



INSTITUTO FEDERAL  
RIO GRANDE DO SUL

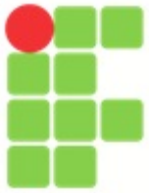
---

# Funcionamento do Computador

---

Profa. Raquel M. Barbosa

([raq.mbarbosa@gmail.com](mailto:raq.mbarbosa@gmail.com))

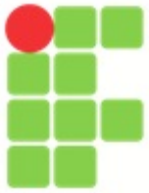


# Funcionamento de um Computador

---

- Podemos pensar em um computador como uma máquina capaz de realizar vários tipos de *processamento*.
- O processamento é uma operação ou transformação sobre dados (entrada) que gera um resultado (saída) visível ao ser humano que o utiliza.

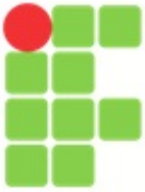




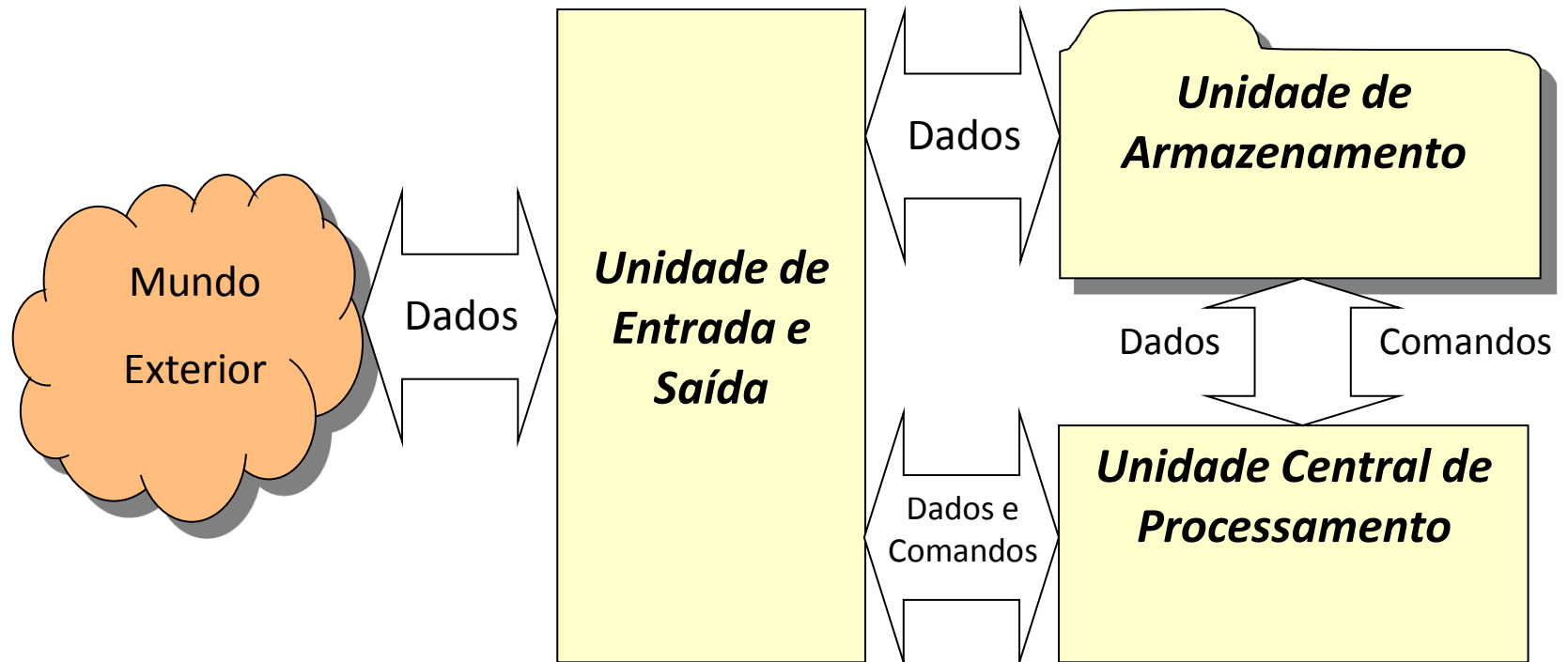
# Funcionamento de um Computador

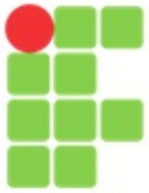
---

- Em computadores
  - Dados de entrada
    - Números (1, -3, 5000)
    - Sequências de caracteres (“IFRS”, “Rio Grande do Sul”)
    - Outros (operação a ser feita em uma calculadora)
  - Processamento é o cálculo do resultado da operação
  - Dado de saída
    - É um número (o resultado da operação, após o cálculo/processamento)
    - Sequências de caracteres



# Componentes do Hardware





# Unidade de E/S

- Comunicação com o mundo exterior
- Obtenção de dados e instruções
- Saída de resultados

## Entrada



Microfone(som)



Scanner (imagens)



Mouse (posições)



Teclado (texto)

## Saída



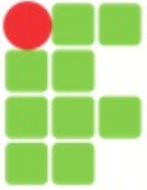
Impressora



Monitor



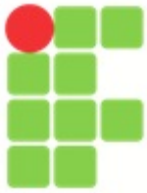
Caixas de som



# Unidade de Armazenamento

---

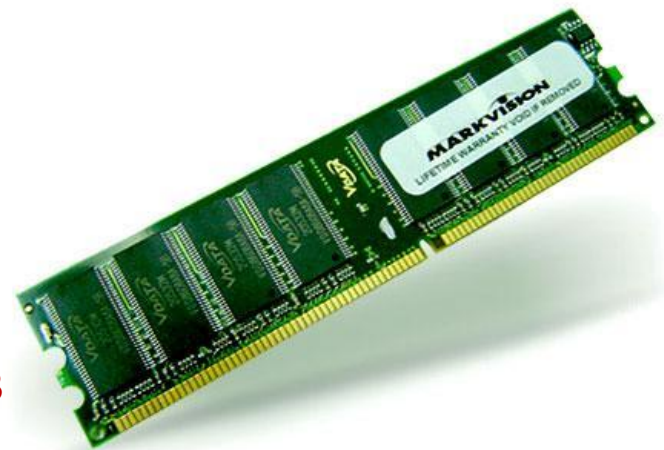
- Armazenar dados e instruções dos programas
- Bloco de anotações do computador
- Dois tipos de dispositivos
  - Voláteis
  - Não voláteis



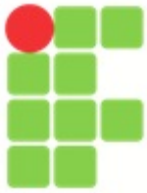
# Armazenamento Volátil

---

- Dispositivos que têm seu conteúdo apagado quando deixam de receber energia elétrica
- Usados como memória temporária durante a execução de programas, por ser de acesso mais rápido
- Exemplo
  - Memória RAM



Pente de memória RAM DDR3 de 1GB

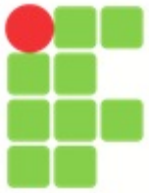


# Armazenamento Não-Volátil

- Mantém seu conteúdo, mesmo sem alimentação elétrica
- Usados para armazenamento de dados e instruções a longo prazo
- Exemplos
  - Disco Rígido e Memórias ROM



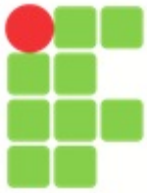




# Unidade Central de Processamento

---

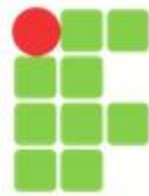
- *CPU – Central Processing Unit*
- Lê, interpreta e executa as instruções dos programas
  - Para isso gerencia outros dispositivos de hardware
- Formada pelo microprocessador
- Cuidado para não confundir com o gabinete dos microcomputadores!



# Microprocessador

- Formado por milhões de transistores organizados em *portas lógicas*
- A organização das portas lógicas especifica o que o microprocessador é capaz de fazer



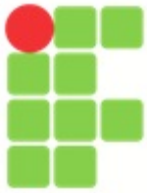


INSTITUTO FEDERAL  
RIO GRANDE DO SUL

---

# Como dados e instruções são processados pela CPU?

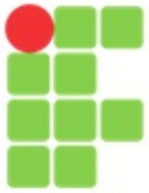
---



# Informação x Dado

---

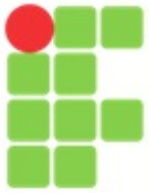
- **Informação** é o resultado do processamento, manipulação e organização de dados de tal forma que represente um acréscimo ao conhecimento da pessoa que a recebe” (Wikipedia).
- Podemos dizer que **dado** é tudo aquilo que pode ser processado, ou seja:
  - números, medições, textos
  - valores lógicos (verdadeiro e falso)



# Informação x Dado

---

- Por exemplo, o número “20” é um dado, pois pode ser processado (em um cálculo, por exemplo) mas não significa nada isoladamente.
- Mas se dissermos “*Hoje a temperatura máxima atingiu 20 °C*” organizamos os dados de forma a obter informação, pois agora existe um significado.

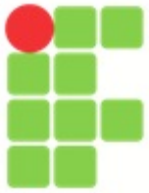


# Como se comunicar com o Computador???

---

- Computador não sabe falar português
- Linguagem de Máquina
  - Tudo é representado na forma de números inteiros
- Cada instrução diferente é representada por um número diferente (soma, entrada)
- Cada tipo de processador possui sua própria linguagem de máquina

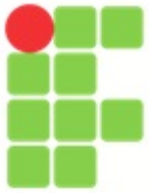




# Dados

---

- Também são representados por números inteiros
- E as imagens, sons, textos e números decimais???
- Imagens: são divididas em pontos chamados *pixels*
  - Cada pixel contém uma cor representada por um ou três números inteiros
- Sons: amostras da onda sonora (amplitude, frequência, etc) são representadas por números

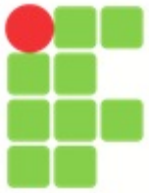


# Números Reais

---

- São representados como um conjunto de dois inteiros
- Para se chegar a esses inteiros se representa o número como uma multiplicação com uma potência de dez
  - Exemplos
    - $0,5 = 5 \times 10^{-1}$
    - $12,34 = 1234 \times 10^{-2}$
  - Os números usados são aqueles que multiplicam a potência e o expoente (Ex: 0,5 seria representado pelos números 5 e -1)

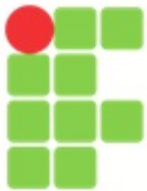




# Textos

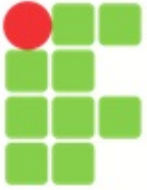
---

- Textos são formados por vários símbolos chamados *caracteres*
- Cada caractere é representado por um número inteiro, de acordo com tabelas de codificação
  - Uma das mais conhecidas é a tabela ASCII



# Parte da Tabela ASCII

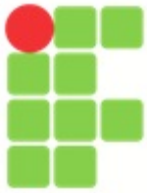
Caractere	Código	Caractere	Código	Caractere	Código
(branco)	32	, (vírgula)	44	8	56
!	33	-	45	9	57
“	34	. (ponto)	46	A	65
#	35	/	47	B	66
\$	36	0	48	C	67
%	37	1	49	D	68
&	38	2	50	E	69
‘	39	3	51	a	97
(	40	4	52	b	98
)	41	5	53	c	99
*	42	6	54	d	100
+	43	7	55	e	101



# Porque Números inteiros?

---

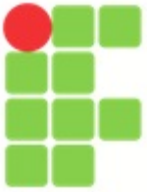
- Números inteiros podem ser facilmente representados na memória RAM do computador
  - Isso se deve ao fato de como a memória é organizada



# Organização da Memória

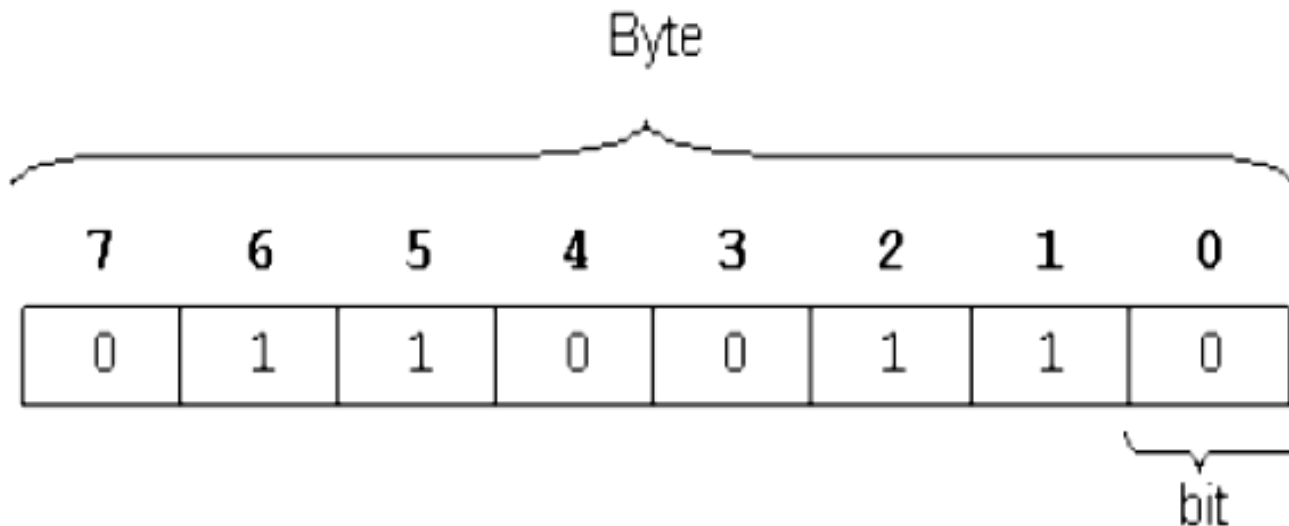
Célula {

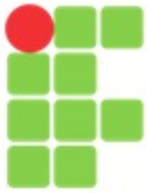
Endereço	Conteúdo da Célula
0	XXXXXXXX
1	XXXXXXXX
2	XXXXXXXX
3	XXXXXXXX
4	XXXXXXXX
5	XXXXXXXX
⋮	⋮



# Células de Memória

- Cada célula é conhecida como *byte*
- Cada byte é formado por oito *bits*

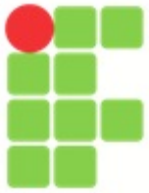




# Bytes na Memória

---

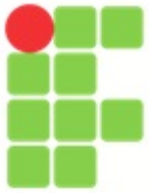
Endereço	Conteúdo da Célula							
0	0	0	0	0	1	1	0	0
1	1	1	1	1	1	1	0	0
2	0	1	1	0	0	1	1	0
3	0	1	0	1	0	1	0	1
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1
⋮	⋮							



# Bit

---

- Pode assumir dois valores
  - Zero (desligado) ou um (ligado)
- Por quê?
  - Representam a variação de uma grandeza
  - Exemplo: Tensão Elétrica
    - 0 – 2 volts = Zero
    - 4 a 5 volts = Um

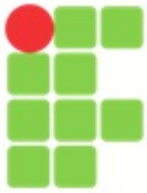


# Palavra

---

- A menor quantidade de dados que se consegue trabalhar é um **byte**
- Computadores atuais permitem trabalhar com pequenas quantidades de bytes de uma única vez, chamadas de *palavras*
- Hoje em dia é comum encontrarmos computadores com palavras de 8 bytes (64 bits)



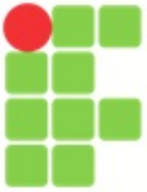


# Números nos bytes

---

- Como escrever um número em um byte se ele aceita só zeros e uns?
  - Utilizando o Sistema Binário

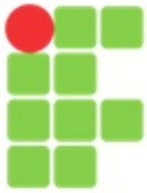




# Representação de Números

---

- Existem formas diferentes de dizer a mesma coisa
- Existem formas diferentes de representar quantidades
  - Exemplo:  $10 = X = \text{dez}$
- Nós comumente utilizamos o *sistema decimal* para representar números



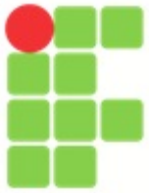
# Sistema Decimal

---

- Utiliza dez algarismos para representar qualquer número
- As posições dos algarismos nos números indicam seu valor

$$5_2 2_1 7_0 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

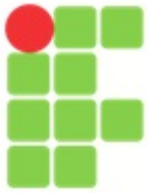
$$1_3 2_2 3_1 4_0 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$



# Outros Sistemas

---

- Podemos utilizar outras quantidades de algarismos para representar números
- A isso se dá o nome de *sistema de numeração*
- Cada sistema de numeração é identificado por uma base, que indica o número de algarismos utilizados
  - Sistema decimal = base 10
- Para obter o valor de um número em uma base qualquer basta multiplicar cada dígito pela base elevada ao número da posição do dígito



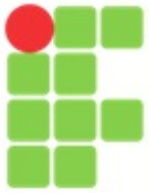
# Sistema Binário

---

- Utiliza apenas dois dígitos para representar números (o zero e o um)
  - Que coincidência, é exatamente o que cabe em um bit!!!

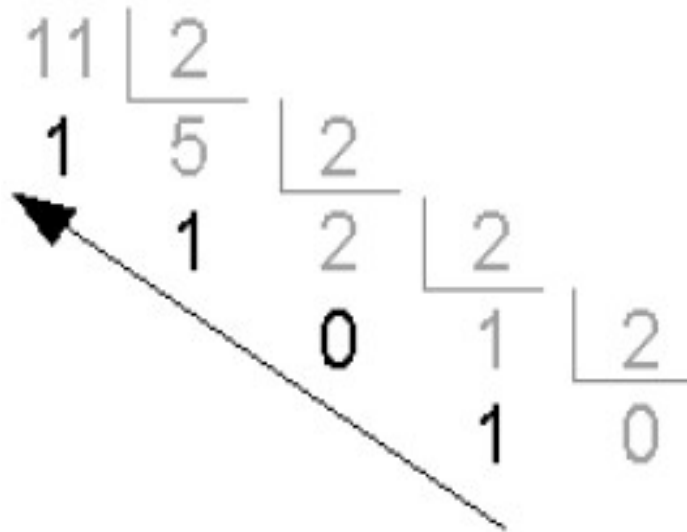
$$1_1 1_0 (\text{base } 2) = 1 \times 2^1 + 1 \times 2^0 = 3$$

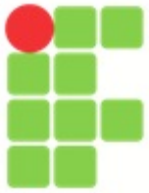
$$1_2 0_1 1_0 (\text{base } 2) = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$$



# Decimal-> Binário

- Caminho inverso dos restos das divisões sucessivas
- Por exemplo, 11 em binário é 1011 pois

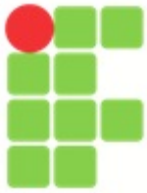




# Exemplos

---

Decimal (Base 10)	Binário (Base 2)
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010



# Números na Memória

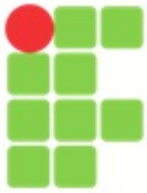
- É possível escrever em cada byte um número de até 8 dígitos binários

Qual o maior número possível de representar com 8 dígitos binários?



- Para números maiores é necessário usar mais bytes



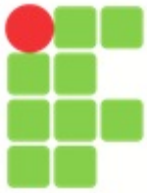


# Número 1234 na Memória

- 1234 em binário é 10011010010
- Na memória ocupa dois

Endereço	Conteúdo da Célula							
⋮	⋮							
1000	0	0	0	0	0	1	0	0
1001	1	1	0	1	0	0	1	0
⋮	⋮							

Primeiros três  
dígitos do número  
Últimos oito  
dígitos do número



# BANANA na Memória

Endereço	Conteúdo da Célula							
⋮	⋮							
1000	0	1	0	0	0	0	1	0
1001	0	1	0	0	0	0	0	1
1002	0	1	0	0	1	1	1	0
1003	0	1	0	0	0	0	0	1
1004	0	1	0	0	1	1	1	0
1005	0	1	0	0	0	0	0	1
⋮	⋮							

B (66)

A (65)

N (78)

A (65)

N (78)

A (65)