

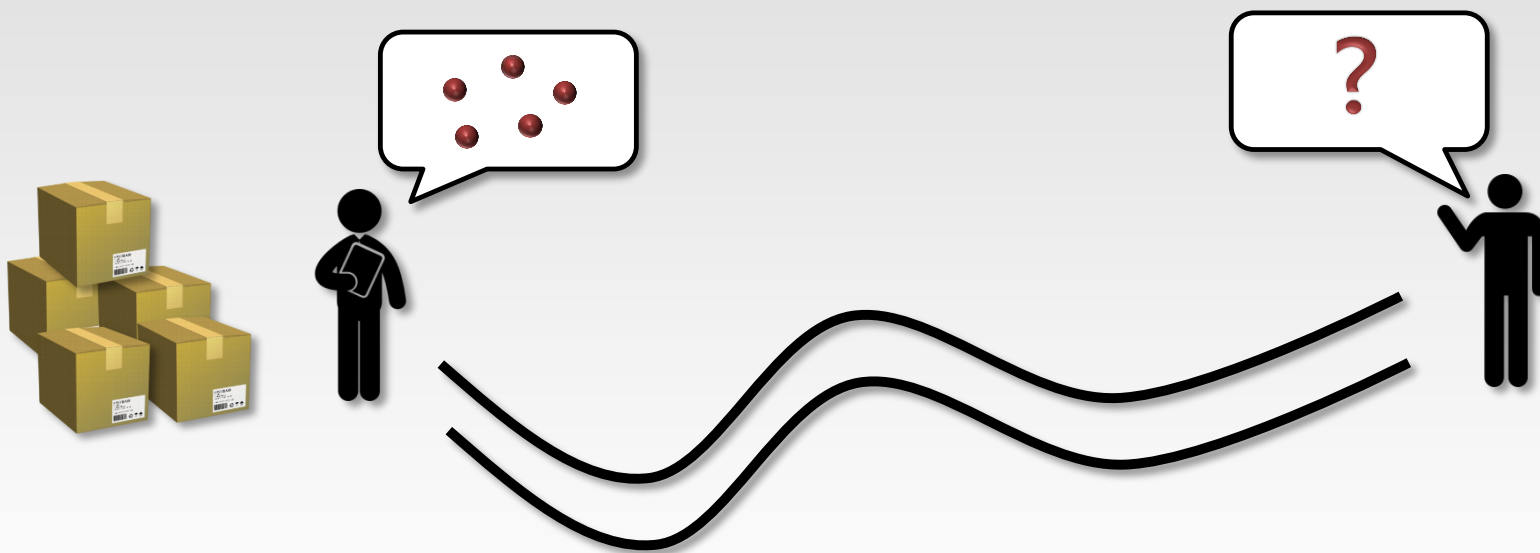
# Introdução à Computação

## Aulas 13/14 – Representação de dados

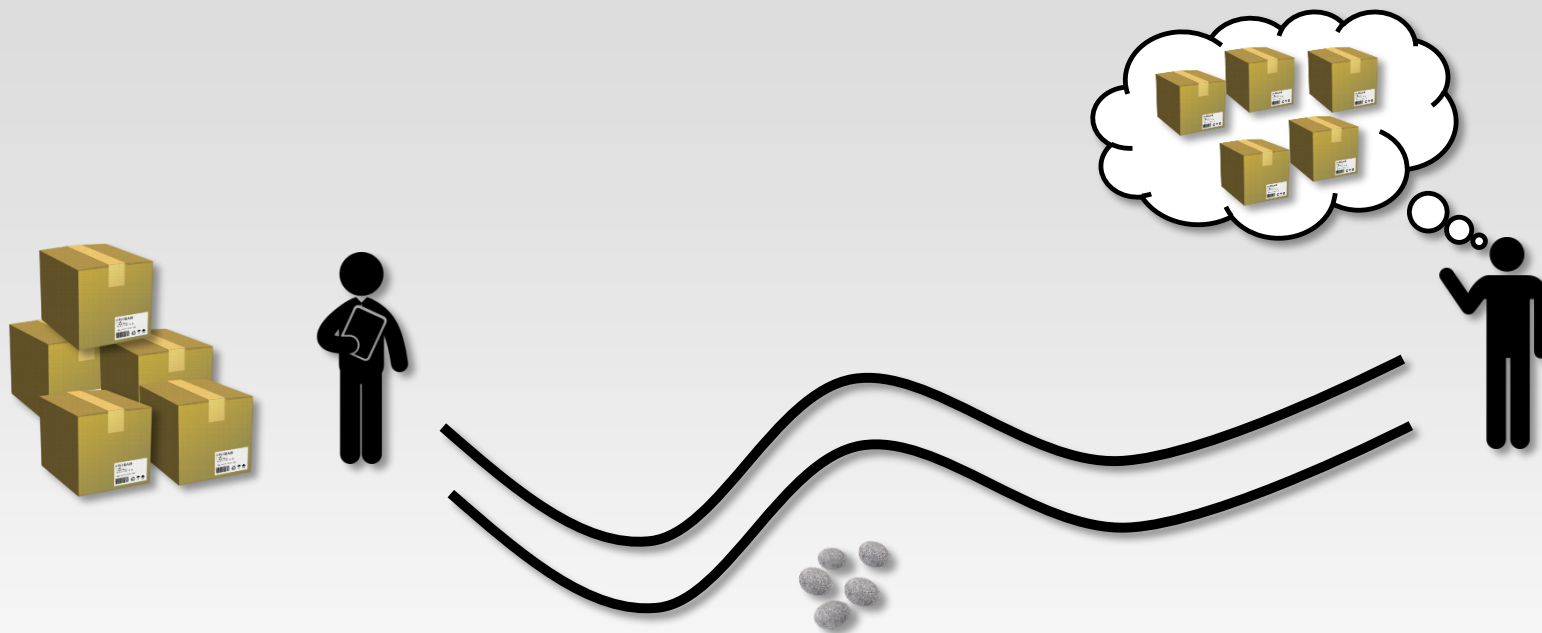
Prof. Rogério Esteves Salustiano  
Prof. Luiz Felipe Ramos Turci

- ✓ Bases Numéricas
- ✓ Representação Binária de Dados
  - ✓ Representação Numérica (Inteiro e Float)
  - ✓ Representação de Caracteres (Tabela ASCII)
  - ✓ Representação de Cores (Espaço RGB)
  - ✓ Representação de Data e Hora (*Epoch*)

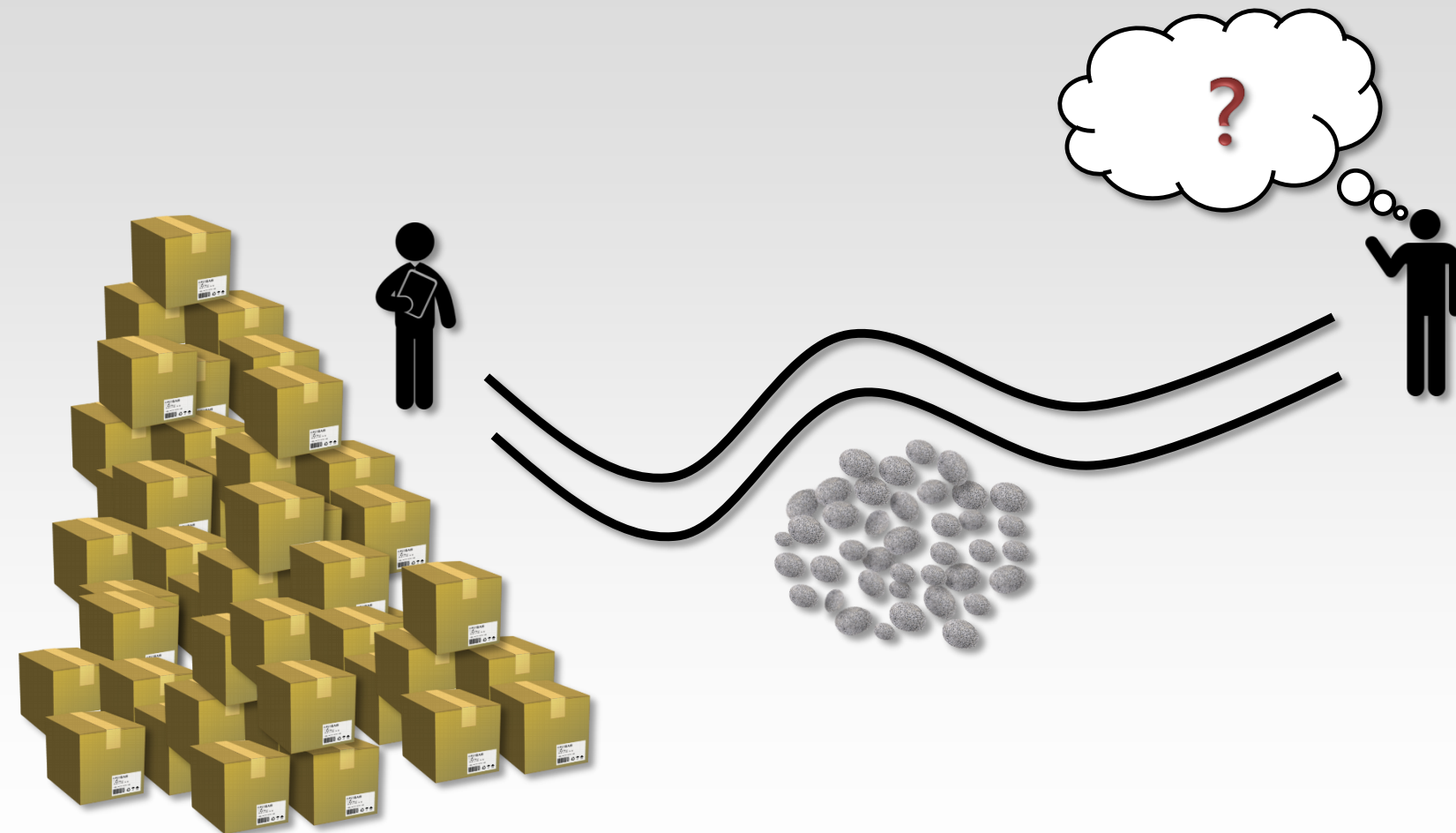
# *Como você realiza a contagem (quantificação) das coisas?*



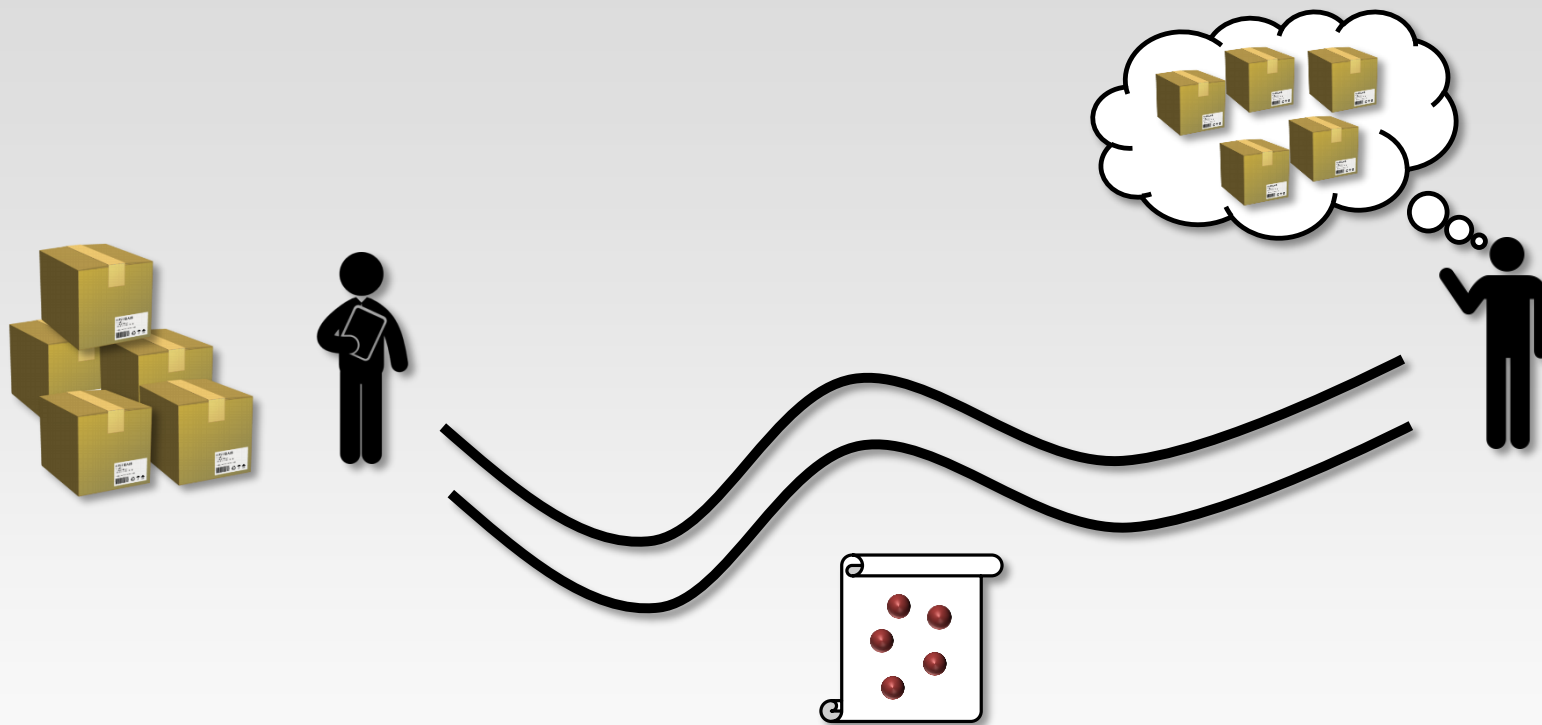
*Como transmitir o dado de uma certa quantidade?*



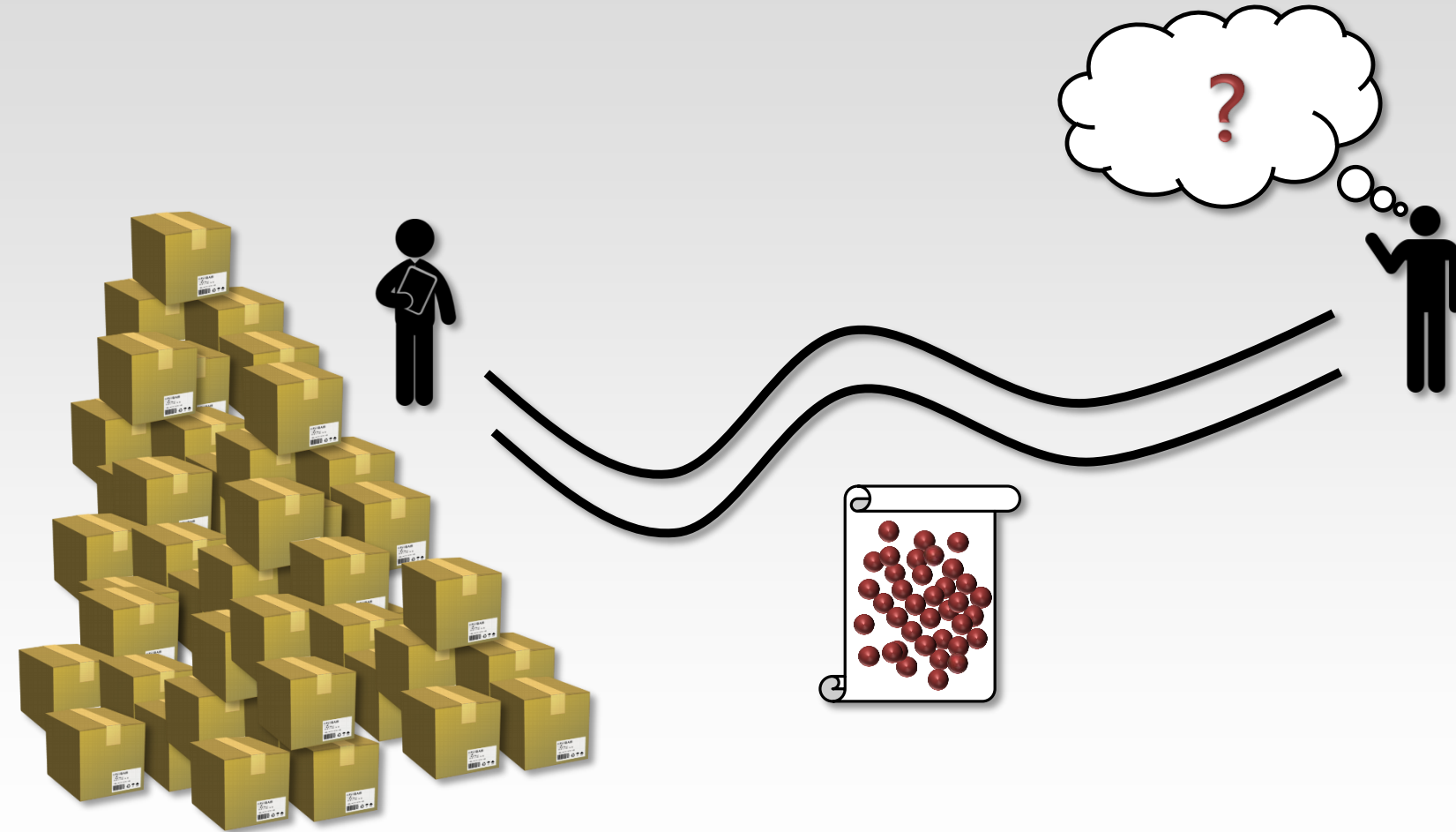
*Como transmitir o dado de uma certa quantidade?*



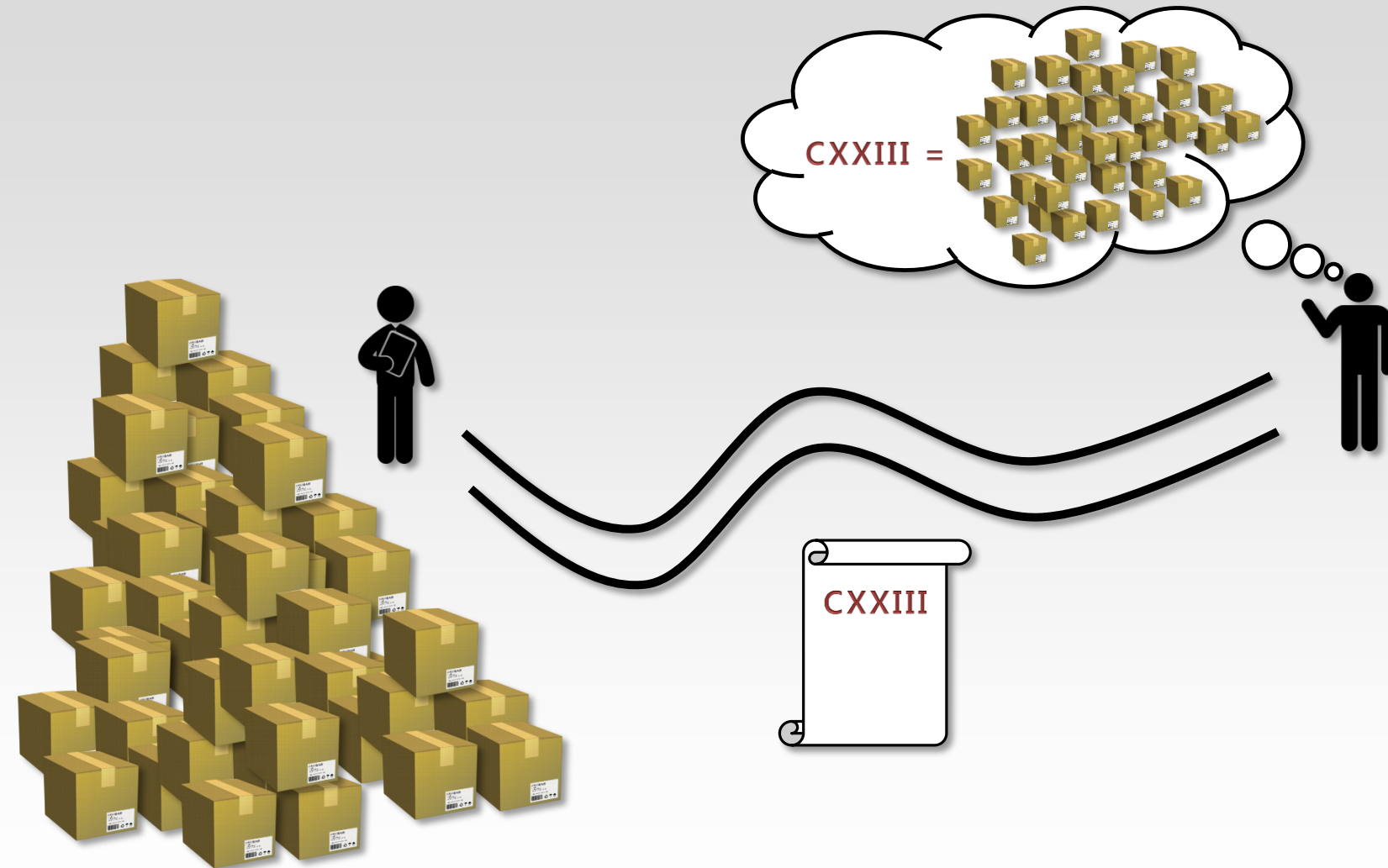
*Como transmitir o dado de uma certa quantidade?*



*Como transmitir o dado de uma certa quantidade?*



*Como transmitir o dado de uma certa quantidade?*



## Sistema de Base Numéricas



**Como surgiu a base 10?**

**Existem outras bases numéricas utilizadas ainda hoje que você conhece?**

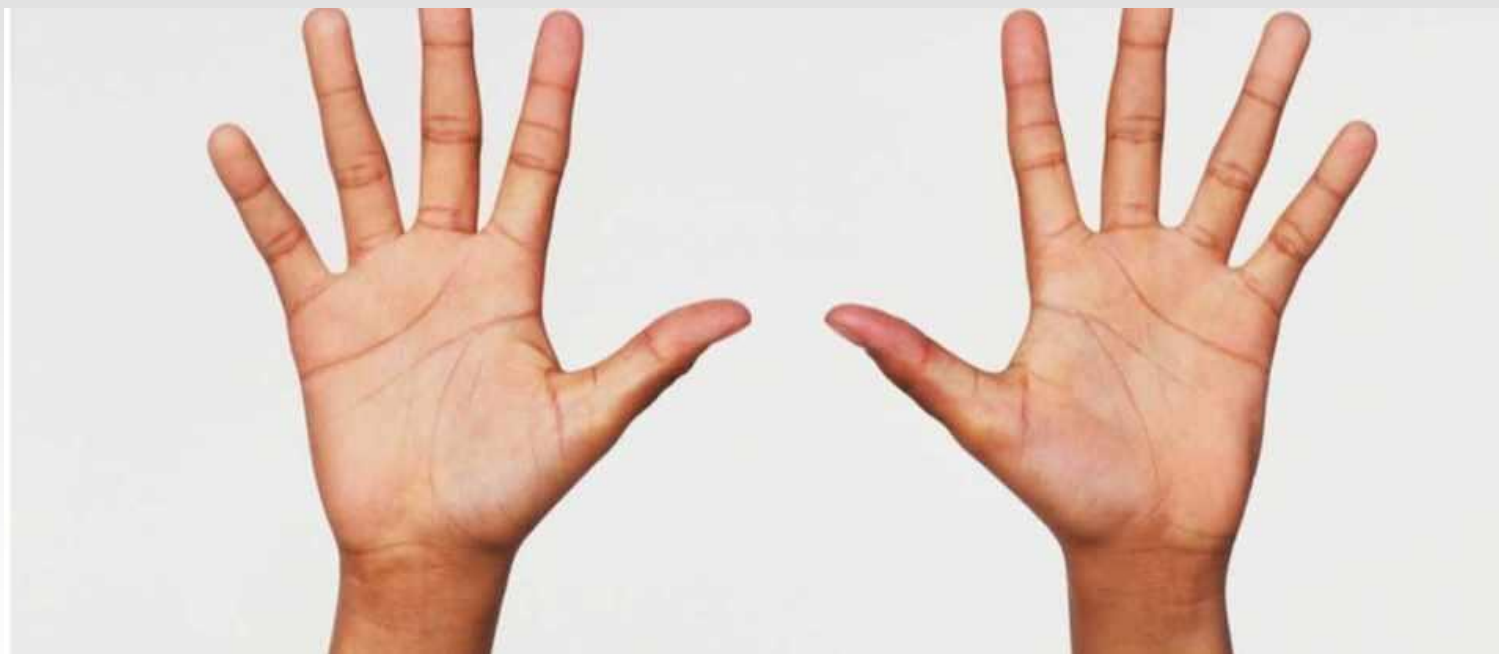
**Como surgiu a dúzia? Ela é uma base numérica?**

**Qual a base numérica utilizada pelo computador? Por que o computador a utiliza?**



## Sistema de Base Numéricas

Como surgiu a base 10?



## Sistema de Base Numéricas



**Existem outras bases numéricas utilizadas ainda hoje que você conhece?**

**Como surgiu a dúzia? Ela é uma base numérica?**

**Qual a base numérica utilizada pelo computador? Por que o computador a utiliza?**

## Base Duodecimal (12)

As vantagens desse sistema é que o número 12 tem mais divisores do que o 10. Enquanto que os divisores de 10 são 2 e 5, a base 12 tem os divisores 2, 3, 4 e 6.

Dessa maneira, um ano teria em meses um número igual a essa base, um dia teria em horas o dobro do valor da base, a hora teria em minutos, cinco vezes o valor da base, a medida do círculo seria trinta vezes o valor da base, etc.

Os povos primitivos dividiram o dia e a noite em 12 partes, pois para eles dia e noite eram entidades antagônicas e chamaram cada uma destas 12 partes de hora. Logo, dia e noite somam 24 horas, duas vezes a base 12.

## Base Duodecimal (12)

Um fato interessante é a possibilidade de se contar de 1 a 12 usando os dedos de uma única mão: basta não considerar o polegar e considerar cada uma das 3 falanges (ou articulações) dos 4 dedos opostos da mesma mão..

## Sistema de Base Numéricas



**Qual a base numérica utilizada pelo computador? Por que o computador a utiliza?**

## Sistema Binário

Leibniz (1646-1716), foi um dos primeiros defensores do sistema binário e propunha que todo pensamento racional se tornasse matemático e defendia uma espécie de linguagem ou escrita universal.

Para ele o número 1 representava Deus e o 0 (zero) corresponderia ao vazio.

Seus colegas contemporâneos ignoraram essas idéias e só foram retomadas no século XIX por George Boole que criou um sistema de lógica simbólica conhecido como Álgebra Booleana. Quase cem anos depois, o sistema binário e a álgebra desenvolvida por Boole tornou possível a criação do computador digital eletrônico.

## Sistema Binário

Os [computadores](#) digitais trabalham internamente com dois níveis de [tensão](#), pelo que o seu sistema de numeração natural é o sistema binário.<sup>[3]</sup>

Com efeito, num sistema simples como este é possível simplificar o cálculo, com o auxílio da [lógica booliana](#). Em computação, chama-se um dígito binário (0 ou 1) de [bit](#), que vem do [inglês](#) *Binary Digit*. Um agrupamento de 8 bits corresponde a um [byte](#) (*BinaryTerm*).

## Sistema Binário

Um processador é formado por milhares de blocos lógicos complexos, formados por portas lógicas básicas.

Em um [circuito digital](#) TTL (Transistor Transistor Logic), os dois níveis de tensão padronizados são 0V (zero volt) e 5V ([cinco](#) volts). Ao projetar um sistema digital, ao invés de trabalhar com níveis de tensão trabalha-se com níveis lógicos, então, no caso do circuito TTL, 0V será representado por “0” e 5V será representado por “1”.

Os níveis de tensão entre eles serão ignorados, ou seja, adotar-se-á uma faixa até a qual será considerado nível lógico zero, e a partir dela, nível lógico 1. Neste caso, de 0V a 2,5V temos “0”, e a partir daí até 5V temos “1”.



## Sistema de Base Numéricas

A base de um sistema de numeração é uma **certa quantidade de unidades** que deve constituir uma **unidade de ordem imediatamente superior**

## Algarismos utilizados na base 10 e quantidades relacionadas

$$\{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$$

0 →

5 → ●●●●●

1 → ●

6 → ●●●●●●

2 → ●●

7 → ●●●●●●●

3 → ●●●

8 → ●●●●●●●●

4 → ●●●●

9 → ●●●●●●●●●

Para a representação de quantidades maiores que a quantidade de símbolos da base, concatenam-se os símbolos:

 $9 + 1 = 10 \rightarrow$  símbolo **1** e símbolo **0**
 $9 + 2 = 11 \rightarrow$  símbolo **1** e símbolo **1**
 $53 \rightarrow$  símbolo **5** e símbolo **3**
 $126 \rightarrow$  símbolo **1** e símbolo **2** e símbolo **6**

## Representação de números na base 10 (unidade, dezena, centena, milhar, ...)

185 =

185 =

4032 =

4032 =

## Algarismos utilizados na base 10 e quantidades relacionadas

$$\{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$$

0 →

5 → ●●●●●

1 → ●

6 → ●●●●●●

2 → ●●

7 → ●●●●●●●

3 → ●●●

8 → ●●●●●●●●

4 → ●●●●

9 → ●●●●●●●●●

Para a representação de quantidades maiores que a quantidade de símbolos da base, concatenam-se os símbolos:

 $9 + 1 = 10 \rightarrow$  símbolo **1** e símbolo **0**
 $9 + 2 = 11 \rightarrow$  símbolo **1** e símbolo **1**
 $53 \rightarrow$  símbolo **5** e símbolo **3**
 $126 \rightarrow$  símbolo **1** e símbolo **2** e símbolo **6**

## Representação de números na base 10 (unidade, dezena, centena, milhar, ...)

$$185 = ( 1 \times 100 ) + ( 8 \times 10 ) + ( 5 \times 1 )$$

$$185 = ( 1 \times 10^2 ) + ( 8 \times 10^1 ) + ( 5 \times 10^0 )$$

$$4032 = ( 4 \times 1000 ) + ( 0 \times 100 ) + ( 3 \times 10 ) + ( 2 \times 1 )$$

$$4032 = ( 4 \times 10^3 ) + ( 0 \times 10^2 ) + ( 3 \times 10^1 ) + ( 2 \times 10^0 )$$

## Símbolos utilizados nas bases numéricas

Base 2 → { 0, 1 }

Base 3 → { 0, 1, 2 }

Base 4 → { 0, 1, 2, 3 }

Base 5 → { 0, 1, 2, 3, 4 }

Base 6 → { 0, 1, 2, 3, 4, 5 }

Base 7 → { 0, 1, 2, 3, 4, 5, 6 }

Base 8 → { 0, 1, 2, 3, 4, 5, 6, 7 }

Base 9 → { 0, 1, 2, 3, 4, 5, 6, 7, 8 }

Base 10 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Base 11 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A }

Base 12 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B }

Base 13 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C }

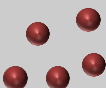
Base 14 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D }

Base 15 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E }

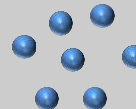
Base 16 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }

...

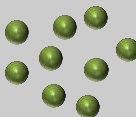
## Representação de quantidades em bases numéricas diversas



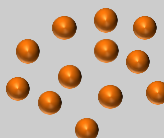
base 10 =  
base 6 =  
base 2 =



base 9 =  
base 12 =  
base 5 =



base 7 =  
base 3 =  
base 16 =



base 16 =  
base 12 =  
base 2 =

## Símbolos utilizados nas bases numéricas

Base 2 → { 0, 1 }

Base 3 → { 0, 1, 2 }

Base 4 → { 0, 1, 2, 3 }

Base 5 → { 0, 1, 2, 3, 4 }

Base 6 → { 0, 1, 2, 3, 4, 5 }

Base 7 → { 0, 1, 2, 3, 4, 5, 6 }

Base 8 → { 0, 1, 2, 3, 4, 5, 6, 7 }

Base 9 → { 0, 1, 2, 3, 4, 5, 6, 7, 8 }

Base 10 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Base 11 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A }

Base 12 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B }

Base 13 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C }

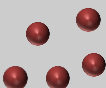
Base 14 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D }

Base 15 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E }

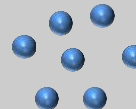
Base 16 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }

...

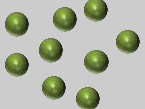
## Representação de quantidades em bases numéricas diversas



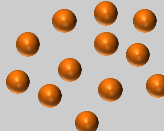
base 10 = 5  
base 6 = 5  
base 2 = 101



base 9 = 7  
base 12 = 7  
base 5 = 12



base 7 = 12  
base 3 = 100  
base 16 = 9



base 16 = C  
base 12 = 10  
base 2 = 1100

## Símbolos utilizados nas bases numéricas

Base 2 → { 0, 1 }

Base 3 → { 0, 1, 2 }

Base 4 → { 0, 1, 2, 3 }

Base 5 → { 0, 1, 2, 3, 4 }

Base 6 → { 0, 1, 2, 3, 4, 5 }

Base 7 → { 0, 1, 2, 3, 4, 5, 6 }

Base 8 → { 0, 1, 2, 3, 4, 5, 6, 7 }

Base 9 → { 0, 1, 2, 3, 4, 5, 6, 7, 8 }

Base 10 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

Base 11 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A }

Base 12 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B }

Base 13 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C }

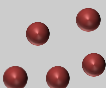
Base 14 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D }

Base 15 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E }

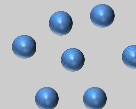
Base 16 → { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }

...

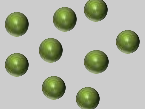
## Representação de quantidades em bases numéricas diversas



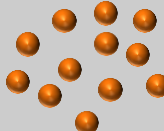
base 10 = 5  
base 6 = 5  
base 2 = 101



base 9 = 7  
base 12 = 7  
base 5 = 12



base 7 = 12  
base 3 = 100  
base 16 = 9



base 16 = C  
base 12 = 10  
base 2 = 1100

## Principais Bases Numéricas utilizadas em Computação

Decimal	Octal	Hexadecimal	Binário
00	00	0	0000
01	01	1	0001
02	02	2	0010
03	03	3	0011
04	04	4	0100
05	05	5	0101
06	06	6	0110
07	07	7	0111
08	10	8	1000
09	11	9	1001
10	12	A	1010
11	13	B	1011
12	14	C	1100
13	15	D	1101
14	16	E	1110
15	17	F	1111

- A **Base Binária** é utilizada pelo computador para armazenar, processar e transmitir dados
- A **Base Decimal** é utilizada pela maioria dos seres humanos
- A **Base Octal** foi muito utilizada para representar números binários de forma compacta
- A **Base Hexadecimal** é atualmente a base mais utilizada para representar números binários em uma forma mais compacta.
- As conversões **octal-binária/binária-octal** e **hexadecimal-binária/binária-hexadecimal** são mais simples de serem realizadas pelos computadores pois **são potências de 2** ( $8 = 2^3$  e  $16 = 2^4$ )

**“Existem 10 tipos de  
pessoas: as que sabem  
números binários e as que  
não sabem”**



## Representação de Números

### Seres Humanos

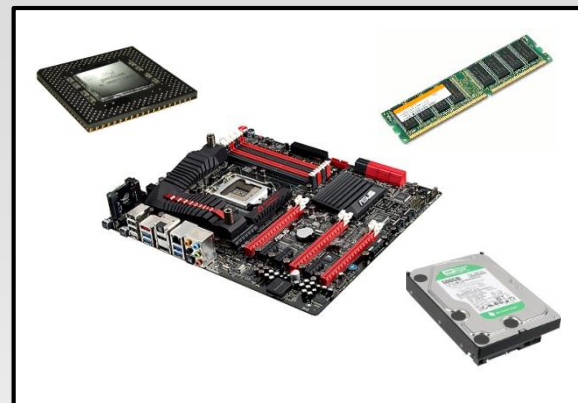


123,45 99,45 1043  
45,62 7859,42 8963  
558,85  $7,8 \times 10^6$

**base 10**

**Entrada de dados**  
**Leitura de dados**

### Computador



00110101011100101010  
01110101001011010001  
01010001010100011010

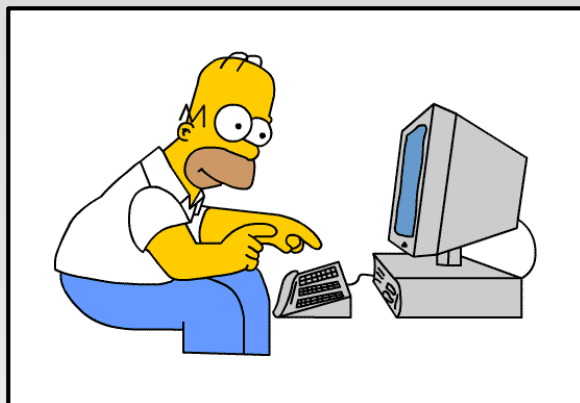
**base 2**

**Processamento de dados**  
**Armazenamento de dados**

**Conversão**

## Representação de Caracteres (Letras e Símbolos)

### Seres Humanos



Moe,  
I have to take care of Maggie  
until Marge arrives. After I go there  
for a beer with the guys.

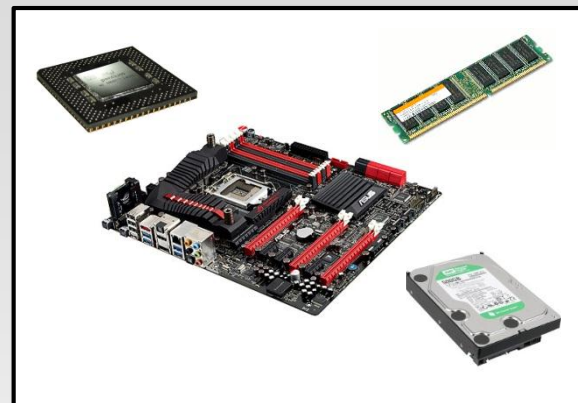
Homer

**caracteres**

**Entrada de dados**

**Leitura de dados**

### Computador



011100010110111110101011110011110  
111101010111100111100111000001111  
111010101111001111001110000011101  
110001011010101110001110100101110

**bits**

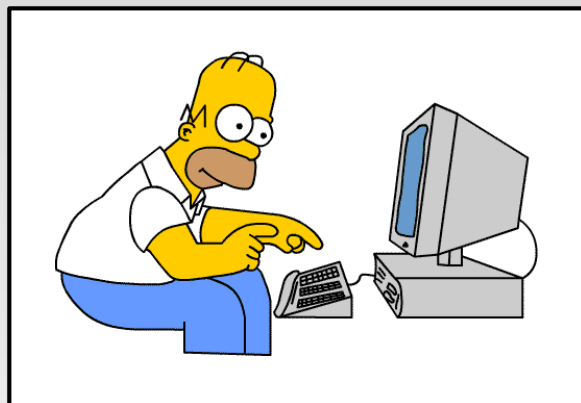
**Processamento de dados**

**Armazenamento de dados**

**Conversão**

## Representação de Caracteres (Letras e Símbolos)

### Seres Humanos

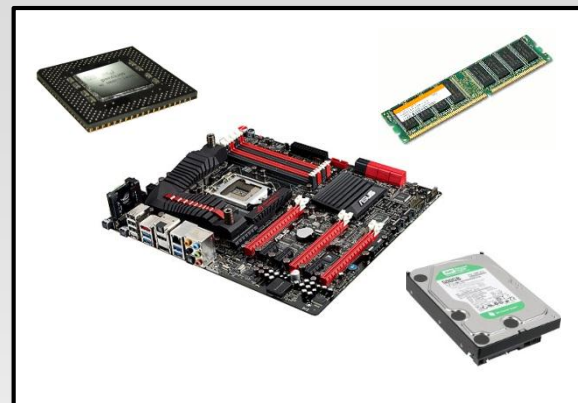


**Imagem/vídeo**

**Entrada de dados**

**Leitura de dados**

### Computador



```
0111000000001111000101010011110  
1000000111010101010000101011101  
11101000011111001001110000011101  
110001011010101110000001110001110  
00000000010101010000101111010101
```

**bits**

**Processamento de dados**

**Armazenamento de dados**

**Conversão**

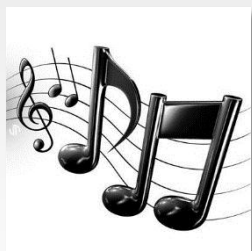
## Representação Binária no Computador

### Memória

Endereço	Conteúdo
000001	00101001
000010	11010111
000011	01100110
000100	00010010
...	...

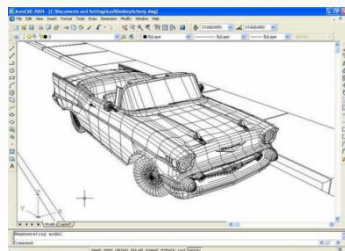
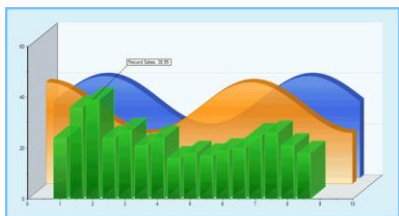
Os **bits** (conteúdo) armazenados de forma ordenada (endereço) representam algum **dado**, que uma vez transformado, tem um **significado** para o ser humano:

**Imagem, Som, Número, Valor Monetário, Texto, Planilha, Mapa, Gráfico, Desenho, etc.**



R\$ 1.251,85  
R\$ 8.512,03  
R\$ 4.886,24  
R\$ 9.934,84  
R\$ 5.215,65

*As armas e os barões assinalados  
Que da Ocidental praia Lusitana,  
Por mares nunca dantes navegados  
Passaram ainda além da Taprobana,  
Em perigos e guerras esforçados  
Mais do que prometia a força humana  
E entre gente remota edificaram  
Novo Reino, que tanto sublimaram;*



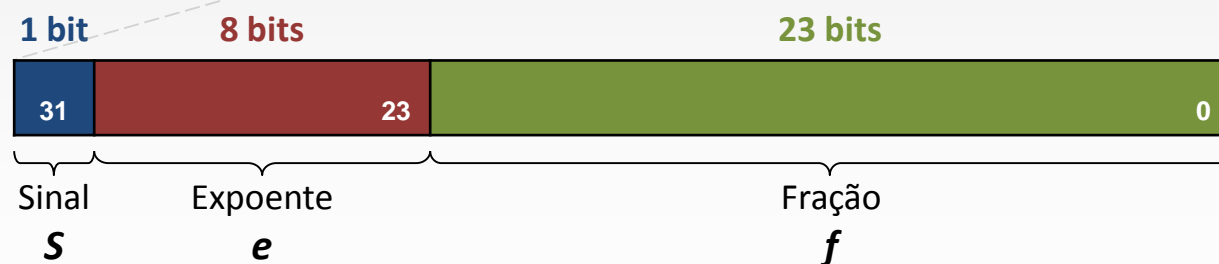
Memória de Cálculo de Atualização de Valores - Tabela de Índices de Cor Min - Ações Cond. 1ª Tx Sell					
Características Identificadoras da Coefrança Judicial					
Autor: Fernando Cardoso					
Réu: Luiz da Silva					
Vari	Nr. do Processo				
Juros até: 10/01/2003	0,00	Juros após: 10/01/2003	1,00		
Multa até: 10/01/2003	20,00	Multa após: 10/01/2003	2,00		
Data de Atualização	03/08/2010		Custas Processuais	40.434,18	
Juros a partir Citação	N		Honorários Execução		
Dados de Identificação e Caracterização dos Valores em Colocação					
Data Vencimento	Valor no Vencimento	Valor Atual	Multas	Juros	Sub Total Reais
20/02/1981	100.000,00	5.005,73	1.001,74	11.079,70	17.090,17
20/03/1981	150.000,00	7.054,58	1.410,91	15.572,43	24.037,92
20/04/1981	100.000,00	4.424,31	884,86	9.743,71	15.052,85
20/05/1981	100.000,00	4.172,86	834,77	9.171,79	14.180,43
Totais	200.000,00	20.661,50	4.132,28	45.567,62	70.361,40
Resumo da Totalização de Valores Atualizados					
Totalizadores		Valores em Reais			
Sub-Total Coefrança		70.361,40			
Honorários		0,00			
Custas Processuais		40.434,18			
TOTAL		110.795,58			
Cento e dez mil e setecentos e noventa e cinco reais e cinquenta e oito centavos					



## Representação de Ponto Flutuante [IEEE754] 32 bits

- ▶ O formato Ponto Flutuante, comumente utilizado na representação de números reais (com casas decimais), permite que as operações matemáticas executadas entre os números representados nesse formato sejam realizadas de forma mais rápida se comparado a outras formas de representação binária
- ▶ Apesar dessa rapidez nas operações, a representação em Ponto Flutuante, assim como ocorre em outras representações binárias de números fracionários, introduz inexatidões representativas. Tais inexatidões são intrínsecas ao processo de conversão entre bases (base 10 para base 2 e vice-versa) e à forma com que os valores são armazenados na representação binária de Ponto Flutuante

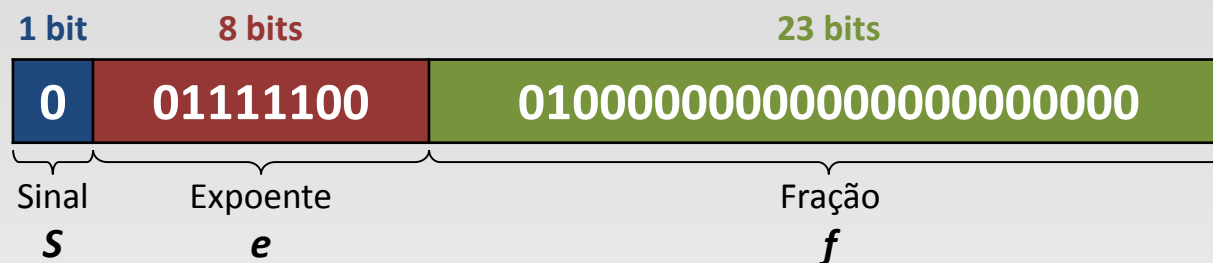
Memória	
Endereço	Conteúdo
000001	001100110011111001101110011010111
000010	10100010111111001101110101010101
000011	00000011001001001100110111010100
000100	01100011101100001110110011011001
...	...



$$v_{(10)} = (-1)^S \times 2^{(e-127)} \times (1 + f)$$

## Representação de Ponto Flutuante [IEEE754] 32 bits

### Exemplo



$$v_{(10)} = (-1)^S \times 2^{(e-127)} \times (1 + f)$$

$$S = 0$$

$$e = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 64 + 32 + 16 + 8 + 4 = 124$$

$$f = 0 \times 2^{-1} + 1 \times 2^{-2} = \frac{1}{4} = 0,25$$

$$v_{(10)} = (-1)^0 \times 2^{(124-127)} \times (1 + 0,25) = \frac{1,25}{8} \Rightarrow v_{(10)} = 0,15625$$

## Representação de Ponto Flutuante [IEEE754] 32 bits

### Valores Especiais:

<i>s</i>	<i>e</i>	<i>f</i>	
0	00000000	000000000000000000000000	→ +0 (Zero)
1	00000000	000000000000000000000000	→ -0 (Zero)
0	11111111	000000000000000000000000	→ $+\infty$ (Infinito Positivo)
1	11111111	000000000000000000000000	→ $-\infty$ (Infinito Negativo)
0	11111111	010000000000000000000000	→ +NaN (Não Número)
1	11111111	010000000000000000000000	→ -NaN (Não Número)



## Representação Numérica vs. Tipos de Dados

- Cada **linguagem de programação/arquitetura computacional** possui um conjunto de **tipos de dados** que permite o armazenamento e o processamento de números
- Cada **tipo de dado** possui uma **quantidade de bits** que permite armazenar um dado
- Um **tipo de dado** quando utilizado para representar **números** estabelece, pela **quantidade de bits que possui**, um **faixa (intervalo) de valores** que o tipo pode armazenar

### Exemplos de tipos de dados numéricos:

	Tipo	Tamanho	Faixa de valores
Números Inteiros	byte	8 bits	-128 a 127
	short	16 bits	-32.768 a 32.767
	int	32 bits	-2.147.483.648 a 2.147.483.647
	long	64 bits	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
Números Reais	float	32 bits	$1,1754943508222875 \times 10^{-38}$ a $3,4028234663852886 \times 10^{38}$ (números positivos e negativos)
	double	64 bits	$2,2250738585072014 \times 10^{-308}$ a $1,7976931348623157 \times 10^{308}$ (números positivos e negativos)

Notação de números reais utilizando potências de 10:  
(na maioria das linguagens de programação/programas)

$$2,56 \times 10^{-18} \quad \longrightarrow \quad 2.56E-18$$

## Operações aritméticas com Números Inteiros e Ponto Flutuante (com casa decimal)

Operação	Operador	Exemplos	
		Números Inteiros	Números Ponto Flutuante
Adição	+	$5 + 3 = 8$	$10.4 + 15.2 = 25.6$
Subtração	-	$13 - 10 = 3$	$23.2 - 0.4 = 22.8$
Multiplicação	*	$5 * 26 = 130$	$12.4 * 3.2 = 39.68$
Divisão	/	$131 / 9 = 14$	$131.0 / 9.0 = 14.56$
Módulo*	%	$131 \% 9 = 5$	não se aplica

Exemplo de divisão inteira:

$$\begin{array}{r}
 256 \quad \begin{array}{|l} 7 \\ \hline 36 \end{array} \\
 \underline{4} \quad 36 \\
 4 \quad 36
 \end{array}
 \quad
 \left.
 \begin{array}{l}
 256 / 7 = 36 \\
 256 \% 7 = 4
 \end{array}
 \right\}
 256 = 36 \times 7 + 4$$

\* A operação **módulo** também é conhecida como **resto da divisão inteira**

*Por que 1 Byte possui 8 bits?*



***Binário (8 bits): 00000000 a 11111111***

***Decimal (3 dígitos): 000 a 255***

Note que um **byte** nada tem de especial, é apenas um número binário de geralmente oito algarismos.

Sua adoção na informática deriva do motivo histórico do código [ASCII](#) haver adotado números de oito bits.

Além de razões meramente construtivas ou operacionais. Por exemplo: os códigos enviados a impressoras para controlar a impressão têm oito *bits*, os valores trocados pelos modems entre computadores também, assim como diversas outras operações elementares de intercâmbio de informações.

## Representação de Caracteres (Letras e Símbolos) na Memória

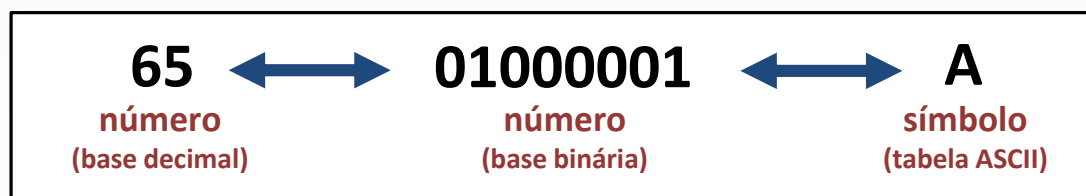
Dicionário Houaiss da Língua Portuguesa

**CARACTERE:** letra do alfabeto, algarismo, sinal de pontuação ou símbolo de qualquer natureza que pode ser introduzido em um computador pelo teclado ou por outro dispositivo de entrada, assim como exibido na tela ou em outro dispositivo de saída.

➤ Um caractere é armazenado na memória de um computador como um conjunto de bits. Esse conjunto de bits, portanto, representa uma letra, um algarismo numérico ou um símbolo.

Caractere	Valor Binário	
R	01010010	← (82)
o	01101111	← (111)
?	00111111	← (63)
/	00101111	← (47)
...	...	

➤ A Tabela **ASCII** (*American Standard Code for Information Interchange*) estabelece a relação entre o valor numérico (0 a 255) e o caractere correspondente. O valor numérico binário corresponde aos bits armazenados na memória e o caractere corresponde ao símbolo gráfico que é visualizado no monitor, impressora, etc. ou ao símbolo indicado na tecla pressionada no teclado



## Tabela ASCII (0 a 127)

dec.	hex.	oct.	ASCII	mnm.	dec.	hex.	oct.	ASCII	dec.	hex.	oct.	ASCII	dec.	hex.	oct.	ASCII
0	00	000		NUL	32	20	040		64	40	100	@	96	60	140	`
1	01	001	☺	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	☹	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	♥	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	♦	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	05	005	♣	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	♠	ACK	38	26	046	&	70	46	106	F	102	63	146	f
7	07	007		BEL	39	27	047	\	71	47	107	G	103	67	147	g
8	08	010		BS	40	28	050	(	72	48	110	H	104	68	150	h
9	09	011		TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	0A	012		LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	♂	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	♀	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015		CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	♪	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	⚙	SI	47	2F	057	/	79	4E	117	O	111	6F	157	o
16	10	020	▶	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	◀	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	↑	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	!!	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	¶	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	§	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	—	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	‡	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	↑	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	↓	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	→	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	←	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	L	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	↔	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	▲	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	▼	US	63	3F	077	?	95	5F	137	_	127	7F	177	␣

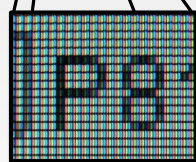
## Tabela ASCII (128 a 255)

dec.	hex.	oct.	ASCII	dec.	hex.	oct.	ASCII	dec.	hex.	oct.	ASCII	dec.	hex.	oct.	ASCII
128	80	200	Ç	160	A0	240	á	192	C0	300	Ł	224	E0	340	Ó
129	81	201	ü	161	A1	241	í	193	C1	301	ł	225	E1	341	ó
130	82	202	é	162	A2	242	ó	194	C2	302	┐	226	E2	342	Ô
131	83	203	â	163	A3	243	ú	195	C3	303	└	227	E3	343	Ò
132	84	204	ä	164	A4	244	ñ	196	C4	304	—	228	E4	344	õ
133	85	205	à	165	A5	245	Ñ	197	C5	305	†	229	E5	345	Õ
134	86	206	â	166	A6	246	ª	198	C6	306	ã	230	E6	346	µ
135	87	207	ç	167	A7	247	º	199	C7	307	Ã	231	E7	347	þ
136	88	210	ê	168	A8	250	¿	200	C8	310	ℓ	232	E8	350	ƒ
137	89	211	ë	169	A9	251	®	201	C9	311	ℝ	233	E9	351	ú
138	8A	212	è	170	AA	252	¬	202	CA	312	℥	234	EA	352	û
139	8B	213	ï	171	AB	253	½	203	CB	313	℞	235	EB	353	ù
140	8C	214	î	172	AC	254	¼	204	CC	314	℟	236	EC	354	ý
141	8D	215	ì	173	AD	255	¡	205	CD	315	=	237	ED	355	Ý
142	8E	216	Ä	174	AE	256	«	206	CE	316	‡	238	EE	356	—
143	8F	217	Å	175	AF	257	»	207	CF	317	¤	239	EF	357	´
144	90	220	É	176	B0	260	☼	208	D0	320	ø	240	F0	360	
145	91	221	æ	177	B1	261	☼	209	D1	321	Ð	241	F1	361	±
146	92	222	Æ	178	B2	262	☼	210	D2	322	Ê	242	F2	362	=
147	93	223	ô	179	B3	263		211	D3	323	Ë	243	F3	363	¾
148	94	224	ö	180	B4	264	┘	212	D4	324	È	244	F4	364	℥
149	95	225	ò	181	B5	265	Á	213	D5	325	ı	245	F5	365	§
150	96	226	û	182	B6	266	Â	214	D6	326	Í	246	F6	366	÷
151	97	227	ù	183	B7	267	À	215	D7	327	Î	247	F7	367	,
152	98	230	ÿ	184	B8	270	©	216	D8	330	İ	248	F8	370	°
153	99	231	Ö	185	B9	271	¶	217	D9	331	Ј	249	F9	371	..
154	9A	232	Ü	186	BA	272		218	DA	332	Г	250	FA	372	.
155	9B	233	ø	187	BB	273	¶	219	DB	333	■	251	FB	373	¹
156	9C	234	£	188	BC	274	¶	220	DC	334	■	252	FC	374	³
157	9D	235	∅	189	BD	275	¢	212	DE	335	¡	253	FD	375	²
158	9E	236	×	190	BE	276	¥	222	DD	336	İ	254	FE	376	■
159	9F	237	f	191	BF	277	γ	223	DE	337	■	255	FF	377	

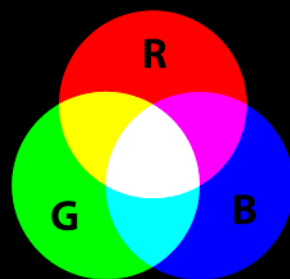
## Espaço de cores

## Como representar uma cor utilizando bits?

Um espaço de cor é um modelo matemático que descreve cada uma das cores que o ser humano pode ver a partir de certas combinações de valores.



Pixel RGB



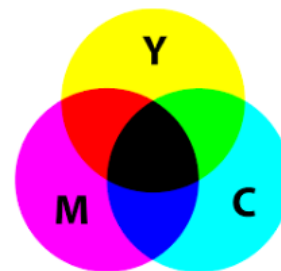
**RGB** (RED-GREEN-BLUE)

*cores aditivas*

Vermelho

Verde

Azul



**CMY** (CYAN-MAGENTA-YELLOW)

*cores subtrativas*

Ciano

Magenta

Amarelo



**CMYK**

(CYAN-MAGENTA-YELLOW-KEY)

*Ciano, Magenta,  
Amarelo e Preto*

**Outros espaços de cores:** HSB (*Hue Saturation Brightness*), HSL (*Hue Saturation Luminance*), Munsell, PMS (*Pantone Matching System*), NCS (*Natural Color System*)



## Espaço de cor RGB

- Cada uma das três cores primárias (vermelho, verde e azul) é representada por 1 Byte (8 bits), ou seja, valores de 0 até 255 para cada cor primária
- A mistura das três cores primárias produz uma cor sensível ao olho humano

**(R,G,B)**

**0-255 0-255 0-255**


**8 bits + 8 bits + 8 bits = 24 bits**

24 bits → 16.777.216 combinações

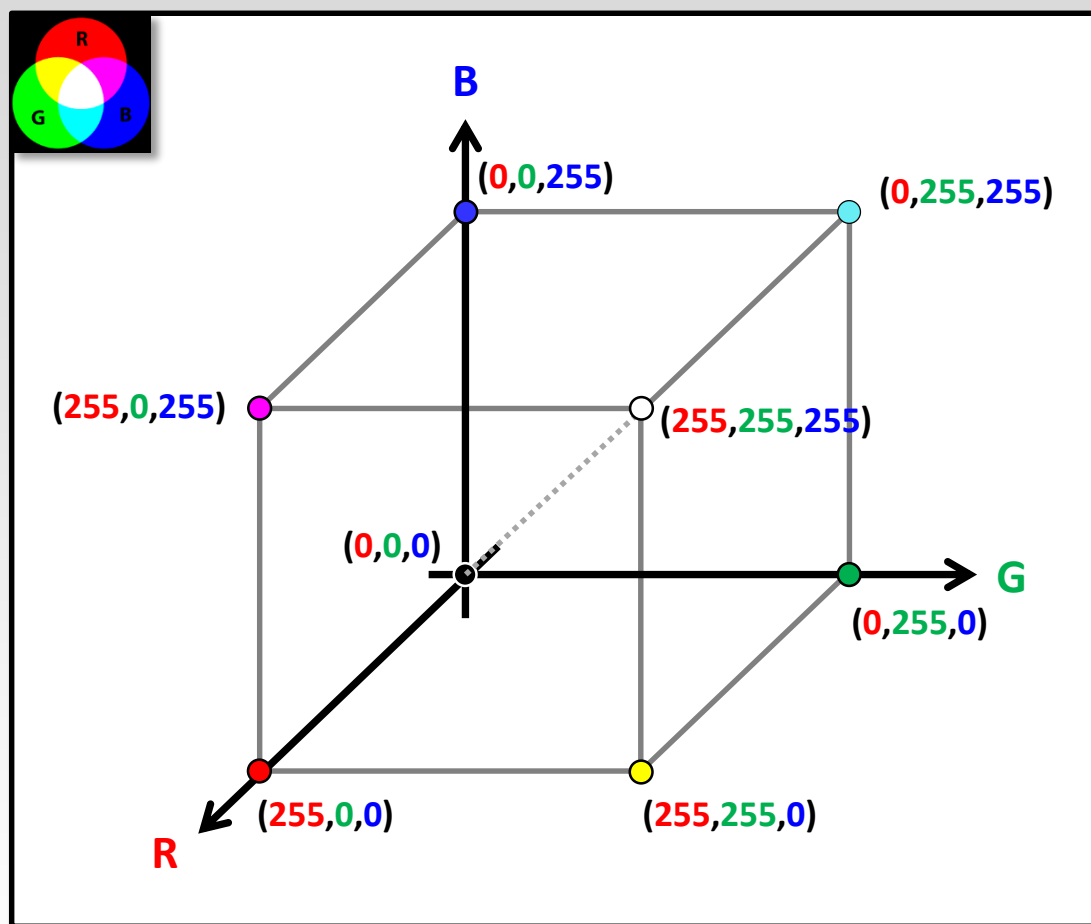
≈ 16,7 milhões de cores!!

*Exemplos:*

 → **(148,48,220)** → 10010100 00110000 11011100

 → **(243,178,49)** → 11110011 10110010 00110001

## Como representar uma cor utilizando bits?



**Curiosidade:** Nossos olhos só conseguem diferenciar apenas 30 tons de cinza!!

## Problema

Um monitor de computador possui uma resolução **full HD** (1080 pixels na vertical por 1920 pixels na horizontal) e uma profundidade de cor de **32 bits** (8 bits para R, 8 bits para G, 8 bits para B e 8 bits para o canal alfa).

**Quantos Bytes de memória são necessários para armazenar todos os pixels exibidos na tela desse monitor?**

$$\begin{array}{c} \text{Quantidade de pixels} \\ \text{na horizontal} \\ \underbrace{1080} \times \underbrace{1920} \times \underbrace{32 \text{ bits}} = 66355200 \text{ bits} = 8294400 \text{ Bytes} \approx 8.3 \text{ MBytes} \\ \underbrace{\hspace{1.5cm}} \quad \underbrace{\hspace{1.5cm}} \quad \underbrace{\hspace{1.5cm}} \\ \text{Quantidade de pixels} \quad \text{Profundidade} \\ \text{na vertical} \quad \text{de cor (bits)} \end{array}$$

**Qual a frequência de processamento mínima (em bps) necessária para que a imagem do monitor possua uma taxa de atualização de 60Hz para todos os pixels?**

$$\begin{array}{c} \text{Frequência de atualização} \\ \text{da tela (60Hz = 60/s)} \\ \underbrace{66355200 \text{ bits}} \times \underbrace{60 \text{ Hz}} = 3981312000 \text{ bits/s} \approx 4\text{Gbps} \\ \underbrace{\hspace{2.5cm}} \\ \text{Quantidade bits necessários} \\ \text{para gerar a imagem da tela} \end{array}$$

## Representação do tempo *Como representar uma data/hora utilizando bits?*

- **Epoch** é um instante no tempo escolhido como origem para a marcação de eventos. As unidades de tempo são contadas a partir da **epoch** de forma que as datas e o tempo (horas, minutos, segundos) podem ser determinadas com exatidão
- Uma grande quantidade de sistemas e linguagens de programação (Unix, Java, C, C++, PHP, Python, etc.) consideram a data de **1 de janeiro de 1970** como a **epoch**. Essa data também é conhecida como **UNIX time** (ou **POSIX time**).
- A **cada segundo** o sistema incrementa um **número inteiro** em uma unidade. Este número inteiro é inicialmente ajustado com alguma referência externa (data de um calendário e hora de um relógio) ou sincronizado com outros sistemas. O tempo, portanto, é determinado pela quantidade de segundos que se passaram desde a **epoch**. Esse número inteiro incrementa 86.400 vezes por dia desde a **epoch**

Data/Hora	Quantidade de segundos desde a <i>epoch</i> Unix	
	Decimal	Binário* (32 bits)
01/01/1970 00:00:00 UTC	0	0000 0000 0000 0000 0000 0000 0000 0000
12/10/1931 22:15:00 UTC	-1.206.150.300	1011 1000 0001 1011 1001 1011 0110 0100
09/09/2001 01:46:40 UTC	1.000.000.000	0011 1011 1001 1010 1100 1010 0000 0000
19/01/2038 03:14:08 UTC	-2.147.483.648	1000 0000 0000 0000 0000 0000 0000 0000

**Problema do ano 2038**  
(19/01/2038 04:14:08)

\* Notação Binária em Complemento de 1