

Departamento de Engenharia de Sistemas e Computação

Arquitetura e Execução

Modelo de von Neumann, software e o ambiente de trabalho em Python

---

João Araujo

Algoritmos Computacionais I

Aula 03

Roteiro

- Sistema de computação: hardware e software (Cap. 1.8–1.10)
- Modelo de von Neumann: CPU, memória, entrada e saída
- Compilada ou interpretada? (1.11)
- Sistema operacional (1.12)
- **Prática:** o interpretador interativo e os scripts `.py` (Cap. 2.1)

A dualidade fundamental

Hardware

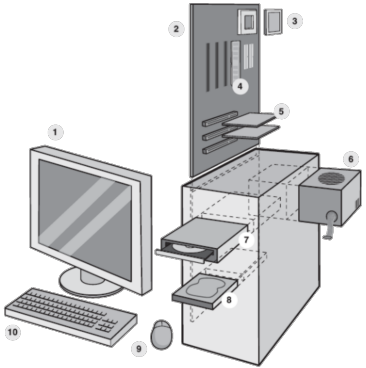
Partes físicas: eletrônicas, elétricas e mecânicas. *Tangível.*

Software

Programas e dados: o lado lógico. *Intangível.*

Um sistema de computação é a fusão dos dois: sem software o hardware não faz nada útil, e sem hardware o software não executa.

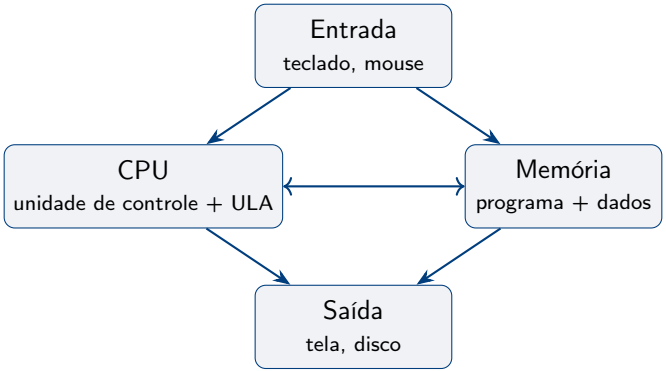
Componentes de um computador



1. Monitor
2. Placa-mãe
3. Processador (CPU)
4. Memória
5. Placas de expansão
6. Fonte de alimentação
7. Drive óptico
8. Disco rígido (armazenamento)
9. Mouse
10. Teclado

Figura do livro (Cap. 1.8).

O modelo de von Neumann



Ideia central (anos 1940): programa e dados ficam na mesma memória. Antes disso, reprogramar um computador exigia trocar cabos.

CPU, memória e ciclo de instrução

- **CPU:** executa instruções; tem unidade de controle, ULA e registradores
- **Memória principal (RAM):** rápida, *volátil*
- **Armazenamento:** lento, permanente (SSD, HD)

Ciclo da CPU

1. Buscar a instrução
2. Decodificar
3. Executar
4. Repetir (bilhões de vezes por segundo)

Frequência típica: 3 GHz $= 3 \times 10^9$ ciclos por segundo.

Software: do código-fonte ao hardware

- A CPU só entende **linguagem de máquina** (binário)
- Humanos preferem **linguagens de alto nível**: Python, C, Java...
- É preciso traduzir: duas estratégias clássicas

Compilação

Traduz tudo antes; gera um executável (C, C++).

Interpretação

Traduz e executa linha a linha (Python clássico).

Linguagem compilada

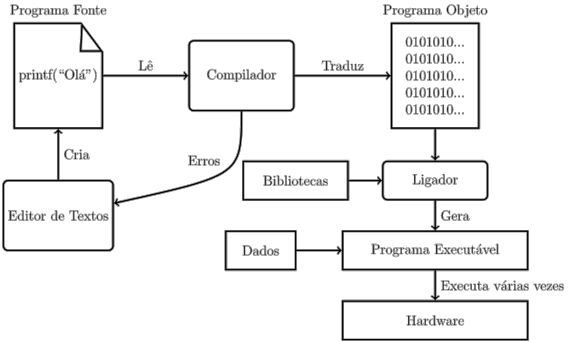


Figura do livro (Cap. 1.11). Vantagem: execução rápida. Desvantagem: ciclo editar–compilar–executar mais longo.

Linguagem interpretada

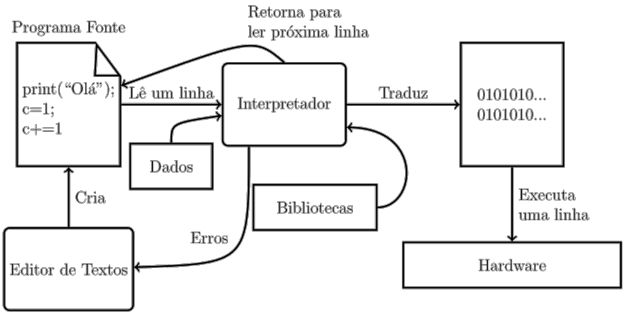
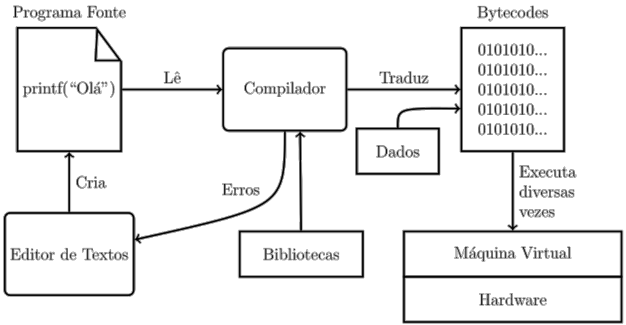


Figura do livro. Vantagem: feedback imediato. Desvantagem: mais lenta que o código compilado.

E o Python? Semi-interpretado



O fonte é traduzido para *bytecode*, executado por uma **máquina virtual** (Python Virtual Machine).

Sistema operacional

O sistema operacional (Linux, Windows, macOS, Android...) é o software que **gerencia o hardware** e serve de intermediário para os programas:

- Gerencia processos e a memória
- Controla arquivos e dispositivos de entrada e saída
- Fornece a interface com o usuário
- Isola um programa do outro (segurança)

Consequência prática

Quando seu programa abre um arquivo ou imprime algo, ele *pede* isso ao sistema operacional.

Ambiente de trabalho: duas formas de usar o Python

Interpretador interativo

Digite `python` no terminal; o prompt `>>>` aguarda um comando.
Ótimo para *experimentar*.

Script `.py`

Arquivo de texto com o programa; executa com `python programa.py`.
Ótimo para *guardar e reutilizar*.

Instalação do Python (Windows e Linux/Ubuntu): Apêndice do livro.

Prática 1: o interpretador interativo

```
>>> 2 + 3 * 4
14
>>> (2 + 3) * 4
20
>>> 7 / 2
3.5
>>> 7 // 2
3
>>> 7 % 2
1
>>> print("Olá, mundo!")
Olá, mundo!
>>> exit()
```

Prática 2: o primeiro script

Crie o arquivo `ola.py` no editor:

```
nome = input("Qual é o seu nome? ")
print("Olá,", nome, "!")
```

Execute no terminal:

```
$ python ola.py
Qual é o seu nome? Ada
Olá, Ada !
```

- Salve com *codificação UTF-8* para os acentos funcionarem
- Erros de sintaxe aparecem com o número da linha: leia a mensagem!

Para fixar

1. Desenhe o modelo de von Neumann e diga o que cada bloco guarda ou faz.
2. Python é compilado ou interpretado? Explique.
3. Qual a diferença entre digitar no editor `idle` e executar um `.py`?
4. Por que a RAM perde os dados ao desligar o computador e o SSD não?

Exercícios do livro para esta aula (Cap. 1 e Cap. 2): ver a lista de revisão da Aula 14 e o site `https://www.araujo.eng.uerj.br/livro/exercicios/cap1/`.

Resumo da aula

- Sistema de computação = hardware + software
- von Neumann: programa e dados na mesma memória; CPU executa o ciclo busca–decodifica–executa
- Python é *semi-interpretado*: bytecode + máquina virtual
- Sistema operacional gerencia recursos e arquivos
- Usaremos o interpretador interativo para testar e scripts `.py` para guardar
- Próxima aula: pensamento algorítmico, fluxogramas e pseudocódigo