



O que é um Computador? (Definição Formal)

Um **computador** é uma **máquina eletrônica programável**, projetada para **receber dados de entrada**, **processá-los** de acordo com um conjunto de instruções previamente definidas (programa), e **fornecer resultados** como saída. Ele é capaz de executar operações aritméticas, lógicas, de controle e de entrada/saída, em conformidade com uma arquitetura bem definida.



Do ponto de vista estrutural

De acordo com o modelo clássico de **John von Neumann** (1945), que ainda serve de base para a maioria dos sistemas atuais, um computador é composto por:

1. **Unidade de Entrada (Input)**: recebe dados externos (como teclado, mouse, sensores).
2. **Memória Principal**: armazena dados e instruções de forma temporária (RAM).
3. **Unidade de Controle**: interpreta as instruções e controla o fluxo de dados.
4. **Unidade Lógica e Aritmética (ULA)**: executa operações matemáticas e lógicas.
5. **Unidade de Saída (Output)**: transmite os resultados para o ambiente externo (como monitor, impressora).
6. **Barramentos**: interconectam todas essas unidades (barramento de dados, endereços e controle).



Do ponto de vista funcional

Segundo **Tanenbaum** (2014), o funcionamento básico de um computador pode ser descrito como um ciclo contínuo:

1. **Busca (Fetch)**: a CPU busca uma instrução na memória.
2. **Decodificação (Decode)**: a unidade de controle interpreta a instrução.
3. **Execução (Execute)**: a ULA executa a instrução.
4. **Escrita do resultado (Write-back)**: o resultado é armazenado em memória ou enviado para saída.

Este ciclo é conhecido como o **ciclo de instrução** e é repetido milhares a milhões de vezes por segundo (em escala de GHz).



Computador como Máquina Abstrata

Formalmente, um computador pode ser modelado como uma **máquina de Turing prática** — uma máquina finita com memória e uma unidade de controle programável. Embora não infinita como a proposta por Turing (1936), um computador real implementa a mesma ideia: manipular símbolos com base em regras, o que o torna capaz de simular qualquer outro computador — uma propriedade chamada de **universalidade computacional**.



Componentes físicos (hardware) vs. Lógicos (software)

- **Hardware**: é a parte física, os circuitos, chips, cabos, dispositivos de E/S etc.

- **Software:** são os programas e sistemas operacionais que controlam o hardware e fornecem funcionalidade ao usuário.

Citações Fundamentais

- **Tanenbaum (2014)** – *Structured Computer Organization*:

“Um computador é uma máquina que executa instruções armazenadas em memória, com base em uma arquitetura de controle que interage com entradas e saídas de forma organizada e previsível.”

- **Patterson e Hennessy (2013)** – *Computer Organization and Design*:

“Computadores são sistemas que transformam dados de entrada em informações úteis, por meio de uma organização hierárquica de componentes interdependentes.”

- **John von Neumann (1945)** – *First Draft of a Report on the EDVAC*:

“A unidade de controle e a unidade de processamento devem operar em conjunto sobre uma única memória, tanto para instruções quanto para dados, permitindo a automação de tarefas computacionais complexas.”

Computadores no Contexto de Compiladores

Computador como Máquina de Execução

No contexto de **compiladores**, o computador é visto não apenas como um dispositivo físico, mas como uma **máquina abstrata de execução**, capaz de **entender e processar instruções** codificadas em uma linguagem de máquina.

O compilador traduz um programa escrito em **linguagem de alto nível** (como C, Java, Python) para uma linguagem de **baixo nível ou linguagem de máquina** (código de máquina ou Assembly), que a **Unidade de Controle** da CPU pode interpretar e executar.

Formalização: Máquina Abstrata e a Matemática do Computar

A origem formal do conceito de "computador" está enraizada na **Teoria da Computação**, cujos fundamentos foram lançados por nomes como:

- **Alan Turing** (1936): Máquina de Turing.
- **Alonzo Church** (1936): Cálculo Lambda.
- **Stephen Kleene, Emil Post, Kurt Gödel**: Sistemas formais e decidibilidade.

Esses modelos **matematizam o conceito de computar** e estabelecem **os limites do que é possível computar**.

Máquina de Turing: O Modelo Idealizado de Computador

Componentes da Máquina de Turing:

1. **Fita infinita** (memória)
2. **Cabeçote de leitura/escrita**
3. **Conjunto finito de estados**
4. **Função de transição** (a "programação")

Ela mostra que qualquer problema computável pode ser resolvido com um **modelo de controle finito + memória ilimitada**.

Essa estrutura inspirou os compiladores modernos ao definir **o que pode ou não ser computado**, e como problemas são **reconhecidos por autômatos**.

Compiladores: Ponte entre teoria e máquina real

Um **compilador** implementa conceitos **formais** e **práticos**, como:

1. Análise léxica (Autômatos Finitos Determinísticos - AFD)

- Palavras-chave, identificadores, literais.
- Baseado em **gramáticas regulares** e expressões regulares.

2. Análise sintática (Autômatos de Pilha / Gramáticas Livres de Contexto)

- Reconhece a estrutura do código.
- Utiliza **gramáticas formais** (ex: BNF, EBNF).

3. Geração de código intermediário

- Transforma para uma forma próxima da máquina, porém independente da arquitetura.

4. Geração de código de máquina

- Traduz para **Assembly ou binário**, executável pela CPU real.
- Conhecimento da arquitetura (registradores, instruções, barramentos).

O Computador como Executor de Linguagens Formais

Um computador, ao fim, é o **executor de linguagens formais**.

- Toda linguagem compilada precisa ser reconhecida por um **autômato formal** (AFD, AP, MT).
- O compilador é construído com base nesses modelos.
- O computador físico executa o **resultado dessa tradução**.

Fontes Teóricas Fundamentais

- **Hopcroft, Motwani, Ullman – Introduction to Automata Theory, Languages, and Computation**

Apresenta as fundações teóricas do que significa "computar", com enfoque em compiladores.

- **Aho, Lam, Sethi, Ullman – Compilers: Principles, Techniques, and Tools** (o famoso *Dragão*)

Obra clássica sobre compiladores, relacionando teoria formal e práticas reais de implementação.

- **Alan Turing – On Computable Numbers, with an Application to the Entscheidungsproblem**

Define computabilidade formal e a ideia da máquina de propósito geral.

Conclusão

Um **computador**, no contexto de compiladores, não é apenas um dispositivo físico que executa instruções. Ele é o **modelo concreto de uma máquina abstrata** definida na matemática. Compiladores utilizam as **teorias de linguagens formais, autômatos e computabilidade** para transformar código-fonte em **instruções executáveis** pelo hardware.

Formalmente, um **computador** é uma **máquina programável** baseada em uma arquitetura lógica e física bem definida, capaz de processar dados por meio de um conjunto de instruções. Ele é composto por unidades interdependentes e opera ciclicamente para transformar entradas em saídas significativas.

Esse conceito une fundamentos teóricos (como a Máquina de Turing), arquiteturas práticas (como a de von Neumann), e implementações físicas (hardware moderno), formando a base da computação contemporânea.