Introdução à Computação

MC102

Arquitetura de Computadores

Arquitetura de Von Neumann

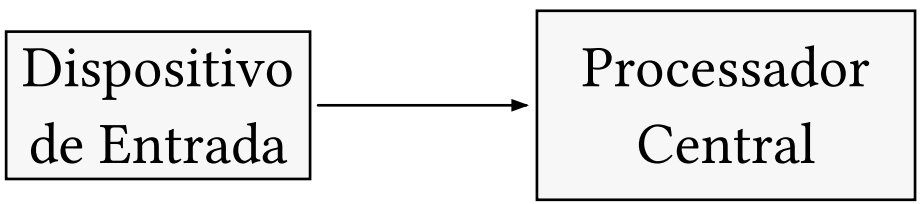
Os computadores modernos baseiam-se na **Arquitetura de *Von Neumann***.

Arquitetura de Von Neumann

Os computadores modernos baseiam-se na **Arquitetura de *Von Neumann***.

Arquitetura de Von Neumann

Os computadores modernos baseiam-se na **Arquitetura de *Von Neumann***.

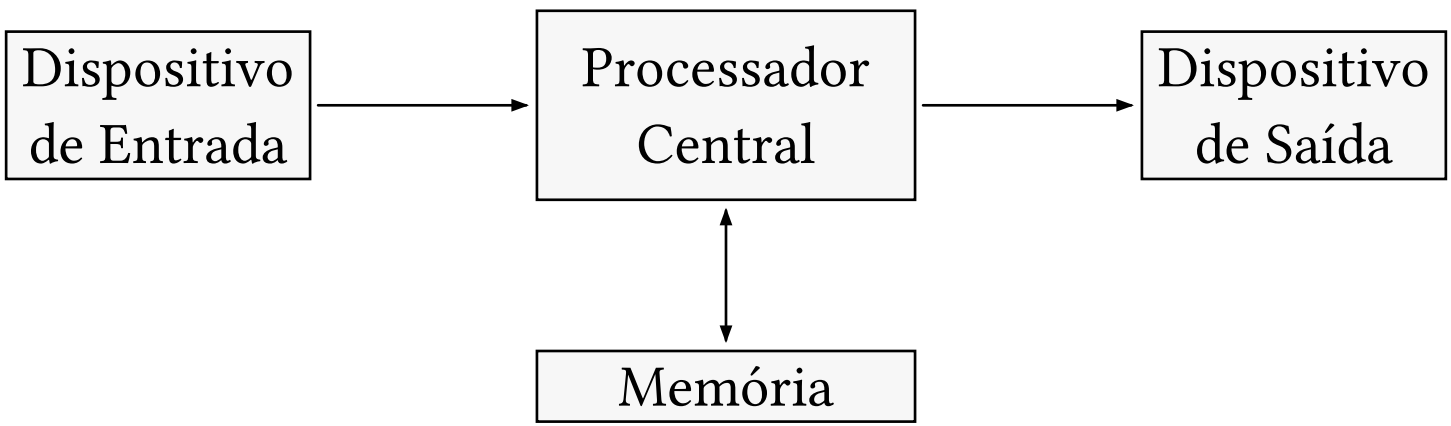


Arquitetura de Von Neumann

Os computadores modernos baseiam-se na **Arquitetura de *Von Neumann***.

Arquitetura de Von Neumann

Os computadores modernos baseiam-se na **Arquitetura de *Von Neumann***.



Dispositivos de Entrada e Saída

Dispositivos de Entrada e Saída

Dispositivos de Entrada:

Dispositivos de Entrada e Saída

Dispositivos de Entrada:

- Teclado
- Rato (Mouse)
- HD / SSD
- Ecrã Tátil (Touch)
- Microfone e Câmara
- Sensores (Temperatura, etc.)

Dispositivos de Entrada e Saída

Dispositivos de Entrada:

- Teclado
- Rato (Mouse)
- HD / SSD
- Ecrã Tátil (Touch)
- Microfone e Câmara
- Sensores (Temperatura, etc.)

Dispositivos de Saída:

Dispositivos de Entrada e Saída

Dispositivos de Entrada:

- Teclado
- Rato (Mouse)
- HD / SSD
- Ecrã Tátil (Touch)
- Microfone e Câmara
- Sensores (Temperatura, etc.)

Dispositivos de Saída:

- Ecrã (Tela)
- Impressora
- HD / SSD (também é saída!)
- Rede
- Placa de Som
- Vibração / Feedback Tátil

O Coração da Máquina

O Coração da Máquina

Processador Central (CPU): Coordena o funcionamento do computador.

O Coração da Máquina

Processador Central (CPU): Coordena o funcionamento do computador.

- Analisa e executa instruções.
- Obtém dados da memória e guarda resultados.
- Ativa dispositivos de entrada/saída.

O Coração da Máquina

Processador Central (CPU): Coordena o funcionamento do computador.

- Analisa e executa instruções.
- Obtém dados da memória e guarda resultados.
- Ativa dispositivos de entrada/saída.

Memória (RAM):

- Cada posição da memória guarda um **dado**.
- Podemos **ler** ou **escrever** dados.
- Toda a posição de memória tem um **endereço** (é assim que ela é acedida).
- As instruções do programa a ser executado também ficam alojadas na memória!

Algoritmos e Programas

Algoritmos

Algoritmos

Informalmente, um algoritmo é uma **sequência de passos** que realiza uma tarefa.
Os passos precisam de ser simples e não dependem da linguagem de programação.

Algoritmos

Informalmente, um algoritmo é uma **sequência de passos** que realiza uma tarefa. Os passos precisam de ser simples e não dependem da linguagem de programação.

> 🔥 **Tip**
>
> Estamos interessados não apenas em aprender Python, mas em aprender a **projetar algoritmos**.

Do Pseudocódigo ao Python

Do Pseudocódigo ao Python

Um pseudocódigo é uma forma abstrata de escrever o programa, focada na clareza e na solução.

Do Pseudocódigo ao Python

Um pseudocódigo é uma forma abstrata de escrever o programa, focada na clareza e na solução.

Pseudocódigo

```
soma = 0
leia num
enquanto num for diferente de zero:
    soma = soma + num
    leia num
escreva soma
```

Python

```python
soma = 0
num = int(input("Número: "))
while num != 0:
    soma = soma + num
    num = int(input("Número: "))
print("A soma é", soma)
```

Representação de Dados e Memória

A Máquina de Zeros e Uns

A Máquina de Zeros e Uns

- **Hardware** é a parte física (processador, memória). **Software** é a parte lógica (programas).

A Máquina de Zeros e Uns

- **Hardware** é a parte física (processador, memória). **Software** é a parte lógica (programas).
- O **Bit** é a menor unidade de medida do computador (apenas valores 0 ou 1).

A Máquina de Zeros e Uns

- **Hardware** é a parte física (processador, memória). **Software** é a parte lógica (programas).
- O **Bit** é a menor unidade de medida do computador (apenas valores 0 ou 1).
- Um **Byte** é uma sequência de 8 Bits.

A Máquina de Zeros e Uns

- **Hardware** é a parte física (processador, memória). **Software** é a parte lógica (programas).
- O **Bit** é a menor unidade de medida do computador (apenas valores 0 ou 1).
- Um **Byte** é uma sequência de 8 Bits.

> ⚡ **Danger**
>
> Tudo no computador são zeros e uns! Textos, imagens, sons, vídeos... A diferença é apenas **como** interpretamos esses Bytes.

Linguagens e Execução

O que é um Programa?

O que é um Programa?

Um programa é uma sequência de instruções escritas numa determinada **Linguagem de Programação** (Python, C, Java, etc.).

O que é um Programa?

Um programa é uma sequência de instruções escritas numa determinada **Linguagem de Programação** (Python, C, Java, etc.).

O texto em si é chamado de **código-fonte**.

O que é um Programa?

Um programa é uma sequência de instruções escritas numa determinada **Linguagem de Programação** (Python, C, Java, etc.).

O texto em si é chamado de **código-fonte**.

Linguagem Interpretada (Python, JavaScript) O interpretador abre o código-fonte e traduz instrução a instrução em tempo real.

O que é um Programa?

Um programa é uma sequência de instruções escritas numa determinada **Linguagem de Programação** (Python, C, Java, etc.).

O texto em si é chamado de **código-fonte**.

Linguagem Interpretada (Python, JavaScript) O interpretador abre o código-fonte e traduz instrução a instrução em tempo real.

Linguagem Compilada (Ex: C/C++) O código é transformado por completo num arquivo executável. Depois disso, já não precisa do compilador.

O Ambiente Computacional

O Ambiente Computacional

- **Sistema Operacional (Linux, Windows, macOS):** Gere o hardware e permite que os programas corram de forma segura.

O Ambiente Computacional

- **Sistema Operacional (Linux, Windows, macOS):** Gere o hardware e permite que os programas corram de forma segura.
- **Aplicações:** Os programas finais que o utilizador usa (ex: navegadores web).

O Ambiente Computacional

- **Sistema Operacional (Linux, Windows, macOS):** Gere o hardware e permite que os programas corram de forma segura.
- **Aplicações:** Os programas finais que o utilizador usa (ex: navegadores web).
- **Compiladores / Interpretadores:** Responsáveis por transformar o código fonte em aplicações (também são aplicações!).

Dúvidas?