

Computadores, Linguagens e Algoritmos

Aula 1

Conceitos básicos

1. O que é um computador?
2. Dados e informação
3. Componentes de um computador
4. Hardware x software
5. Algoritmo
6. Linguagens: baixo nível x alto nível

O que é um computador

É um equipamento eletrônico que opera sob o controle de instruções armazenadas em sua unidade de memória, que pode

- * aceitar dados (entrada),
- * processar dados (fazendo operações aritméticas e lógicas),
- * produzir informação (saída), e
- * armazenar os resultados para uso futuro.

O que é um computador

É um equipamento eletrônico que opera sob o controle de instruções armazenadas em sua unidade de memória, que pode

- * aceitar dados (entrada),
- * processar dados (fazendo operações aritméticas e lógicas),
- * produzir informação (saída), e
- * armazenar os resultados para uso futuro.

O que é um computador



Cray-1

Supercomputador de grande sucesso-1976

Dados e informação

* Dados

- Figuras e símbolos (números, palavras, imagens, vídeos, som)

* Informação

- Dados organizados, com significado, e úteis

Componentes de um computador

* Dispositivos de entrada

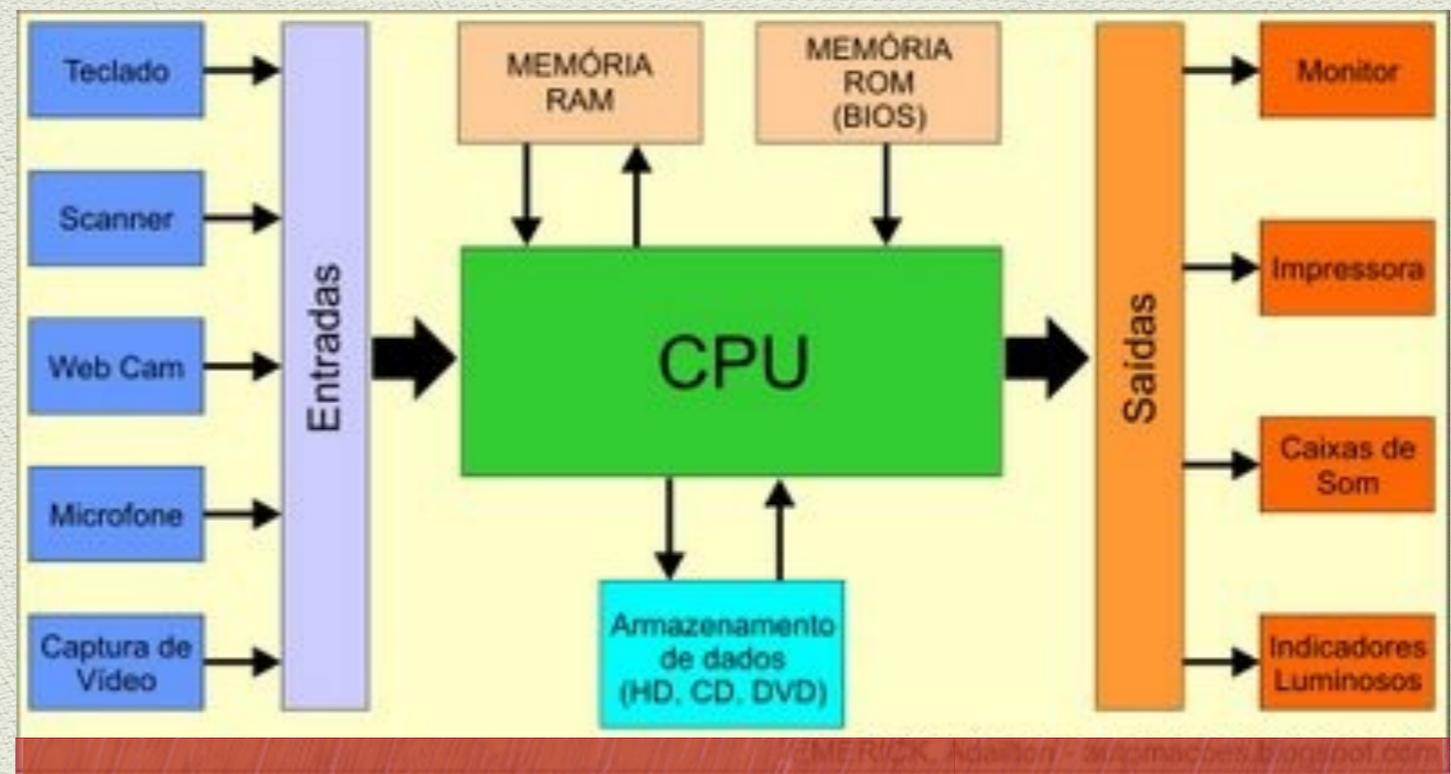
* Processador

CPU (Central Processing Unity) - "cérebro do computador"

* Memória

- Principal: RAM e ROM
- Memória secundária

* Dispositivos de saída

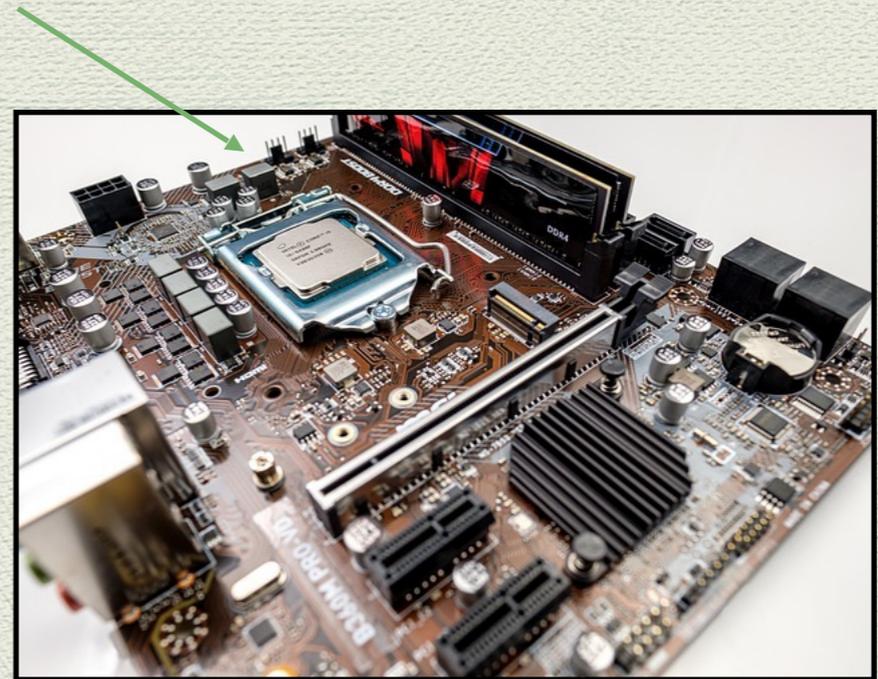


Componentes de um computador

* Processador

- CPU (Central Processing Unity) -
"cérebro do computador"

- Fica acoplado na placa-mãe (*motherboard*)
- Realiza as instruções dadas ao computador
- Faz operações aritméticas, lógicas, ... (milhões por segundo)
- Tipos: Intel, AMD, Motorola, Pentium,...
- Velocidade: medida em Hertz (número de ciclos por segundo)
Megahertz, Gigahertz (= 1 bilhão de Hz)



Componentes de um computador

* Memória

Dispositivos para armazenar dados e instruções

- **Memória principal**

- Memória RAM (Random Access Memory)

- volátil

- Memória ROM (Read Only Memory)

- não-volátil, não pode ser mudada

- Cache (área de armazenamento temporário)

- p/ informações que são acessadas frequentemente

- **Memória secundária**

- HD (Hard Disk), SSD (Solid State Drive), *pen drives*, DVD, ...

- não-volátil (HD: magnético 1 TB) (SSD: CI semicond. 120 GB)



Read Only Memory

Hardware x Software

* Hardware

- Componentes físicos do computador

Ex: CPU, monitor, HD, teclado, mouse, modem, cabos,...

* Software

- Conjunto de instruções (programa) para o computador executar
- Projetado para ajudar o usuário a realizar tarefas específicas num computador
- Tipos: software de sistemas e de aplicações

Hardware x Software

* Software

Software de sistemas

- Programas para prover uma plataforma para outros software.
— Ex: Sistema Operacional (SO): gerencia a memória e operações sobre arquivos (abrir, salvar, remover, renomear), controla dispositivos periféricos, monitora o desempenho do sistema, ...
Exemplos: Windows, MacOS, Linux, UNIX, ...
— Ex: Software de automação industrial.

Software de aplicações

- Ex: processadores de texto (Word, LaTeX,...), visualizador de arquivo, planilhas, editor de fotos/vídeos, navegador web, ...

Algoritmo

* O que é ?

É uma sequência finita de passos —ações executáveis— que visam resolver um dado problema, e que tem as seguintes características:

- seus passos devem estar bem definidos (sem ambiguidade);
- seus passos devem ser factíveis;
- sua execução deve ser efetiva (deve resolver o problema para o qual foi projetado).

Algoritmo x Programa de computador

Um programa de computador é uma sequência de instruções
-- numa determinada linguagem --
que especifica ao computador quais tarefas ele deve executar.

É a formalização da sequência de passos de um algoritmo projetado para um dado problema.

Programas de computador são escritos em linguagens específicas.

Vamos aprender a **linguagem Python**
para nos comunicarmos com um computador.

Paralelamente, vamos aprender a projetar **algoritmos!**

Linguagens de programação

Meios de comunicação para repassar instruções ao computador

Linguagem de baixo nível (ou de máquina)

- Códigos binários (sequências de 0 e 1) com sintaxe e semântica específicas. Ex. de um código binário: 01010111 (8 bits = 1 Byte)
- Aproveitam melhor a arquitetura do computador

Linguagem de alto nível

- Tem sintaxe e semântica mais próxima da linguagem humana
Ex: `print("Olá! Boa noite")` ← comando em Python
- Mais fáceis de serem aprendidas
- Ex: Python, C, C++, Java, Javascript, Fortran, Pascal, Algol, ...

Por que (para que) tantas linguagens diferentes?

Ex: programa em Python

```
# Comando "while" (enquanto)

# =====

# arquivo: imprime1a50.py

# Este programa imprime os inteiros de 1 a 50

# -----

def main():

    print("Lista dos números inteiros de 1 a 50")

    num = 1

    while num <= 50:

        print(num)

        num = num + 1

# -----

main()
```

Problemas (exemplos)

- ◆ **Problema 1** - Dada a data de nascimento de uma pessoa na forma dd/mm/aaaa, e a data de hoje 01/09/2020, determinar a **idade** dessa pessoa em **número de dias**.

Problemas

- ◆ **Problema 2** - Dada uma data dd/mm/aaaa, determinar a **data do dia seguinte (na forma) dd/mm/aaaa.**

Problemas

- ◆ **Problema 3** - Imprimir o calendário de um dado ano no formato:

July							August							September						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4							1			(*)	2	3	4	5
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10	11	12
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17	18	19
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24	25	26
26	27	28	29	30	31		23	24	25	26	27	28	29	27	28	29	30			
							30	31												

Problemas

- ◆ **Problema 4** - Dada uma data dd/mm/aaaa, determinar **qual é o dia da semana** correspondente a essa data.

(Você pode descobrir em que dia da semana vc nasceu.)

Problemas

- ◆ **Problema 5 - [Problema de ordenação]**
Dada uma sequência de números (resp. nomes)
ordená-la crescentemente (resp. alfabeticamente).

Entrada: 8 2 11 33 27 6 9 48 15

Saída: 2 6 8 9 11 15 27 33 48

Problemas

- ◆ **Problema 6 - [Problema de intercalação]**
Dadas duas sequências ordenadas de números (resp. nomes) produzir uma sequência única ordenada crescentemente (resp. alfabeticamente).

Entrada:

sequência 1:	2	16	28	49	61					
sequência 2:	5	9	33	50	60	89				

Saída: 2 5 9 16 28 33 49 50 60 61 89