



INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

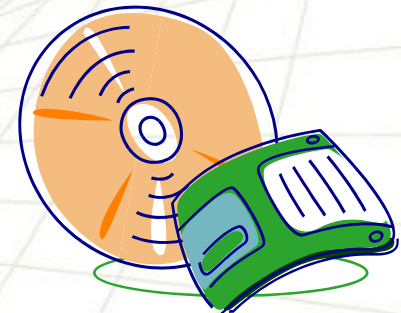


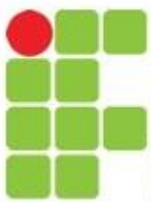
# Unidades de Medidas Computacionais

Professor:

Vilson Heck Junior

[vilson.junior@ifsc.edu.br](mailto:vilson.junior@ifsc.edu.br)





INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

Unidades de Medidas Computacionais

# QUANTIFICAÇÃO DE DADOS

- 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;
  - Etc...

- 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;

# Agrupando 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

# Conhecendo o 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 pronuncia ou escrita:
  - 1000 gramas: 1Kg
  - 100000 metros: 100Km
- 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: 1KM = 1m x  $10^3$ , logo = 1000m
- Como a base numérica computacional é binária, na computação costumamos dividir os números na base 2.
  - Ex: 1KB = 1B x  $2^{10}$ , logo = 1024B

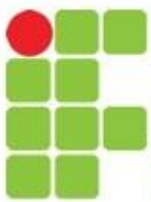


- Mais comuns:

Lembrar!

Unidade	Quantidade
Byte ( <b>B</b> )	1 (composto por 8 bits [ <b>b</b> ])
Kbyte ( <b>KB</b> ) [ <i>kilobyte</i> ]	$1 \times 2^{10} = 1024$
Mbyte ( <b>MB</b> ) [ <i>megabyte</i> ]	$1 \text{KB} \times 2^{10}$ ou $1 \text{B} \times 2^{20} = 1.048.576$
Gbyte ( <b>GB</b> ) [ <i>gigabyte</i> ]	$1 \text{MB} \times 2^{10}$ ou $1 \text{B} \times 2^{30} = 1.073.741.284$
Tbyte ( <b>TB</b> ) [ <i>terabyte</i> ]	$1 \text{GB} \times 2^{10}$ ou $1 \text{B} \times 2^{40}$
Pbyte ( <b>PB</b> ) [ <i>petabyte</i> ]	$1 \text{TB} \times 2^{10}$ ou $1 \text{B} \times 2^{50}$
Ebyte ( <b>EB</b> ) [ <i>exabyte</i> ]	$1 \text{PB} \times 2^{10}$ ou $1 \text{B} \times 2^{60}$
Zbyte ( <b>ZB</b> ) [ <i>zettabyte</i> ]	$1 \text{EB} \times 2^{10}$ ou $1 \text{B} \times 2^{70}$
Ybyte ( <b>YB</b> ) [ <i>yottabyte</i> ]	$1 \text{ZB} \times 2^{10}$ ou $1 \text{B} \times 2^{80}$

- 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.



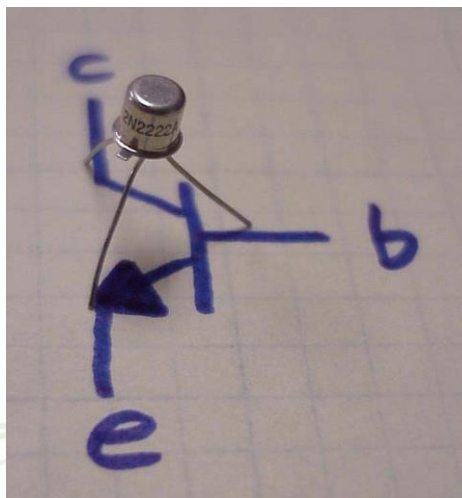
INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

Unidades de Medidas Computacionais

# REPRESENTAÇÃO DE DADOS

# Por que *bit*?

- **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.

- 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

- [ASCII Wikipedia](#) – Tabela Completa
- Exemplos:

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

# Padrão Unicode

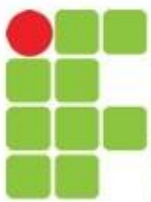
- Permite ao computador representar texto em qualquer sistema de escrita existente;
- Mais de 107 mil caracteres comportados;
- É composto por um conjunto de diagramas de códigos e metodologias de codificação;
- É desenvolvido e mantido por um consorcio sem fins lucrativos chamado *Unicode Consortium* 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: [Exemplo](#).

# Armazenando Dados

- Faça esse experimento:
  - Abra o Bloco de Notas e insira a frase: ***Boa tarde!***  
Salve o arquivo no disco com o nome de ***curso.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.



INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

Unidades de Medidas Computacionais

# CONVERSÃO ENTRE BASES NUMÉRICAS BINÁRIO

# Conversão entre Bases

- Binária:
  - $1010_{(2)}$
- Decimal:
  - $10_{(10)}$
- Octal:
  - $12_{(8)}$
- Hexadecimal:
  - $A_{(16)}$



# (10) $\rightarrow$ (2)

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

## Decimal <sub>(10)</sub>      Binário <sub>(2)</sub>

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



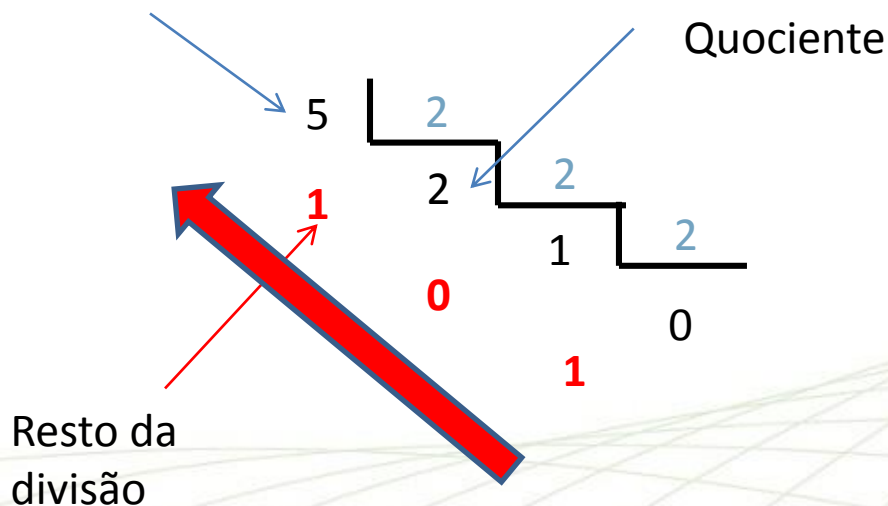
Existem 10 tipos de pessoas:

- As que entendem binário;
- As que não entendem binário.



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

Número a ser convertido



$$5_{(10)} = \mathbf{101}_{(2)}$$

*O número binário é escrito a partir dos restos das divisões e sempre de baixo para cima.*

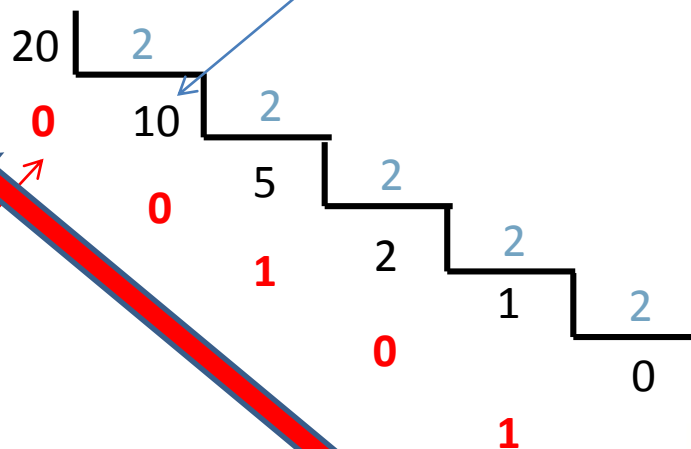
Ex: Converter o número 20 para a base 2.

## Importante

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

Número a ser convertido

Quociente



$$20_{(10)} = 10100_{(2)}$$

Resto da divisão

*O número binário é escrito a partir dos restos das divisões e sempre de baixo para cima.*

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

<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
$0 * 128$	$0 * 64$	$1 * 32$	$0 * 16$	$0 * 8$	$1 * 4$	$1 * 2$	$0 * 1$

$$- 0 + 0 + 32 + 0 + 0 + 4 + 2 + 0 = \mathbf{38}_{(10)}$$

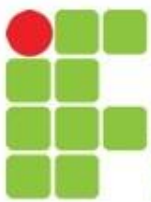
# Conversão entre Bases

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

<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
$1 * 128$	$0 * 64$	$1 * 32$	$1 * 16$	$0 * 8$	$1 * 4$	$0 * 2$	$1 * 1$

$$- 128 + 0 + 32 + 16 + 0 + 4 + 0 + 1 = \mathbf{181}_{(10)}$$

1.  $99_{(10)}$  em binário?
2.  $325_{(10)}$  em binário?
3.  $7858_{(10)}$  em binário?
4.  $28591_{(10)}$  em binário?
5.  $101010_{(2)}$  em decimal?
6.  $11001100_{(2)}$  em decimal?
7.  $111011010001_{(2)}$  em decimal?
8.  $1000000000000000_{(2)}$  em decimal?
9. Converta sua idade para binário.



INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

Dados e Numeração

# OPERAÇÕES LÓGICAS



- As vezes teremos operações lógicas para realizar com números binários;
- Estas operações estão relacionadas à lógica proposicional (estuda em Programação) e também ao cálculo de sub-redes em Redes;
- Por definição, assuma:
  - Dígito binário 0 = Falso;
  - Dígito binário 1 = Verdadeiro;

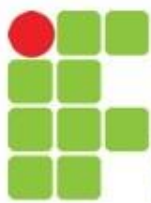
- As Principais Operações lógicas são:

<u>Português:</u>	<u>Inglês:</u>
Não	Not
E	And
Ou	Or
Ou exclusivo	Xor

- Numa operação matemática, recebemos pelo valores numéricos e iremos gerar 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:
  - $10101$  ou  $01010 = 11111$

# 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)



# Operador Lógico Não

$$\begin{array}{r} \text{Não} \quad \underline{11} \quad \leftarrow (3) \\ 00 \quad \rightarrow (0) \end{array}$$

$$\begin{array}{r} \text{Não} \quad \underline{110011} \quad \leftarrow (51) \\ 001100 \quad \rightarrow (12) \end{array}$$

# Operador Lógico E

- Escrita:  $v1 \text{ E } v2 = \text{resultado}$
- Tabela Verdade:

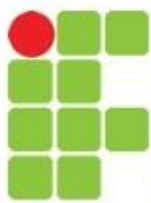
v1	v2	resultado
0	0	0
0	1	0
1	0	0
1	1	1

# Operador Lógico E

$$\begin{array}{r}
 101 \leftarrow (5) \\
 \text{E } 011 \leftarrow (3) \\
 \hline
 001 \rightarrow (1)
 \end{array}$$

$$\begin{array}{r}
 1010101 \leftarrow (85) \\
 \text{E } 0110011 \leftarrow (51) \\
 \hline
 0010001 \rightarrow (17)
 \end{array}$$





# Exercícios (E)

1.	1101	(2)	<b>E</b>	101	(2)
2.	111	(2)	<b>E</b>	7	(10)
3.	50	(10)	<b>E</b>	100	(10)
4.	32	(10)	<b>E</b>	63	(10)
5.	7	(10)	<b>E</b>	15	(10)

# Operador Lógico OU

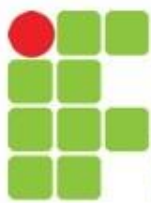
- Escrita:  $v1 \text{ OU } v2 = \text{resultado}$
- Tabela Verdade:

v1	v2	resultado
0	0	0
0	1	1
1	0	1
1	1	1

# Operador Lógico OU

$$\begin{array}{r}
 101 \leftarrow (5) \\
 \text{OU } 011 \leftarrow (3) \\
 \hline
 111 \rightarrow (7)
 \end{array}$$

$$\begin{array}{r}
 1010101 \leftarrow (85) \\
 \text{OU } 0110011 \leftarrow (51) \\
 \hline
 1110111 \rightarrow (119)
 \end{array}$$



# Exercícios (OU)

1. 1101<sub>(2)</sub>

**OU** 101<sub>(2)</sub>

2. 111<sub>(2)</sub>

**OU** 8<sub>(10)</sub>

3. 40<sub>(10)</sub>

**OU** 101<sub>(10)</sub>

4. 31<sub>(10)</sub>

**OU** 33<sub>(10)</sub>

5. 333<sub>(10)</sub>

**OU** 666<sub>(10)</sub>

# O que faz este circuito?

- Assumindo:

$$S = \bar{A} \text{ **xou** } B$$

$$C = A \text{ **e** } B$$

- Sendo:

$$\begin{array}{r} C \cdot \\ A \\ B \\ \hline C \quad S \end{array}$$

# Somador Completo

- Assumindo:

$$S = ((A \text{ XOU } B) \text{ E } \text{NÃO}(Ci)) \text{ OU } (\text{NÃO}(A \text{ XOU } B) \text{ E } Ci)$$

$$Co = ((A \text{ E } B) \text{ E } \text{NÃO}(Ci)) \text{ OU } ((A \text{ OU } B) \text{ E } Ci)$$

- Sendo:

Co	Ci
	A
	B
	S

# Operador Lógico OU Exclusivo

- Escrita:  $v1 \text{ XOU } v2 = \text{resultado}$
- Tabela Verdade:

v1	v2	resultado
0	0	0
0	1	1
1	0	1
1	1	0

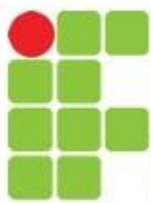




# Operador Lógico XOR

$$\begin{array}{r} 101 \leftarrow (5) \\ \text{XOR } 011 \leftarrow (3) \\ \hline 110 \rightarrow (6) \end{array}$$

$$\begin{array}{r} 1010101 \leftarrow (85) \\ \text{XOR } 0110011 \leftarrow (51) \\ \hline 1100110 \rightarrow (102) \end{array}$$



# Exercícios (XOU)

1.  $1101_{(2)}$

**XOU**  $101_{(2)}$

2.  $111_{(2)}$

**XOU**  $9_{(10)}$

3.  $60_{(10)}$

**XOU**  $121_{(10)}$

4.  $41_{(10)}$

**XOU**  $23_{(10)}$

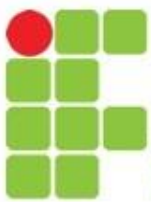
5.  $222_{(10)}$

**XOU**  $444_{(10)}$

- Converta cada letra do seu nome para decimal;

LETRA	BINÁRIO	LETRA	BINÁRIO	LETRA	BINÁRIO	LETRA	BINÁRIO
A	1000001	H	1001000	O	1001111	V	1010110
B	1000010	I	1001001	P	1010000	W	1010111
C	1000011	J	1001010	Q	1010001	X	1011000
D	1000100	K	1001011	R	1010010	Y	1011001
E	1000101	L	1001100	S	1010011	Z	1011010
F	1000110	M	1001101	T	1010100		
G	1000111	N	1001110	U	1010101		

- Some os números!



INSTITUTO FEDERAL  
SANTA CATARINA  
Campus Lages

Unidades de Medidas Computacionais

# CONVERSÃO ENTRE BASES NUMÉRICAS HEXADECIMAL