdots + a_{-m} * b^{-m}$$

  - Exemplo: $(101{,}110)_2 = (?)_{10}$

Aritmética com Números Binários

- **Adição**
  - **Regras:**
    - 0 + 0 = 0
    - 0 + 1 = 1
    - 1 + 0 = 1
    - 1 + 1 = 0 (e "vai 1" para o dígito de ordem superior)
    - 1 + 1 + 1 = 1 (e "vai 1" para o dígito de ordem superior)

Aritmética com Números Binários

Adição

- Na soma, segue-se sempre a ordem das colunas da direita para esquerda, tal como uma soma em decimal.
- Exemplo: 11100 + 11010

$$
\begin{array}{ccccccc}
 & 1 & 1 & & & & \\
 & & 1 & 1 & 1 & 0 & 0 \\
+ & & 1 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 1 & 0 \\
 & & \text{“vai” 1} & \text{“vai” 1} & & &
\end{array}
$$

Aritmética com Números Binários

Subtração

- Regras:
  - 0 - 0 = 0
  - 0 - 1 = 1 ("vem 1 do próximo")
  - 1 - 0 = 1
  - 1 - 1 = 0

  - Estouro = *borrow* (empréstimo) ou "vem 1"

Aritmética com Números Binários

Subtração

- Como é impossível tirar 1 de zero, o artifício é "pedir emprestado" 1 da casa de ordem superior
- Quando o dígito de ordem superior for 0, então procuramos pelo próximo dígito de ordem superior, até que ele seja 1
- Este bit torna-se então 0 e a todos os bits pulados (bits de valor 0) damos o valor 1

Aritmética com Números Binários

- **Subtração**
  - Exemplo: 11100 - 01010

$$
\begin{array}{cccccc}
 & & & \text{empréstimo} & & \\
 & & & & 1 & \\
 & 1 & 1 & \cancel{1} & 0 & 0 \\
- & 0 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 & 0
\end{array}
$$

Aritmética com Números Binários

- **Multiplicação**
  - **Regras:**
    - 0 x 0 = 0
    - 0 x 1 = 0
    - 1 x 0 = 0
    - 1 x 1 = 1

- A única diferença ao se realizar multiplicação em binários, em relação à multiplicação em decimal, é que a soma final deve ser feita em binário.

Aritmética com Números Binários

- **Multiplicação**
  - Exemplo: 1011 x 1101

				1	0	1	1
			X	1	1	0	1
				1	0	1	1
			0	0	0	0	
		1	0	1	1		
	1	0	1	1			
1	0	0	0	1	1	1	1

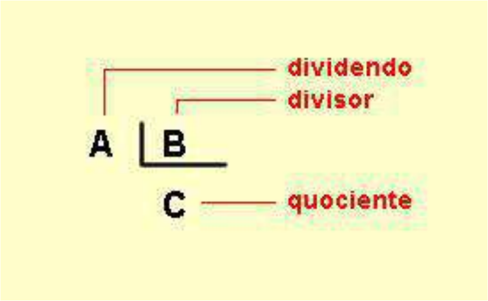
Aritmética com Números Binários

Divisão

- Pode ser feita de maneira idêntica à divisão decimal
- A diferença reside no fato das multiplicações e subtrações internas ao processo serem feitas em binário
- O valor do **divisor** deve ser igual ou menor que o modo **dividendo** e, se for igual ou menor é escrito 1 no quociente. Esse valor é multiplicado pelo divisor e subtraído do dividendo, até atingir o valor zero, no caso da divisão exata.

Aritmética com Números Binários

- **Divisão**
  - Exemplo: 110111/101:




```
110111 | 101
101      1011
00111
  101
  0101
   101
   000
```

55/5 = 11

Exercícios

1) Qual o decimal equivalente a $(11011011)_2$?

2) Qual o binário equivalente à sua idade?

3) Converter os seguintes números decimais para números binários

a) 39

b) 0, 5625

c) 256, 75

d) 129, 625

Exercícios

4) Execute as seguintes operações:

a) 0011 + 1110

b) 1110 – 0100

c) 1101101/1011

d) 10011 x 1101

Bibliografia

• MANZANO, A. L. N. G.; MANZANO, M. I. N. G.
Estudo dirigido de informática básica. 7.ed. Érica, 2007.

- Capítulo 03

Obrigado pela atenção! :)