

```
1
2
3  Algoritmo 'Lógica de' {
4
5      [ Programação 1 ]
6
7
8
9
10
11
12 }
13
14
```

Unidade 01

Prof. Daniel Caixeta



```
1  'Sumário' {
2      </01> 0 início [...]
3          < 1.1. Uma breve introdução. 1.2. Alguns conceitos [...]. >
4
5      </02> Como a máquina vê [...]
6          < 2.1. Bits e Bytes. 2.2. A representação [...]. >
7
8      </03> Como a máquina pensa [...]
9          < 3.1. Sistema posicional. > < 3.2. As bases [...]. 3.3. Conversão entre
10         bases 2, 8 e 16. 3.4. Conversão da base decimal (10) para binária (2). >
11
12      </04> A Lógica binária
13          < 4.1. Sistemas dicotômicos e a Lógica de Boole. 4.2. A lógica binária. >
14  Referência
15 }
```

1 01 {
2
3

4
5 [Iniciando ]
6
7

8 < Introdução aos estudos [...] >
9
10

11
12 }
13
14

1.1. Uma breve introdução

- Hoje em dia os computadores são essencialmente importantes, principalmente quando se trata-se de comunicação, *e-commerce* redes sociais, métricas comportamentais, processos decisórios, etc.
- “Quase tudo” é computacionalmente possível, e através de algoritmos, esses processos tem se tornado cada vez mais autônomos e automatizados.
- Então a pergunta é:

O que poderia vir após a Era Digital?



1.2. Alguns conceitos [...]

- A informática pode ser entendida como a ciência que estuda o conjunto de informações e conhecimentos por meios digitais.

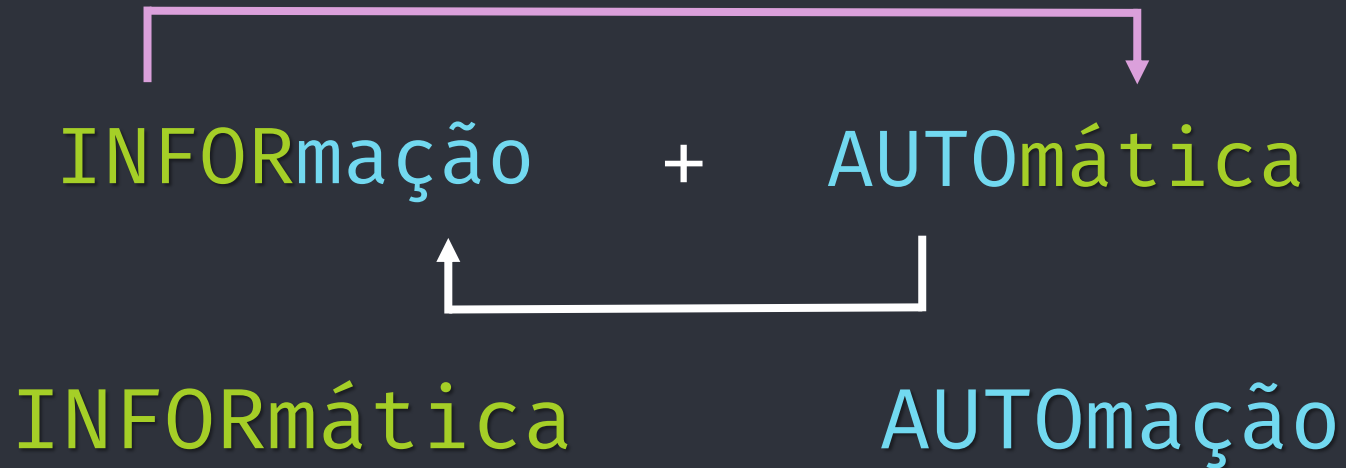


Figura 1. Informática e Automação - Conceitos.

- A informática descreve o conjunto de disciplinas/ciências que através de suas interrelações constroem soluções para a(o):



Figura 2. Tarefas realizadas por máquinas/computadores.

- A computação pode ser definida como a busca de solução para um problema a partir de entradas (*inputs*), de forma a obter resultados (*outputs*) depois de processado os dados através de um algoritmo.

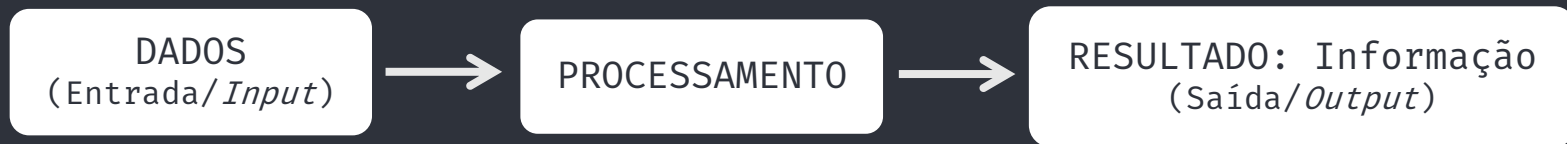


Figura 3. Etapas básicas de um processamento de dados. (MONTEIRO, 2014).

- Durante milhares de anos, a computação foi executada com caneta e papel, com giz e ardósia, ou mentalmente, por vezes com o auxílio de tabelas ou utensílios artesanais.

- A palavra computador significa aquele que faz cálculos, seja ele, pessoa ou máquina. Não há dúvida que as pessoas foram os primeiros computadores [...]. Inclusive, o surgimento de uma simples calculadora mecânica causou uma revolta, pois as pessoas tiveram medo de perder seus empregos. (FARIAS, 2013).

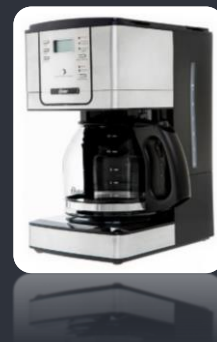


- Já para Monteiro (2014), trata-se de uma máquina (conjunto de partes eletrônicas e eletromecânicas) capaz de sistematicamente coletar, manipular e fornecer resultados da manipulação de informações para um ou mais objetivos (ver Figura 3).

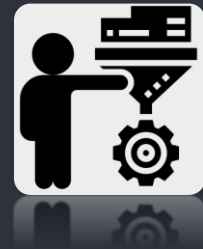
- Por se tratar de uma máquina programável, essa é capaz de realizar uma variedade de tarefas, cálculos, rotinas, procedimentos, etc., seguindo uma sequência de comandos de acordo com o que foi ordenado.
- Trata-se de um sistema computacional, formado pela integração de componentes que atuam como uma entidade cujo propósito é processar dados, i.e., realizar algum tipo de operação aritmética/lógica envolvendo dados, de modo a produzir diferentes níveis de informação.



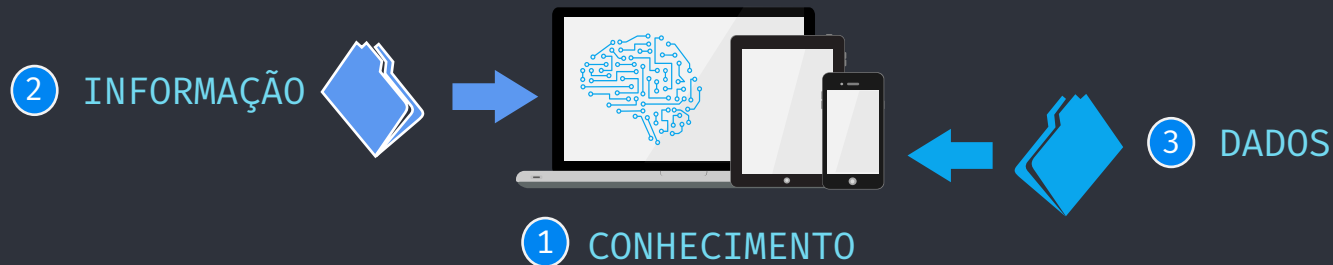
=



- 1 • Por ser uma máquina composta de vários circuitos e
2 componentes eletrônicos, também já foi chamado de equipamento
3 de processamento eletrônico de dados. (MONTEIRO, 2014).
4
- 5 • Definindo processamento de dados (*Data Processing*):
6
7 Consiste em uma série de atividades ordenadamente
8 realizadas com o objetivo de produzir um arranjo
9 determinado de informações a partir de outras
10 obtidas inicialmente. (*ibidem*).
11
- 12 • A manipulação das informações que foram coletadas no início
13 da atividade chama-se processamento. (*ibidem*).
14



- 1 • Já dados e informações podem ser tratados como sinônimos ou
2 não. Dados é a matéria-prima, e informação é o resultado do
3 processamento.
- 4 • Subtende-se ainda, informação como dados organizados para o
5 atendimento ou emprego para uma finalidade, podendo ser
6 usado por uma pessoa/usuário ou um grupo.
- 7 • Como o conhecimento e a tomada de decisão são importantes
8 em diversas áreas e em diferentes níveis organizacionais,
9 os conceitos podem mudar, dependendo do contexto.



- Segundo Monteiro (2014), com a imensa quantidade de informações processadas e atualizadas rapidamente, tornou-se imprescindível e essencial a utilização de computadores em praticamente todas as atividades e segmentos da sociedade.




Figura 4. A Internet das coisas/*Internet of things* (IoT).

- O próprio avanço tecnológico na área de telecomunicação também contribuiu para o crescimento do uso de computadores, *notebooks*, *tablets*, *smartphones*, etc., visto que permitiu a interligação entre as redes de comunicação de dados.
- O principal exemplo é a *internet*, que permite a comunicação entre qualquer dispositivo, em qualquer parte do planeta.



Figura 5. A *internet* - Rede de comunicação mundial.

02 {

[Como a máquina ]

< Dos *bits* aos *bytes* [...] >

}

2.1. Bits e Bytes

- Um *bit* ou dígito binário (*binary digit*), é:
[...] a unidade básica que os computadores e sistemas digitais utilizam para trabalhar, e pode assumir apenas dois valores, 0 ou 1. (FARIAS, 2013).
- Já um *byte* é uma sequência de 8 *bits*. Portanto, o *byte* é a menor unidade de armazenamento utilizada pelos computadores. Quer dizer, que jamais conseguiremos salvar menos do que 8 *bits* em uma informação.

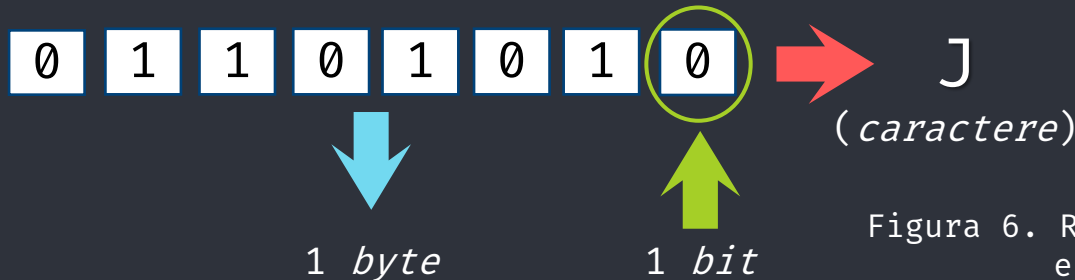


Figura 6. Representação de um byte e um bit.

2.2. A representação [...]





- Um *bit* só pode assumir dois valores (0 ou 1), portanto, só será possível representar exatamente dois estados distintos. (FARIAS, 2013).

Tabela 1. Representação com um *bit*.

Bit	Porta	Lâmpada	Detector de movimento	Estado civil
0	Fechada	Desligada	Sem movimento	Solteiro
1	Aberta	Ligada	Com movimento	Casado

- Para representar mais de dois valores distintos precisamos de uma sequência de *bits* maior. Na Tabela 2, é apresentado exemplos utilizando uma sequência de 2 *bits*, obtendo assim 4 possibilidades.

Tabela 2. Representação com dois *bits*.

Sequência de bits	Semáforo
00	Desligado 
01	Pare 
10	Atenção 
11	Siga 

- Segundo Farias (2013), o número de possibilidades diferentes que podemos representar depende do tamanho da sequência que estamos utilizando, mais precisamente:

$$P = 2^n, \text{ onde } n = \text{tamanho de } \textit{bits}.$$

- Exemplos:

$$2^1 = 2 \qquad 2^2 = 4 \qquad 2^3 = 8 \qquad 2^4 = 16$$

$$2^5 = 32 \qquad 2^6 = 64 \qquad 2^7 = 128 \qquad 2^8 = 256$$

(1 *byte*)

$$16 \text{ bits} = 65.535$$

$$32 \text{ bits} = 4.294.967.295$$

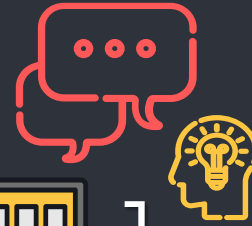
$$64 \text{ bits} = 18.446.744.073.709.551.615$$

03 {

[Como a máquina



]



< Como o computador pensa e executa? >

}

3.1. Sistema posicional

- O método de numeração de quantidades que adotamos, utiliza um sistema de numeração posicional.
- Significa que a posição ocupada por cada algarismo em um número altera seu valor de uma potência decimal (base 10) para cada casa à esquerda. Vejamos o exemplo abaixo:

$$128_{10} = \underbrace{1 \times 10^2}_{100 \text{ Centena}} + \underbrace{2 \times 10^1}_{20 \text{ Dezena}} + \underbrace{8 \times 10^0}_{8 \text{ Unidade}}$$

3.2. As bases [...]

- A base de um sistema é a quantidade de algarismos disponíveis em sua representação.
- Citemos alguns exemplos:

</1> Base decimais (b_{10}) \rightarrow 0 1 2 3 4 5 6 7 8 9

</2> Base binária (b_2) \rightarrow 0 1

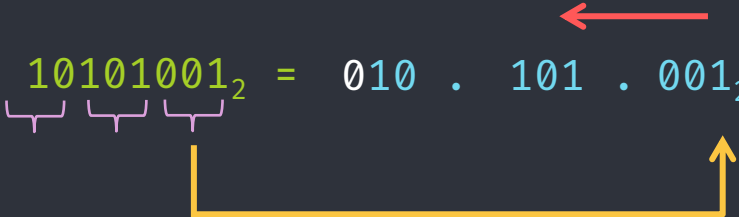
</3> Base octal (b_8) \rightarrow 0 1 2 3 4 5 6 7

</4> Base hexadecimal (b_{16}) \rightarrow 0 1 2 3 4 5 6 7 8 9 A B C D E F
(10 algarismos + 6 símbolos)

3.3. Conversão entre bases 2, 8 e 16

- As conversões mais simples são as que envolvem bases que são potências entre si. (FARIAS, 2013).
- Leva-se em consideração que $2^3 = 8$ e $2^4 = 16$.
- Exemplifiquemos a conversão 2^3 , que funciona da seguinte forma:
 - 1º. Separa-se os algarismos de um número binário em grupos de três (começando sempre da direita para a esquerda).
 - 2º. Converte cada grupo de três algarismos por seu equivalente em octal.
- Vejamos:

1
2 $10101001_2 = 010 . 101 . 001_2$
3
4
5
6
7
8
9
10
11
12
13
14



• Olhando a tabela de conversão direta temos:

$$010_2 = 2_8 \quad 101_2 = 5_8 \quad 001_2 = 1_8 \quad 251_8$$

$$10101001_2 = 251_8$$

Bin.	Octal
000	0
001 →	1
010 →	2
011	3
100	4
101 →	5
110	6
111	7

- Agora a conversão entre as bases 2 e 16.
- Como $2^4 = 16$, seguimos o mesmo processo anterior, bastando agora separarmos em grupos com quatro algarismos e converter cada grupo seguindo a Tabela ao lado. Por exemplo:

$$11010101101_2 = \underbrace{0110}_2 . \underbrace{1010}_2 . \underbrace{1101}_2$$

$$0110_2 = 6_{16}$$

$$1010_2 = A_{16}$$

$$1101_2 = D_{16}$$

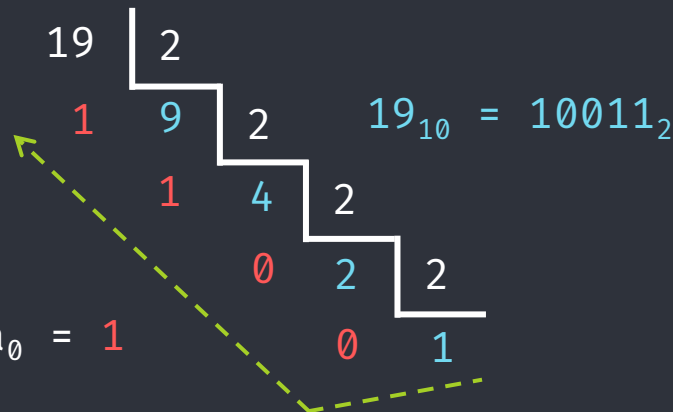
$$11010101101_2 = 6AD_{16}$$

Bin.	Hexa.	Bin.	Hexa.
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

3.4. Conversão da base 10 para bin.

- 1
- 2 • Já a conversão de números da base 10 para uma base
- 3 qualquer, emprega-se algoritmos que serão de ordem inversa
- 4 das anteriores.
- 5 • O número decimal será dividido sucessivas vezes pela base
- 6 2, o resto de cada divisão ocupará sucessivamente as
- 7 posições de ordem 0 e 1, e assim por diante, até que o
- 8 resto da última divisão resulte em quociente 0.
- 9
- 10 • Esse último quociente irá ocupar a posição mais alta ordem.
- 11
- 12
- 13
- 14

- Exemplo: Converta o numero 19_{10} para a base 2.



$$a_4 = 1 \quad a_3 = 0 \quad a_2 = 0 \quad a_1 = 1 \quad a_0 = 1$$

- Usando a conversão anterior como prova real, temos:

$$10011_2 = (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 19_{10}$$

04 {

[A Lógica binária]

< Operadores e circuitos lógicos >

}

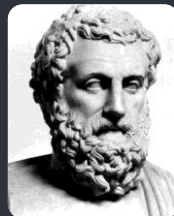
4.1. Sistemas dicotômicos e a Álgebra de Boole

- Segundo Daghlia (2008), o mundo em que vivemos apresenta situações dualísticas em sua grande maioria, ou seja, com dois estados que mutuamente se excluem.

Situações dualísticas.

Valor_1	Valor_2
1	0
Sim	Não
Dia	Noite
Preto	Branco
Ligado	Desligado

- 1 • Segundo Daghlia (2008) a Lógica começou a
2 se desenvolver no século IV a.C., com
3 Aristóteles.
- 4 • Neste período os filósofos gregos passaram a
5 usar em suas discussões sentenças lógicas
6 enunciadas nas formas afirmativas e
7 negativas, resultando assim em grande
8 simplificação da realidade no dia a dia
9 [...]. E quase 2.000 anos depois, por volta
10 de 1666, Leibniz usou em vários trabalhos o
11 que chamou de *Calculus ratiotinator*. Essas
12 ideias nunca foram teorizadas, porém seus
13 escritos trazem a ideia do que seria a
14 Lógica Matemática.



Aristóteles
(384 – 322 a.C.)



Gottfried W. Leibniz
(1646 – 1716)

- Em 1854, George Boole formalizou o tratamento da lógica, chamada de Álgebra Booleana, e definida como um conjunto de operadores e axiomas, assumidos como verdadeiros, e sem a necessidade de prova. (Güntzel & Nascimento, 2001).
- Em 1938, C. E. Shannon aplicou esta álgebra para mostrar que os circuitos elétricos de chaveamento podem ser representadas com dois valores. (*ibidem*).
- Para Güntzel & Nascimento (2001), diferentemente da álgebra ordinária, onde as variáveis podem assumir valores no intervalo $(-\infty ; +\infty)$, as variáveis booleanas só podem assumir um número finito de valores.

0 ou 1

- Em particular, na álgebra booleana de dois valores, cada variável pode assumir um dentre dois possíveis. Por exemplo:

Exemplo de operadores.

Operador	Valores
V ou F	Verdadeiro ou Falso
C ou E	Certo ou Errado
0 ou 1	–

- Portanto são sistemas dicotômicos. Computacionalmente dizendo, 0 e 1, a qual é também utilizada na eletrônica digital de circuitos.

4.2. A lógica binária

- 1
2 • Em 1854, George Boole publica a Álgebra booleana como sendo
3 um sistema completo que permitia a construção de modelos
4 matemáticos para o processamento computacional.
- 5 • O interessante na lógica booleana é que a partir de três
6 operadores básicos (NOT, AND e OR), podemos construir
7 circuitos lógicos capazes de realizar diversas operações
8 computacionais.
- 9 • Como o número de valores que cada variável pode assumir é
10 finito (e pequeno), o número de estados que uma função
11 booleana pode assumir também será finito, o que significa
12 que podemos descrevê-las utilizando tabelas.
13
14

- Essa tabela recebe o nome de Tabela Verdade, e nela são listadas todas as combinações de valores que as variáveis de entrada podem assumir e os correspondentes valores da função (saídas).

</1> Operador NOT

O operador unário NOT, negação binária, resulta no complemento do operando, ou seja, será um *bit* 1 se o operando for 0, e será 0 caso contrário, conforme apresentado tabela ao lado, onde A é o *bit* de entrada e S é a resposta, ou *bit* de saída.

Tabela verdade do operador NOT

A	S ou A'
0	1
1	0

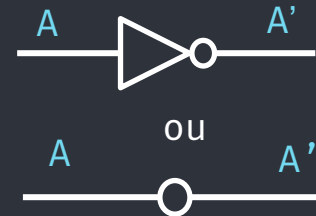


Figura 6. Representação gráfica do operador lógico NOT, com seus valores de entrada e saída.

</2> Operador AND

O operador binário AND, ou conjunção binária devolve um *bit* 1 sempre que ambos operandos sejam 1, conforme podemos confirmar na tabela ao lado, onde A e B são *bits* de entrada, e S é o *bit*-resposta, ou *bit* de saída.

Tabela verdade do operador AND

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

$$(A \cdot B)$$

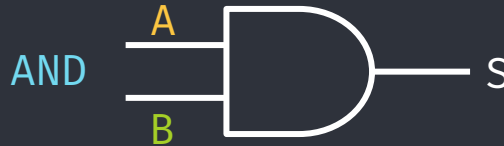


Figura 7. Representação gráfica do operador lógico AND, com seus valores de entrada e saída.

</3> Operador OR

O operador binário OR, ou disjunção binária devolve um *bit* 1 sempre que pelo menos um dos operandos seja 1, conforme podemos confirmar na tabela ao lado, onde A e B são os *bits* de entrada, e S é o *bit*-resposta, ou *bit* de saída.

Tabela verdade do operador OR

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

$$(A + B)$$



Figura 8. Representação gráfica do operador lógico OR, com seus valores de entrada e saída.

</1> DAGHLIAN, Jacob. Lógica e Álgebra de Boole. 4ª ed. 12ª reimpr. São Paulo : Atlas, 2008.

</2> FARIAS, Gilberto. Introdução à computação. UFPB, 2013.

</3> GÜNTZEL, José Luís; NASCIMENTO, Francisco de Assis. Introdução aos Sistemas Digitais (v.2001/1). Disponível in:
<https://www.inf.ufsc.br/~j.guntzel/isd/isd.html>.

</4> TANENBAUM, Andrew S. AUSTIN, Todd. Organização Estruturada de Computadores. 6ª ed. Pearson, 2013.

</5> MONTEIRO, Mário A. Introdução à Organização dos Computadores. 5ª ed. LTC. 2014.

A 'Obrigado' Is {

Algoritmo & Linguagem de Programação 1

|
}

